BSS-ITS: Blockchain Scaling Scheme with Sharding for Intelligent Transportation System

Scale Blockchain for Better Data Exchange and Storage with Full Sharding for Intelligent Transportation System

Yufei Liu*
Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University
yufeiliu@bjtu.edu.cn

Jiqiang Liu
Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University
jqliu@bjtu.edu.cn

Jian Wang†
Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University
wangjian@bjtu.edu.cn

Tianhao Liu
Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University
llllll@bjtu.edu.cn

Xudong He
Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University
hexudong@bjtu.edu.cn

ABSTRACT

With the development of modern technologies, Intelligent Transportation System (ITS) has made great progress and brought convenience to every aspect of people's lives. ITS has become more complicated and uncertain due to the existing security and privacy problems. Blockchain, being a decentralized and tamper-resistant technology, can provide an environment for ITS, allowing ITS entities to perform transactions securely and trustfully. However, the inefficiency problem of blockchain, severely restricts its development and application. Studies only combine traditional blockchain with ITS, but do not consider scaling blockchain to improve performance. This paper devotes to scale blockchain that adapted to ITS, improving the efficiency of the blockchain without sacrificing the decentralization and security. A sharding scheme is proposed, which takes network sharding, data sharding and state sharding into account. Nodes are divided into several shards. Different shards can process data in parallel and store related data in their own ledger. Backup shard is put forward in this paper to enhance the shard security. And network will reshuffle when one or more shards break down. Some experiments are performed using Hyperledger Fabric and network simulator 3. Throghput, storage and delay are tested under sharding scheme and non sharding solution. The results show that sharding method is effective in scaling blockchain for ITS.

CCS CONCEPTS

• **Information systems**; • **Information storage systems**; • **Storage management**;

KEYWORDS

Intelligent Transportation System, Blockchain, Sharding, Hash, Efficiency

ACM Reference Format:
Yufei Liu, Jiqiang Liu, Jian Wang, Tianhao Liu, and Xudong He. 2021. BSS-ITS: Blockchain Scaling Scheme with Sharding for Intelligent Transportation System: Scale Blockchain for Better Data Exchange and Storage with Full Sharding for Intelligent Transportation System. In *2021 4th International Conference on Blockchain Technology and Applications (ICBTA 2021), December 17–19, 2021, Xi'an, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3510487.3510506

1 INTRODUCTION

As the modern sensing, communication and computing technologies and devices develops, Intelligent Transportation System has made continuous progress in academic research and practical applications, bringing significant impacts on daily lives [1]. Due to the increasing uncertainty and complexity of the strategies and mechanisms involved, ITS shows a high degree of complexity, and there are many issues that need to be addressed. One problem is that ITS is highly centralized, the system may be crashed and unable to reach when it encounters malicious attacks [2]. The other problem is that entities of ITS are hard to trust each other [3].

Some scholars have combined blockchain with various aspects of applications in ITS [4]. Use the advantages of blockchain to solve

*Yufei Liu is currently working toward the Master's degree in cyberspace security at Beijing Jiao Tong University, Beijing, China.

†Jian Wang received Ph.D. degree in cryptography from Beijing University of Posts and Telecommunications, in 2008. Since July 2008, Dr. Wang has been teaching and researching at Beijing Jiaotong University. His current research interests includes big data security and analysis, quantum computing, cryptography application and authentication technology, and computer forensics technology.

ICBTA 2021, December 17–19, 2021, Xi'an, China
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8746-0/21/12…$15.00
https://doi.org/10.1145/3510487.3510506

the existing problems of ITS. Blockchain is a tool for solving trusted interactions in a decentralized way, and it has gradually come into view since the publication of the Bitcoin white paper in 2008 [5]. Blockchain adopts data encryption, timestamps, distributed consensus, and economic incentives to achieve decentralized transaction management. The impossible triangle of blockchain, i.e., efficiency, security and decentralization, often cannot be satisfied at the same time [6] and has been the limitation and bottleneck of blockchain's development. And efficiency is often sacrificed to meet decentralization and security. However, ITS is sensitive to time delay [7]. It is difficult to meet the demand of real-time by applying the traditional blockchain to ITS, which also hinders the application of blockchain for ITS.

There are studies aiming to scale the blockchain and improve the efficiency of the blockchain [8]. There exist two main ways scaling blockchain called on-chain extension and off-chain extension. The off-chain scaling is often limited, and the on-chain scaling can be done by increasing the block capacity [9], but it is not very effective in improving blockchain's throughput; and directed acyclic graph structure are adopted for parallel processing, and each block may have more than one parent block, so it is difficult to achieve the final consistency. Sharding is difficult to implement, but it is also the promising method to scale blockchain. The existing research about sharding mostly focuses on public chains and UTXO (unspent transaction output) models, but does not provide solutions for real-life applications.

In this paper, we mainly adopt sharding to scale the blockchain for ITS. Scaling blockchain makes blockchain better used in ITS and solves the security and privacy problems common in ITS. The main contributions of this paper are as follows.

- Apply the sharding to scale blockchain for intelligent transportation application.
- Propose a blockchain sharding scheme adapted to the intelligent transportation scenario.
- Bring up backup shards to solve the centralization and security problems of sharding.

The chapters of this paper are organized as follows: the first section introduces the current situation of ITS, Blockchain and Blockchain scaling technology, the second section introduces the research progress of blockchain application in ITS and sharding, the main scheme of the paper is introduced in the third section, the fourth section performs the simulation experiments, and the fifth section concludes the paper.

2 RELATED WORKS

This section discusses some related research works, so far, have not been found aimed at scaling the use of blockchain in specific scenarios. The following content will be divided into two parts to introduce the relevant research work respectively.

2.1 Intelligent Transportation System and Blockchain

Blockchain applications in ITS are diversified, such as, security and privacy, record storage and transaction model. Lu et al. propose a blockchain-based anonymous reputation system that uses predefined specifications to ensure the trustworthiness of the system [10], which is believed to solve the security and privacy problems in ITS. A study combines blockchain and a traffic event verification system in ITS [11] to achieve the goal of quickly and effectively protecting vehicle privacy and reducing the impact of malicious vehicles transmitting confusing information. Through the trust value of the vehicle as well as the history behavior to determine whether the vehicle has transmitted false information, can effectively identify the malicious nodes.

There are transactions in intelligent transportation applications too, such as vehicle refueling, charging, and parking, etc. Combined with blockchain, which stores transaction records securely and permanently on the chain [12], also simplifies the transaction process. By recording vehicle-related data and driving history on the blockchain [13], the cause of the accident can be clearly identified and the responsible party can be determined through the data on the chain. Using blockchain to store some data will inevitably consume a lot of energy and take up a lot of storage space. A study [14] tried to solve this problem by reducing the number of transactions mainly through optimal transaction models and picking some moments to update the ledger instead of doing it in real time.

Obviously, blockchain can provide solutions to some centralized system, addressing the existing issues. However, some researchers ignore the problems of blockchain efficiency or don't get the real nature of the problem. Blockchain itself should be scaled.

2.2 Sharding

Sharding, as one of the most promising technology to scale blockchain, has gained much attention. Elastico [15] is the first work to apply the sharding to blockchain, including network sharding and transaction sharding. Its shard size is small and cannot handle cross-shard transactions. An intra-committee consensus algorithm is proposed in [16] and supports performing state sharding to optimize the storage problem. It needs to randomly re-shard periodically, which involves a lot of data migration work. The idea of ledger pruning is mentioned in [17], which also facilitates the reduction of excessive backup content of nodes. It also proposes atomicity protocols for cross-shard transactions. However, it needs to reorganize the network to resistant malicious attackers and brings data migration problems. Considering the fact that sharding will reduce the attack cost of a single shard, a kind of Chu-ko-nu mining [18] is proposed to make the shard consensus security equal to the consensus security when it is not sharded. But since an appropriate incentive mechanism is needed to make as many nodes as possible participate in Chu-ko-nu mining, which also poses the risk of centralization.

Sharding schemes mentioned in this section are all based on public blockchain. As blockchain is more widely used, some application fields prefer to adopt consortium blockchain for better management. So, scaling consortium blockchain should also be included in the scope of research.

3 BSS-ITS DESIGN

Considering the features of ITS, consortium blockchain is preferred to be the basic framework of ITS. Sharding consists of network sharding, data/transaction sharding and state sharding, as shown

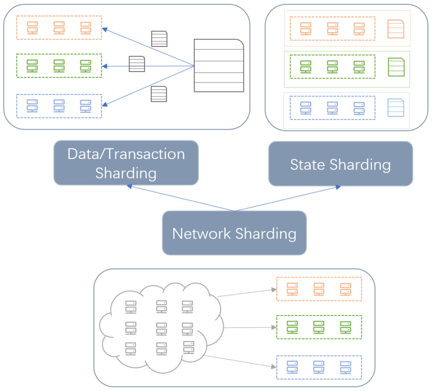


Figure 1: Network sharding, data sharding and state sharding.

in Figure 1. Network sharding partitons nodes into several communities, called shards, and it is the basis of data sharding and state sharding. Distributing data to different shards for verification is the data sharding process. With state sharding, nodes store blocks that generated by specific shards. Sharding presents in this work may be quite different from other works due to the ITS. Detailed information will be introduced in the latter sections. Table 1 lists and explains symbols that occur in this paper. And some tuples used in paper are presented in Table2.

3.1 Network Sharding

The network model in this paper is shown in Figure 2, and the main components include roadside units (RSUs), vehicles and some infrastructure. RSU nodes are responsible for verifying, storing and sharing the received uploaded data; vehicles, as entities with mobility, are responsible for collecting the surrounding roads and traffic information and uploading them to the nearby RSU nodes, and obtaining the required data information from RSUs to make driving predictions, etc. The infrastructure exists as some auxiliary in the scenario, and its data exchange will not be described in detail.

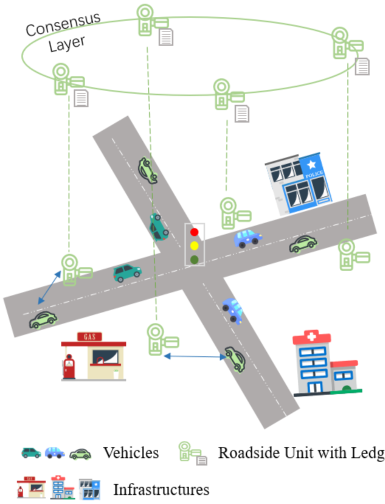


Figure 2: Network model of Intelligent Transportation System with blockchain.

1. Identity Registration

Table 1: Description of symbols

Symbol	Description	Symbol	Description
ID	vehicle's real identity	Location	RSU node's physical location
Type	vehicle's type, eg. BMW, Benz	Pk_r, Sk_r	key pairs of RSU node
Color	vehicle's color	District	region that RSU node locates
Position	vehicle's physical location	$Road_{no}$	road that RSU node locates
Pk_v, Sk_v	key pairs of vehicle	Description	detailed information of where RSU node is located
T-Value	vehicle's trust value	hash()	hash function, get a fixed length string of the parameter
IP	IP address of RSU node	$shard_{index}$	RSU node's shard index
Provider	manufacturers of RSU	$shard_{num}$	number of shards

Table 2: Description of tuples

Tuple Info	Description
(ID, Type, Color, Position)	vehicle's initial identity tuple
(Pk_v, Sk_v, Position, T-Value)	vehicle's identity tuple after registration
(IP, Provider, Location)	RSU node's initial identity tuple
(IP, Pk_r, Sk_r, Provider, Location)	RSU node's identity tuple after registration
(District, $Road_{no}$, Description)	RSU node's Location field
(IP, Pk_r, Sk_r, Provider, Location, $Shard_{index}$)	RSU node's identity tuple after sharding
(IP, Pk_r, Location, $Shard_{index}$)	RSU node's routing info

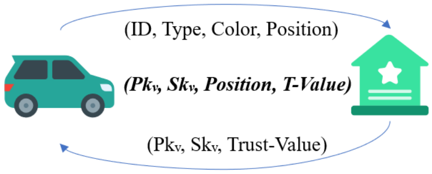


Figure 3: Vehicle registers an identity.

Both RSUs and vehicles need a legitimate identity before they can participate in the network. Therefore, when joining the network, both types of entities need to register their identities with a trust authority (TA) and obtain a public-private key pair that can represent legitimate identity. First of all, in order to register an identity, the vehicle sends a set of information that represent its own characteristics to TA, including ID, Type, Color and Position. TA randomly generates a pair of public and private keys for the vehicle, and specifies the initial trust value of the vehicle after receiving registry request. The trust value will be influenced by the subsequent behavior of the vehicle and will affect the judgment of RSU on the reliability of the uploaded data of the vehicle. Use the public key to identify the vehicle and hide the real identity of the vehicle. After the registration, the identification tuple of the vehicle can be seen in Table 2, the process is shown in Figure 3. The identity registration of RSU is similar with vehicle registration. RSU node sends IP, Provider and Location to TA. TA also generates a pair of public and private keys for RSU. Once the registration is completed, the identification tuple of RSU can be seen in Table 2

2. Nodes partition

Select some nodes into the same shard with reference to the physical location of the nodes. The Location field of RSU node mentioned in identity registration contains 3 parts, i.e., the district, the road label, and description, see in Table 2. When performing network partition, it is necessary to determine the division range according to the network size. There are two ways to calculate the shard index of nodes, see formula 1 and 2.

$$shard_{index} = hash\,(District)\ \ mod\ shard_{num} \tag{1}$$

$$shard_{index} = hash\,(District,\ Road_{no})\ \ mod\ shard_{num} \tag{2}$$

Considering the predictability of this partition, a malicious node may gather nodes within the same shard in advance and launch a node aggregation attack. Therefore, map the hash value to the hash ring and get a random number by verifiable random function (VRF). As shown in Figure 4. where the nodes of the same color belong to the same shard. Add a certain degree of randomness, making it difficult for malicious nodes to predict the results of nodes partition.

3. Intra-shard Routing

After the network nodes are divided, the nodes need to store the routing information of other nodes within the same shard to facilitate the data consensus verification process. For each shard, a temporary master node is randomly selected. Broadcast all the master nodes' information in the network, and the rest of the nodes store the received information. Each node sends routing information to its master node in the same shard, and the temporary master nodes construct the intra-shard routing table, and then send to other nodes in the shard based on the routing information. Other nodes store the received routing table, as described in Algorithm 1

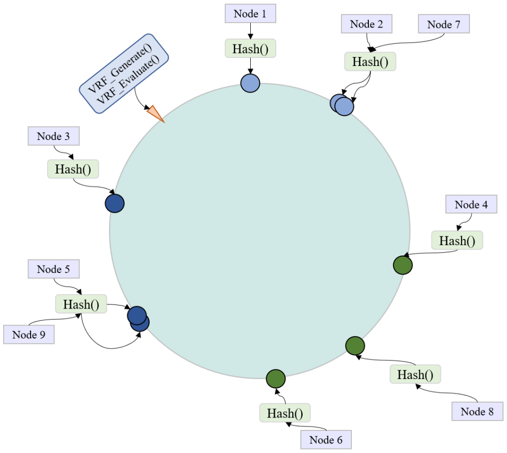


Figure 4: Nodes partition.

Algorithm 1 Construct intra-shard routing table

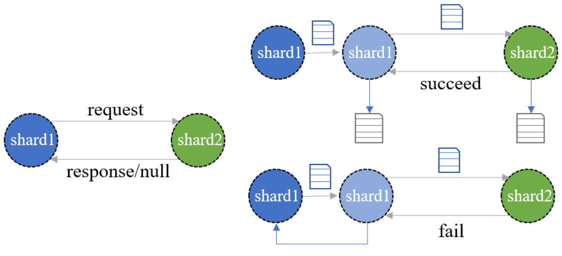
```
procedure Construct routing table
    T_leader ⟵ [], Nodes ⟵ []
    K ⟵ number of shards, C ⟵ number of members of a shard
    for each  i ∈ [1, K] do
        num ⟵ random (1, C)
        T_leader.add(node_num)
    end for
    broadcast(T_leader)
    for each leader in T_leader
        for each node in Nodes
            if shardIndex(node) = shardIndex(leader)
                node.sendInfo(leader)
            end if
        end for
    end for
    for each leader in T_leader do
        table ⟵ leader.constructRouting()
        leader.sendRouting(table)
    end for
end procedure
```

3.2 Data Sharding

Based on network sharding, disjoint data is distributed to different shards. Shards can verify data in parallel, improving the overall throughput.

1. Data Distribution and Consensus Validation

The vehicle collects uploads data to a nearby RSU node. The RSU node that receives the data acts as the creator of the current shard block and forwards the data information to the other nodes in the same shard for consensus verification. The shard nodes only participate in the consensus verification process of the uploaded data



a) unidirectional request b) bidirectional process

Figure 5: Cross-shard communication examples.

that location related. So that the data verification can be performed simultaneously between different shards.

The intra-shard consensus uses the Practical Byzantine Fault Tolerance (PBFT) [19] algorithm, which consists of three main phases, i.e., pre-prepare, prepare, and commit phases. In the pre-preparation phase, the current block creator forwards the data to other nodes. In the preparation phase, each node sends the verification results to other nodes in the shard. And the commit phase, each node sends confirmation or invalidation messages based on the number of valid results received.

2. Cross-Shard Communication

The data validation mentioned before belongs to the category of intra-shard communication. After partitioning the network, it is inevitable that data requests between different shards will be processed, which belongs to the category of cross-partition communication.

Cross-shard communication can be divided into two types. One is a unidirectional data request, where the result only affects the requesting party, which satisfies atomicity. The other is a bidirectional data request, where the result affects both parties. And there may be a situation where one party succeeds in execution and the other fails, which violates atomicity. Therefore, the second type of cross-shard communication is handled in an asynchronous manner, where one party executes firstly and sends the result to the other party. If the other party succeeds in execution, it replies to the requesting party and both parties pack the data processing record into a block and store it on their own ledger. If the execution fails, it replies to the requesting party and roll back to the initial state of execution, as shown in Figure 5

3.3 State Sharding

Based on the network sharding, data can be distributed to different shards in a disjoint manner. Each shard can only store the data related to the shard. The data is stored by the nodes of a specific shards instead of all nodes of the network, reducing excessive redundant backups and allowing the nodes to store more data.

1. Backup Shards

With the introduction of state sharding, the data stored by nodes can be reduced to a certain extent. But nodes also face the problem of data unavailability due to shard failure or attack. After the network is divided, the number of nodes in the shard decreases compared to the undivided one, and the corresponding attack cost decreases. Therefore, a corresponding backup shard is set for each shard. The backup shard is responsible for storing the key data of the corresponding shard and performing secondary verification of the shard blocks generated by the shard, i.e., trust first and verify later. The backup shard is selected by calculating the physical distance between the center of each shard, see Figure 6, and selecting the nearest shard or shards as the backup shard(s).

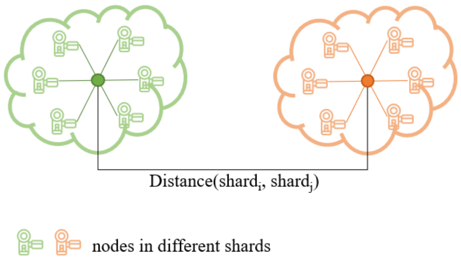


Figure 6: Distance between two shards.

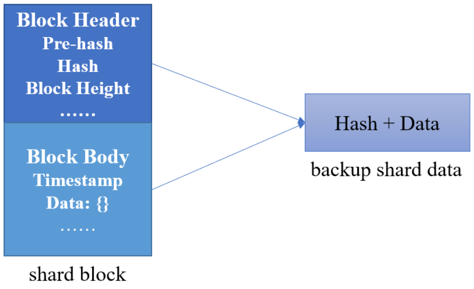


Figure 7: Shard blocks and related backup shard data.

Each shard has its corresponding backup shard and also acts as a backup shard of one or some shards. There is no dependence on a central shard, so the whole system can avoid centralization. The shard generates shard blocks and sends the main information of blocks to backup shard for secondary verification, as shown in Figure 7. In addition, backup shard will store the related data in its ledger too.

In addition to verifying the received data, the backup shard also has the responsibility of monitoring whether the shard is in normal state simultaneously. Backup shard monitors the shard's state by verifying the data and regularly requesting the corresponding shard to send shard blocks. So, if adversary does not want to be detected doing evil, it is almost necessary to gather more than 2/3 nodes of each shard in the whole network.

2. Data Pruning

Intelligent transportation applications are sensitive to both latency and information timeliness. Data, that beyond a certain time, is no longer of any practical meaning [20] in the network. Therefore, set check point at the position where data reaching a specific length of time from the current moment and remove the previous data on the chain. To avoid temporary data unavailability, the specific time is set to 3 hours. Therefore, data checkpoint setting and data deletion is performed every 3 hours instead of timely and frequently. Establishing checkpoints at the deleted data is mainly to construct

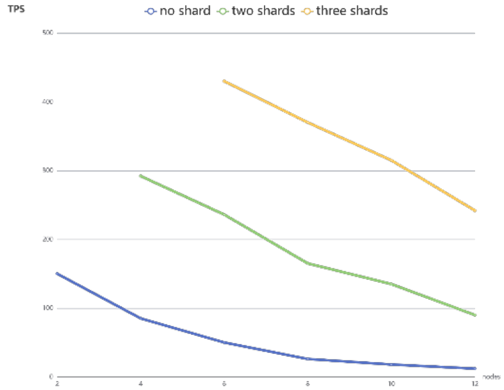


Figure 8: Experimental results about throughput.

a Merkle tree of the data to be deleted, store the root node of the Merkle tree, and link at the deleted position.

4 EXPERIMENTAL EVALUATION

The Hyperledger Fabric [22] environment was built locally using a virtual machine Ubuntu 20.04 with 8GB RAM and 120G hard disk. And use the Hyperledger tape [23] to test the blockchain performance. Also, use the network simulation tool ns-3 [24] to simulate the node communication process and test the latency during processing. Experiments on throughput, storage and delay are done under non sharding and sharding conditions. Detailed information and analysis are presented as follows.

1. Throughput

Limited by the performance of the computer, run 12 nodes to do the tests. There are three types of setting for the network, one is the original with no shard, one is divided into two shards, and the last one is partitioned to three shards. The throughput of the system with different setting is compared in Fig. 8. Apparently, throughput is higher with more shards. On the one hand, the number of nodes in the shard is reduced, and the consensus time is shortened. On the other hand, each shard can work independently and in parallel.

2. Data Storage

Each node stores the shard block generated by the shard that node belongs to and part of the main data as the backup shard. The size of memory space occupied by a single shard block is defined as one unit, which is called BS, and the corresponding data stored in the backup shard is about 1/2 of one unit, i.e., 0.5 BS. The network setting is same as the throughput tests.

Figure 9 compares the memory space consumption in different shard cases. The specific units are omitted in the figure. It can be seen that the storage space occupied after sharding tends to be less than that without sharding, and the network with more shards occupies less storage space than the one with fewer shards. Apparently, nodes store less block data when the number of shards is higher.

3. Delay

The time delay mainly refers to the time taken for the whole process from the vehicle uploading data or requesting data to the RSU to getting a reply. In the experiment, the packet size is 1024 bytes, each packet transmission interval is 0.05s, the channel transmission rate is 100Mbps, and the channel delay is 2ms. Experiments are done with different number of nodes in a shard, with the number of nodes set from 10 to 100.

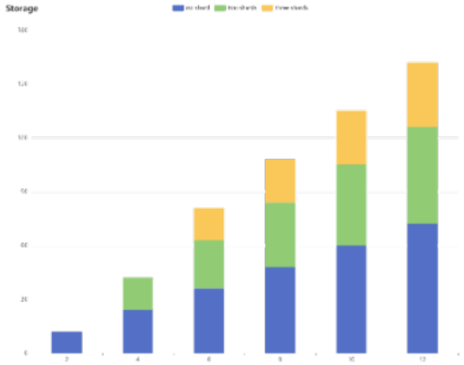


Figure 9: Comparison of data storage.

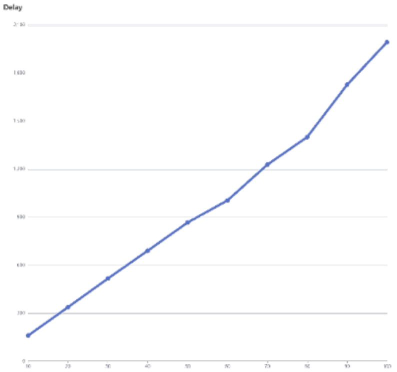


Figure 10: Experimental results about delay.

Figure 10 compares the latency of the system with different number of nodes in the shard, and the latency of the system tends to increase when the number of nodes increases, but the overall latency is still in the acceptable range. Set the appropriate number of nodes in a shard according to delay requirements.

5 CONCLUSION

In this paper, we focus on the scaling blockchain with sharding for ITS. Therefore, the blockchain can better solve the problems common in the ITS. A sharding approach is proposed to improve the performance of blockchain and ensure the security as well as decentralization. The experimental results show that the sharding scheme is effective for blockchain scaling, after which more detailed applications will be considered and detailed experiments will be conducted to optimize the scheme.

ACKNOWLEDGMENTS

This work was supported in part by the Major Scientific and Technological Innovation Projects of Shandong Province, China (No.2019JZZY020128). Scientific and Technological Development

Plan Projects of China National Railway Group Limited (No. N2020W005) also gave significant assistance to finish this work.

REFERENCES

[1] Y. Yuan and F. Wang, "Towards blockchain-based intelligent transportation systems," 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016, pp. 2663-2668, doi: 10.1109/ITSC.2016.7795984.

[2] D. Maffiola, S. Longari, M. Carminati, M. Tanelli and S. Zanero, "GOLIATH: A Decentralized Framework for Data Collection in Intelligent Transportation Systems," in IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2021.3123824.

[3] S. Distefano, A. D. Giacomo and M. Mazzara, "Trustworthiness for Transportation Ecosystems: The Blockchain Vehicle Information System," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 4, pp. 2013-2022, April 2021, doi: 10.1109/TITS.2021.3054996.

[4] M. B. Mollah *et al.*, "Blockchain for the Internet of Vehicles Towards Intelligent Transportation Systems: A Survey," in IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4157-4185, 15 March15, 2021, doi: 10.1109/JIOT.2020.3028368.

[5] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008, [online] Available: https://bitcoin.org/bitcoin.pdf.

[6] Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," 2017 IEEE International Congress on Big Data (BigData Congress), 2017, pp. 557-564, doi: 10.1109/BigDataCongress.2017.85.

[7] J. Zhang, F. Wang, K. Wang, W. Lin, X. Xu and C. Chen, "Data-Driven Intelligent Transportation Systems: A Survey," in IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, pp. 1624-1639, Dec. 2011, doi: 10.1109/TITS.2011.2158001.

[8] K. Wang and H. S. Kim, "FastChain: Scaling Blockchain System with Informed Neighbor Selection," 2019 IEEE International Conference on Blockchain (Blockchain), 2019, pp. 376-383, doi: 10.1109/Blockchain.2019.00058.

[9] S. Kim, Y. Kwon and S. Cho, "A Survey of Scalability Solutions on Blockchain," 2018 International Conference on Information and Communication Technology Convergence (ICTC), 2018, pp. 1204-1207, doi: 10.1109/ICTC.2018.8539529.

[10] Lu, Z.; Wang, Q.; Qu, G.; Liu, Z. Bars: A blockchain-based anonymous reputation system for trust management in vanets. In Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy in Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018; pp.98–103.

[11] Al-Ali, M.S.; Al-Mohammed, H.A.; Alkaeed, M. Reputation Based Traffic Event Validation and Vehicle Authentication using Blockchain Technology. In Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2–5 February 2020; pp. 451–456.

[12] Miller, D. Blockchain and the Internet of Things in the Industrial Sector. IT Prof. 2018, 20, 15–18.

[13] Guo, H.; Meamari, E.; Shen, C.C. Blockchain-inspired Event Recording System for Autonomous Vehicles. In Proceedings of the 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN), Shenzhen, China , 15–17 August 2018; pp. 218–222.

[14] Sharma, V. An energy-efficient transaction model for the blockchain-enabled internet of vehicles (IoV). IEEE Commun. Lett. 2018, 23, 246–249.

[15] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert and P. Saxena, "A secure sharding protocol for open blockchains", Proc. ACM SIGSAC Conf. Comput. Commun. Secur., pp. 17-30, Oct. 2016.

[16] M. Zamani, M. Movahedi and M. Raykova, "RapidChain: Scaling blockchain via full sharding", Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), pp. 931-948, 2018.

[17] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta and B. Ford, "OmniLedger: A secure scale-out decentralized ledger via sharding", Proc. IEEE Symp. Secur. Privacy (SP), pp. 583-598, May 2018.

[18] J. Wang and H. Wang, "Monoxide: Scale out blockchains with asynchronous consensus zones", Proc. 16th USENIX Symp. Netw. Syst. Design Implement. (NSDI), pp. 95-112, Feb. 2019, [online] Available: https://www.usenix.org/conference/nsdi19/presentation/wang-jiaping.

[19] Castro M , Liskov B . Practical Byzantine Fault Tolerance[J]. ACM Transactions on Computer Systems (TOCS), 2002.

[20] Chen H , Wang Y . SSChain: A full sharding protocol for public blockchain without data migration overhead[J]. Pervasive and Mobile Computing, 2019, 59:101055.

[21] P. Zheng, Q. Xu, Z. Zheng, Z. Zhou, Y. Yan and H. Zhang, "Meepo: Sharded Consortium Blockchain," 2021 IEEE 37th International Conference on Data Engineering (ICDE), 2021, pp. 1847-1852, doi: 10.1109/ICDE51399.2021.00165.

[22] Androulaki E , Manevich Y , Muralidharan S , *et al.* Hyperledger fabric: a distributed operating system for permissioned blockchains[C]. the Thirteenth EuroSys Conference. 2018.

[23] https://github.com/Hyperledger-TWGC/tape.git.

[24] https://www.nsnam.org/.