Journal of Network Intelligence ©2022 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **7**, Number **4**, November **2022**

Shared Car Platform Scheme Based on Optimized PBFT Algorithm

Di Zhang

School of Aeronautics and Astronautics,
Zhejiang University
No.38, Zheda Road, Hangzhou, 310027, China
d.zhang@zju.edu.cn

Min Zhou*

School of Aeronautics and Astronautics,
Zhejiang University
No.38, Zheda Road, Hangzhou, 310027, China
zhoumin@zju.edu.cn

Li-Wei Qu

Shandong Institute of Space Electronic Technology
Space Road, Yantai, 264000, China
quliwei5599@126.com

Hua Chen

School of Aeronautics and Astronautics,
Zhejiang University
No.38, Zheda Road, Hangzhou, 310027, China
chenhua@zju.edu.cn

Jiong-Jiong Mo

School of Aeronautics and Astronautics,
Zhejiang University
No.38, Zheda Road, Hangzhou, 310027, China
jiongjiongmo@zju.edu.cn

Fa-Xin Yu

School of Aeronautics and Astronautics,
Zhejiang University
No.38, Zheda Road, Hangzhou, 310027, China
fxyu@zju.edu.cn

*Corresponding author: Min Zhou

Received July 7, 2022, revised August 13, 2022, accepted October 4, 2022.

Abstract. *Car sharing service is an innovative way of transportation, which can improve the efficiency of traffic time. However, the traditional centralized scheme faces the problems of high cost, low efficiency, and difficulty in restricting non-compliance. We propose an automobile sharing scheme based on consortium blockchain, which uses the Honesty and Practical Byzantine Fault Tolerance (HPBFT) method, combines personal credit with automobile identity, and combines reward and punishment mechanisms to restrict and reduce malicious default events. Smart contract standardizes the entire transaction process and allows transaction data to be tracked throughout. In this scenario, we verify that the HPBFT algorithm achieves at least 36.52% higher throughput and 15.71% lower response latency than the Practical Byzantine Fault Tolerance (PBFT) through the Hyperledger Fabric open source framework.*

Keywords: Car-sharing; Consortium blockchain; Data security; Smart contract; Hyperledger Fabric;

1. **Introduction.** Automobile sharing is a new service mode through organizational innovation, human-vehicle redistribution, resource sharing, and resource utilization. Its emergence has solved the traffic problems in urban areas to a certain extent, such as road traffic congestion, fuel combustion pollution, and parking shortage caused by the increase in the number of vehicles [1, 2]. The car sharing system provides the benefits of using private vehicles without the cost and responsibility of ownership. Users can use their smartphones to book and rent shared cars on the online service platform in the system. However, since information is transmitted through public networks, malicious attackers can easily eavesdrop, forge, delete and modify information [3]. If the digital key or code for access control is exposed, malicious attackers can completely control the shared car and steal it. The system must therefore ensure a credible network environment for secure communications. In addition, it also needs the authentication function to check whether the user has the right and ability to drive the car. Users must submit their information (identity and driving license) to service providers when applying for car-sharing services [4, 5]. Service providers need to verify that customers have the right and ability to drive before enjoying car-sharing services through service providers.

In traditional car sharing systems, user information and service information can be stored and controlled in a centralized service server. However, centralized servers are subject to a single point of failure from a malicious attacker. Once the data server is compromised, all transaction records are deleted and difficult to recover. In addition, if a user commits fraud during car sharing and the shared records are tampered with or rewritten, it will be difficult to obtain evidence of a user's crime from these records. In addition, if users commit fraud in the process of car sharing, and the shared records are tampered with or rewritten, it is difficult to obtain criminal evidence of users from these records, and the disclosure of stored information will bring serious privacy problems [6, 7]. Therefore, although the emergence of car sharing system can alleviate traffic problems, it also has the following problems: 1) users are denied access to transaction details, and malicious tampering is not supervised; 2) The lack of supervision by the enterprise and the government in the transaction process leads to the abuse of resources; 3) There is no code of responsibility between the company, the user and the car when problems arise.

Blockchain is considered a credible data storage technology and is essentially a decentralized distributed book database[8]. Data is maintained by multiple nodes in the decentralized network, and the special chain storage method effectively ensures the tamperproof and traceability of data, which will not cause data leakage and loss when a single node breaks down [9]. The consortium blockchain will authenticate and manage the rights of nodes joining the network and will not make the data public. Therefore, it is suitable for dealing with business scenarios that require consensus among organizations. So it is suitable for dealing with business scenarios that require consensus among organizations. In order to solve the above problems, we propose a car-sharing scheme based on the consortium blockchain. The main contributions of this paper are as follows.

1. Proposed a framework of car-sharing solution based on consortium blockchain, defined the responsibilities of each component and the workflow of this scheme;
2. A smart contract suitable for the platform scheme is designed to standardize the transaction details, record the whole process of vehicle usage records, and facilitate accident accountability and information traceability;
3. An Honest Practical Byzantine Fault Tolerant (HPBFT) algorithm is proposed, which combines the personal credit system with vehicle identity and uses reward and punishment mechanism to limit user malicious behavior. The experimental results show that the throughput of HPBFT is 36.52 % higher than that of Practical Byzantine Fault Tolerant (PBFT) in this scenario.

The research background and work of this paper are introduced above, followed by the related work of this paper. There are component architecture, the working process of the system and the detailed design of the smart contract in the next chapter. The fourth chapter introduces HPBFT improvement ideas of the algorithm. The fifth chapter is the experimental part, which builds a test environment based on the Hyperledger Fabric open-source framework for testing and results analysis. The last chapter summarizes the work of this paper.

2. Related work.

In recent years, the emergence of blockchain technology and its decentralized, tamper-proof and secure features have inspired some researchers to study the field of car sharing. Valaštin et al. [10] proposed a smart contract running Solidity language to realize a car sharing system based on Ethereum. Kim et al. [11] proposed a distributed car-sharing authentication protocol and provided mutual authentication for the proposed model through Automatic Verification of Internet Security Protocols and Applications (AVISPA) and The Burrows Abadi Needham (BAN) logic analysis. Xu et al. [12] proposed a blockchain network for car sharing data pricing using the Stackberg game. Vaidaya et al. [13] discussed the security and privacy issues in vehicles and put forward explicit requirements for car sharing systems. Zhou et al. [14] used Hyperledger Fabric to verify the proposed distributed car-sharing control system, showing that the deployment of smart contracts can effectively improve data security and trust. Although the above solution solves the basic problem of combining car sharing system and blockchain, they make no effort to establish and maintain the relationship between individual credit, service providers, and regulators, which makes it impossible to confirm and trace the responsibility once malicious acts arise.

Based on the related work, there is no complete solution system for the malicious behavior of users and the regulation of government enterprises in the car sharing field, so this paper proposes a solution for car sharing based on the consortium blockchain and an optimized HPBFT algorithm based on this scenario.

3. Car sharing platform framework.

Considering that the platform is composed of many companies and the data involves personal information and privacy, we use the consortium blockchain as the platform infrastructure, which can strengthen the control of data access, and the data is not open to the public, and effectively protect the interests of users [15].

3.1. **Overview.** Based on the current transaction flow of the shared platform, we designed the platform framework as shown in Figure 1. The framework can be divided into four parts. Ministry of Transportation, service operators, blockchain networks and users. Here are the details of each section. Transportation Department: The transportation department plays an important role in the transaction verification stage. The transport department can check the use of vehicles, that is, order information, to see if there are traffic violations and whether the user's real-name information is verified. Standardize user and vehicle behavior through traffic rules and vehicle operation rules.

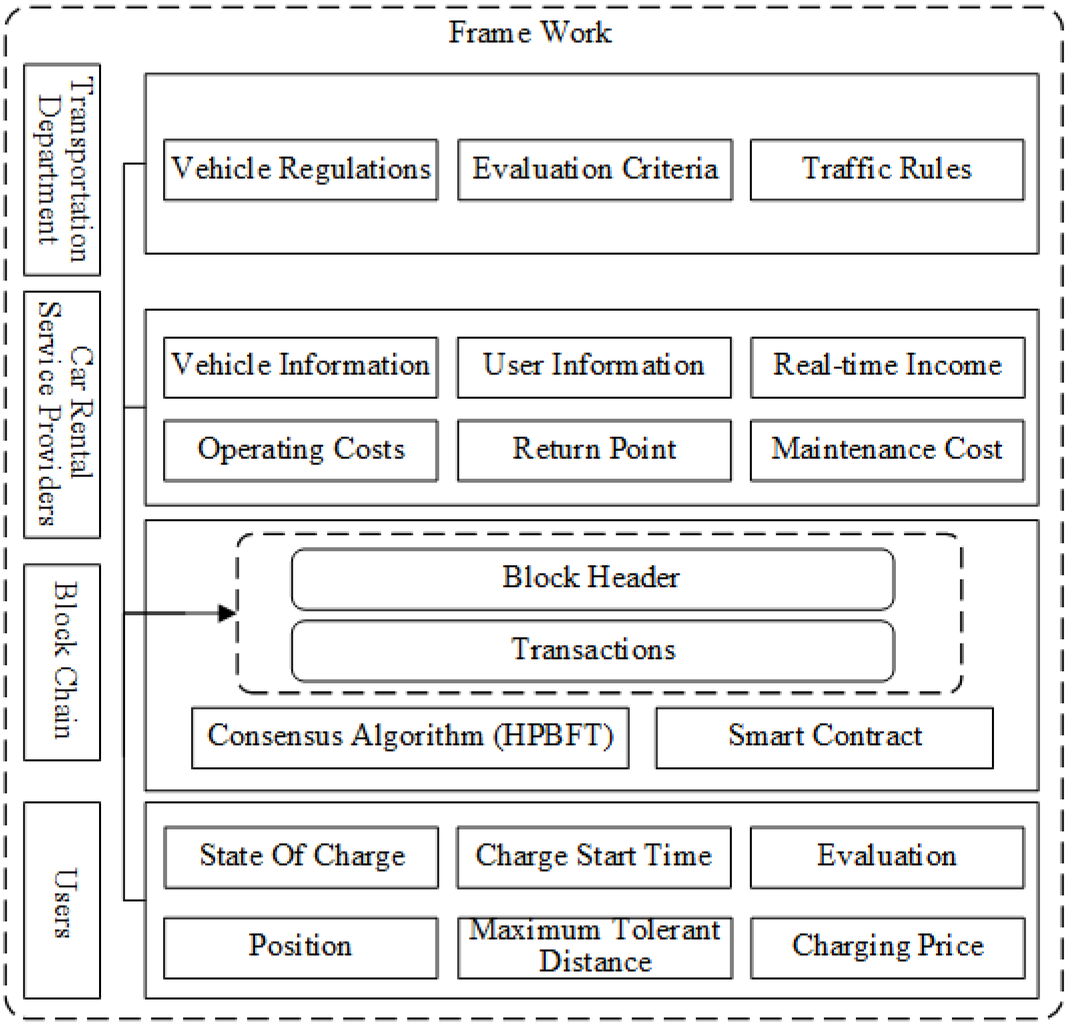

FIGURE 1. Component Architecture.

Car Rental Service Providers: Provide vehicles for users to use. The service operator needs to register the vehicle information into the blockchain network through a smart contract. The traffic department first audits the qualification of the service operator, checks the vehicle information after the audit is passed, and completes the encryption of the vehicle information after all are passed. The service operator has the right to check the user's order information and confirm the authenticity of the order, and after reaching a consensus with the nodes of the transportation department, generates a block and conducts transactions on the chain.

Blockchain Network: A blockchain network is formed by connecting users, service operators, and transportation departments to the network layer. The various actions of the four actors are carried out under the constraints of smart contracts, and the consensus mechanism ensures the consistency of the data of each node.

User: It is the initiator of the transaction and the user of the vehicle. The user initiates the request and registers the corresponding information (user identity information and user driving qualification) into the blockchain network, and the smart contract completes the verification of the information, packages the information to generate blocks, and writes them to the blockchain after consensus is completed. Users can check the car status and billing rate through this system. After initiating a transaction, the service provider and the transportation department will reach a consensus. After the transaction is approved, they can get the right to use the car.

Vehicle: Each vehicle is equipped with a multi-sensor matrix and the real-time data of the car will follow the end of the user's order for uploading operation. The final vehicle data is stored in the blockchain network for retention as a basis for user accident recovery and information traceability.

3.2. Workflow.

The workflow of the proposed car-sharing is shown below.

1. The rental car service operator registers the vehicle information on the blockchain and requests the vehicle nodes to upload the current status. Users use the client to register personal information and complete user registration after verification by the transportation department.

2. The user selects a suitable vehicle by checking the status and unit price of the car and initiates a request.

3. The transportation department verifies the user's information and the user's driving qualification, the service operator verifies the user's information, the user's wallet balance, and the vehicle status, and the user confirms the transaction after passing. The transaction information is packaged to generate blocks and written to the blockchain after the user's confirmation is correct. The user gets the right to use the vehicle.

4. After the order is completed, the traffic department and the service provider confirm the illegal information and vehicle status information during the process of user access.

5. After the confirmation is correct, the user's wallet balance is deducted, the transaction ends and the user lose the vehicle right.

3.3. Design of Smart Contract.

The smart contract is a protocol that is executed in a computer in the form of chain code and it allows for trusted transactions without third parties [16]. The smart contract settings mainly include three aspects: information registration, transaction commencement, and transaction termination. Combined with the system workflow, a set of smart contracts has been redesigned.

1. Information Registration. Information registration occurs when the rental car service operator registers vehicle information and new users register, and the smart contract completes the verification of vehicle identification information and user qualifications, and finally uploads vehicle information and user information to the blockchain network respectively to complete the information registration contract. The registration of vehicle information and user information can be executed independently. The pseudo-code for the above process implementation is as follows.

Table 1. Information Registration

```
Algorithm 1: Information Registration
Input: information
if the information is valid then
    calculate info hash with UUID (vehicle id or user id)
    if the info hash is new and the qualification pass then
        create a new record
    end
    fill the record with information
    upload the record
end
```

2. Transaction Commencement. Users use the client to know the status of the vehicle for model selection. If the vehicle operation information is normal, then the order can be submitted. The smart contract will confirm the identity of the vehicle ID and user ID, and then bind the user and vehicle to generate an order after the audit. The order information includes user ID, vehicle ID, order initiation time, car start time, car end time, order completion time, and other transaction information. The pseudo-code for the above process implementation is as follows.

Table 2. Transaction Commencement

```
Algorithm 2: Transaction Commencement
Input: user, vehicle
if the user's driving qualification is valid then
    check vehicle status
    if the status is normal then
        calculate order hash with order id
        if order Hash is new then
            create a record that contains this transaction
        end
        upload the record
    else
        report a vehicle for repair
end
```

3. Transaction Termination. The order terminates by parking the vehicle in the parking space and the client automatically initiates a transaction termination request. The digital signatures of the user and the vehicle are first verified and authorization of the vehicle is stopped. The traffic department and service operator check the vehicle status information and violations and then update the vehicle information to keep the vehicle information real-time. If the damage to the vehicle is caused by that user's road violation, the user will pay for the required repairs. The pseudo-code for the above process implementation is as follows.

TABLE 3. Transaction Termination

```
Algorithm 3: Transaction Termination
Input: order
if order State is terminated then
        query vehicle information and fill the information of the order
        take back vehicle authority
        if the vehicle status has an abnormal violation record then
                transportation department decides on punitive measures
                the cost increase with the violation fee
        end
        deduct fees from user Wallet
        update information on the transaction
end
```

4. **Optimized Consensus Mechanism.**

The characteristic of the PBFT consensus algorithm is that it can tolerate the adverse effects of some fault nodes and chaotic nodes on the network, and has polynomial-level algorithm complexity [17]. However, due to the consensus protocol, PBFT has a high probability of malicious master node election, high communication complexity, and poor dynamic scalability [18, 19]. Therefore, we improve the traditional PBFT algorithm from the three aspects of the main node election mechanism, communication complexity and scalability.

4.1. **Node Status Description.**

We describe the different consensus behaviors of nodes in different states of the consensus process by the scores calculated from the nodes' performance in each consensus round and define it as the honesty degree. The nodes with excellent consensus performance are rewarded to increase the incentive for them to reach consensus, and to improve the efficiency of consensus. Additionally, in order to ensure that there are no branches in the PBFT algorithm, the total number of nodes N in the network and the number of byzantine nodes f satisfy Equation.1, there can be at most f byzantine nodes in a network of N nodes, so at least four nodes are required to reach consensus in PBFT with one master node [20].

$$\boldsymbol{N \geq 3f + 1} \tag{1}$$

All nodes in the network are given excellent and good status integrity in the proportion of 1:3 during the initialization of the consortium blockchain network. As the number of consensus increases, there will be a large gap in the honesty between nodes, and the nodes involved in consensus will change thus improving the efficiency of the execution of the consistency protocol.

The honesty level is divided into four levels to describe the behavior of nodes. When byzantine behavior is too numerous or the behavior is too long, we remove nodes from the network and authenticate them again before they can join the network. However, a majority of nodes in the consortium blockchain are honest nodes, so a smaller range is set to mark such nodes, and the other three states respectively are equated to a magnitude of 10-90. If the node status is excellent, it proves that the node can participate in consensus and has the right to be elected as a master node; if the node status is qualified, the node can participate in consensus but cannot be elected as a master node; if the node status is average, the node cannot participate in consensus and can only synchronize consensus results; if the node status is poor, it will be kicked out of the blockchain network and the network administrator will again authenticate and The network administrator can join the network after authenticating and verifying the node again. The state transition diagram of the node is shown in Figure 2.

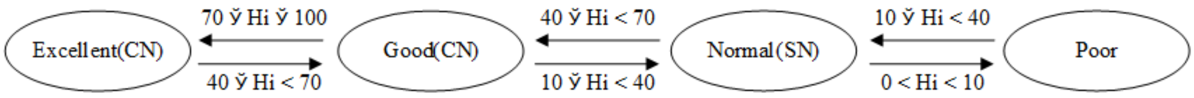

Figure 2. Node State Transition Diagram.

Where $\boldsymbol{H_i}$ is the honesty of nodes, $\boldsymbol{CN}$ is the set of nodes participating in consensus, and $\boldsymbol{SN}$ is the set of nodes that only synchronize the consensus results and out of the consensus process. After network initialization, the integrity of nodes is adjusted according to the consensus performance of nodes in the previous round, and the integrity is also updated during checkpoint protocol implementation. If a node shows malicious or wrong behavior in the consensus process, it is considered that the stability of the node is poor, and the honesty degree should be reduced to punish the node and limit the behavior of the node in the consensus process. If the node actively participates in the consensus and contributes to the final consensus consistency, it is considered that there is room for the node to rise in status and increase honesty to motivate the node's behavior in the next consensus.

4.2. **Consistency protocol optimization.** HPBFT specifies that each node maintains a Node Member List (NML) locally, which records the basic node information such as the number of current system nodes, identity information, IP address, honesty degree and block summary [21]. In order to ensure the correctness of the integrity, nodes need to update the summary information of corresponding nodes in NML in time. It is defined as the summary of the information stored at the node calculated by the node. Figure 3 shows the consensus process of the HPBFT algorithm.

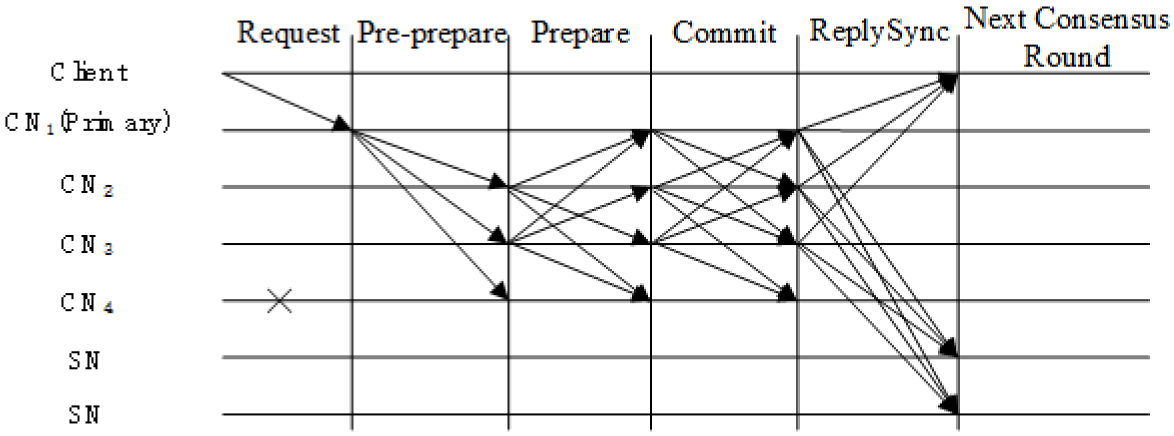

Figure 3. The Flow of HPBFT

Process 1. Starting the blockchain network, giving the initial honesty to the consensus nodes, and electing the master node in the consensus node collection to prepare the consensus.

Process 2. The client $\boldsymbol{c}$ sends a message request to the node $\boldsymbol{CN_1}$ with the message format $\langle \boldsymbol{REQUEST, o, t, c} \rangle$. where $\boldsymbol{o}$ is the specific operation requested, $\boldsymbol{t}$ is the timestamp appended by the client at the time of the request, and $\boldsymbol{c}$ is the client number.

Process 3. After $\boldsymbol{CN_1}$ receiving the request message from the client, the master node verifies the correctness of the signature of the request message from the client, discards the illegal request, and assigns a request number $\boldsymbol{n}$ to broadcast the Pre-prepare message to other nodes in the consensus node collection. The message format is $\boldsymbol{\left\langle \langle PRE-PREPARE, n, d_p, p\rangle, m \right\rangle}$, $\boldsymbol{m}$ is the message content, $\boldsymbol{d_p}$ is the new block summary calculated by the master node, $\boldsymbol{p}$ is the current master node's serial number, and enters the Prepare phase.

Process 4. Replica $\boldsymbol{CN_{\mathrm{i}}}$ in the consensus node collection receives the $\boldsymbol{PRE{-}PREPARE}$ message, verifies whether the message signature is correct and whether the message with number $\boldsymbol{n}$ has been received, the node $\boldsymbol{i}$ calculates the message digest $\boldsymbol{d}$ of request $\boldsymbol{m}$, and updates $\boldsymbol{d_{i,p} = d_p}$, $\boldsymbol{d_{i,i} = d}$to NML. illegal requests are discarded, if the node $\boldsymbol{i}$ has received 2f+1 verified If the node $\boldsymbol{i}$ receives 2f+1 validated pre-prepare messages, it sends a $\boldsymbol{PRE-PREPARE}$ message in the format $\boldsymbol{\langle PREPARE, n, d_i, i\rangle}$ to other nodes in the consensus node collection, and the consensus enters the confirmation phase.

Process 5. Similarly, the node $\boldsymbol{i}$ receives the $\boldsymbol{PRE-PREPARE}$ message and verifies the signature and number. The $\boldsymbol{d}$ and $\boldsymbol{d_i}$ of the messages sent by different $\boldsymbol{CN}$ nodes are compared, and if 2f+1 validated $\boldsymbol{PRE-PREPARE}$ messages are received, a $\boldsymbol{COMMIT}$ message is sent to the replica node $\boldsymbol{CN}$. The message format is $\boldsymbol{\langle\langle COMMIT, n, d_i, i\rangle, D_{local}\rangle}$ and $\boldsymbol{D_{local}}$ is the set of message summaries sent by each node saved by the node $\boldsymbol{i}$.

Process 6. After receiving 2f+1 commit messages, the node $\boldsymbol{i}$ sends a $\boldsymbol{REPLYSYNC}$ message to the client and $\boldsymbol{SN}$ node in the format $\boldsymbol{\langle REPLYSYNC, n, t, c, r, i\rangle}$, $\boldsymbol{n}$ is the message number, $\boldsymbol{t}$ is the timestamp, $\boldsymbol{c}$ is the client, $\boldsymbol{r}$ is the message summary returned to the client, $\boldsymbol{r}$ is this node number.

Process 7. If the client and the synchronous node receive 2f+1 identical $\boldsymbol{REPLYSYNC}$ messages, the client's request has reached network-wide consensus and the synchronous node completes writing the block to the blockchain. Each node in the network calculates and updates the honesty and local NML to prepare for the next round of consensus.

4.3. **Reward & Punishment Mechanism.** In the consensus process of the HPBFT algorithm, the node $\boldsymbol{i}$ obtains the message digest $\boldsymbol{D_{local}(j)}$ computed by other nodes through a three-stage consensus (node $\boldsymbol{i}$ is used as an example). The hash $\boldsymbol{D_{local}(i)}$ is subjected to the operation of Eq. 2, respectively.

$$\boldsymbol{D_{local}(j)_p \oplus D_{local}(i)_p (p = 0, 1, 2 ... n - 1)} \tag{2}$$

According to equation 1, it can be determined whether a Byzantine node appears in this consensus round. If a Byzantine node appears, then the node $\boldsymbol{i}$ will take the initiative to request the digest $\boldsymbol{d_{j,p}}$ from the current node number k in $\boldsymbol{D_{local}(j)}$ from node $\boldsymbol{j}$. The determination of the Byzantine node can be divided into two cases.

1) If no hash result is received or received inconsistently, the node $\boldsymbol{j}$ is considered a Byzantine node.

2) If the received hash result is consistent with the local hash value of the node $\boldsymbol{i}$, then the node numbered $\boldsymbol{p}$ is considered to have an error in the digest calculation and the node $\boldsymbol{p}$ is a Byzantine node.

A node is considered to become a Byzantine node if it shows evil or faulty behavior. Therefore, we record three parameters of node production block number $\boldsymbol{C_{block}}$, malicious count $\boldsymbol{C_{eviling}}$, and fault count $\boldsymbol{C_{fault}}$ as the evaluation criteria of consensus performance. When Byzantine behavior occurs, a timer T with timing time $\boldsymbol{T_{threshold}}$ is started to record the time of node Byzantine behavior. In order to avoid that the master node will still participate in consensus as the master node after Byzantine behavior, so the honesty after the previous round of consensus is used as a reference, and the formula for consensus performance is shown in Equation.3.

$$\boldsymbol{H_t^*} = \begin{cases} \boldsymbol{H_{t-1}^*} + (\frac{C_{block}}{N_{block}} - \frac{T_i(C_{eviling}+C_{fault})}{\sum\limits_{i=1}^{n} T_i(C_{eviling}+C_{fault})} & \boldsymbol{T_i \leq T_{threshold}} \\ \boldsymbol{0} & \boldsymbol{T_i > T_{threshold}} \end{cases} \tag{3}$$

Where $\boldsymbol{H_t^*}$ is the honesty of the current consensus, $\boldsymbol{H_{t-1}^*}$is the honesty of the previous round of consensus,$\boldsymbol{T_i}$is the time when Byzantine error occurs in node $\boldsymbol{i}$. The node $\boldsymbol{i}$ can only improve its honesty and thus restore its original consensus right through excellent consensus performance.

5. **Experimental Results.** In this experiment, we use Docker container virtualization technology to simulate multiple nodes for experiments under the same LAN on the three servers [22]. The Java SDK simulates the transaction request initiated by the client by sending HTTP requests and uses the Caliper test tool to record the delay and throughput of the request. We build the underlying blockchain framework of Hyperledger Fabric by applying HPBFT and traditional PBFT respectively and conduct a comparative experiment on the throughput and response delay of the shared car blockchain application proposed in this paper [23]. With a timer threshold value of 5s, we analyze the algorithm by theoretical and experimental results. The hardware configuration is shown in Table 4.

Table 4. Hardware Configuration

CPU	Memory	Disk (in G)	Bandwidth (in MBps)	Operate System
Intel Xeon 5660 @2.2GHz 24 cores	128G	256, SSD	1000	CentOS 7
Intel Xeon 5660 @2.2GHz 24 cores	128G	256, SSD	1000	CentOS 7
Intel Xeon 5660 @2.2GHz 24 cores	128G	256, SSD	1000	CentOS 7

5.1. **Throughput Analysis.** Throughput is an important indicator to measure the blockchain system, which refers to the number of transactions processed by the system in unit time. In the blockchain field, it is usually regarded as the transaction volume (Transactions per Second, TPS) per unit time [24]. The calculation formula is shown in Equation.4 below.

$$\boldsymbol{TPS} = \frac{\boldsymbol{Transactions_{\Delta t}}}{\boldsymbol{\Delta t}} \tag{4}$$

Where $\boldsymbol{\Delta t}$ is a certain length of time interval, $\boldsymbol{Transactions}$ is the volume of transactions in the blockchain system during the time. The number of nodes and the number of block transactions in a blockchain network affect the calculation of throughput. Therefore, we specify that each block contains 100 transactions, the block generation interval is 2s, and a single client sends 500 requests per second to test the throughput under different numbers of nodes, and the average results after five experiments are shown in Figure 4.

As can be seen in Figure 4, the throughput of both algorithms is more stable with an increasing number of nodes for a different number of nodes, and the throughput of the HPBFT algorithm is always higher than that of the traditional PBFT algorithm. The throughput difference decreases from 155.31 to 134.06, indicating that the rate of decline in HPBFT throughput is less than that of PBFT, while the throughput difference of the HPBFT algorithm improves by 44.14

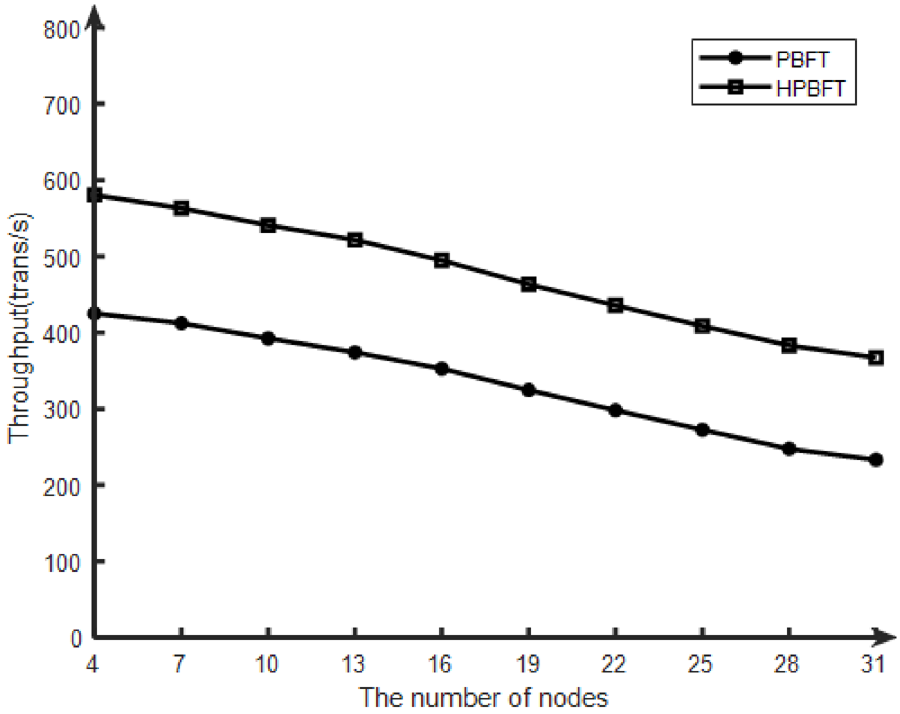

Figure 4. Comparison of Throughput.

5.2. **Latency Analysis.** Response latency is another performance indicator of the blockchain consensus algorithm, which is the time difference between a request sent by the client and the response received [25]. Since there is a certain error in the calculation of a single request, we use to send three sets of the random number of requests and calculate the average of the three sets separately before performing the average calculation to get the final value. We also specify the number of transactions per block to be 100 and the block generation interval to be 2s for the test configuration, and the results are shown in Figure 5.

As can be seen in Figure 5, the difference in response latency between the two algorithms is not very obvious when the number of nodes is small, and it is difficult to distinguish them in the actual application system. However, when the number of nodes increases, the delay difference between HPBFT and PBFT increases from 39.94ms to 589.71ms, indicating that HPBFT has superior delay performance to PBFT when the number of nodes is high. The trend in the figure shows that HPBFT is more stable and has shorter response latency. In the car-sharing field, this performance is sufficient to carry the request pressure of car-sharing platform users, so it can meet the current applications in the car-sharing industry.

6. **Conclusions.** This paper proposes a car-sharing scheme based on a consortium blockchain, uses the honesty-based HPBFT consensus algorithm to define node state transition, solves the problem of electing a faulty master node, and uses a reward and punishment mechanism to punish Byzantine nodes to improve the enthusiasm of nodes in the network. Transactions are standardized and automated by deploying dedicated smart contracts to

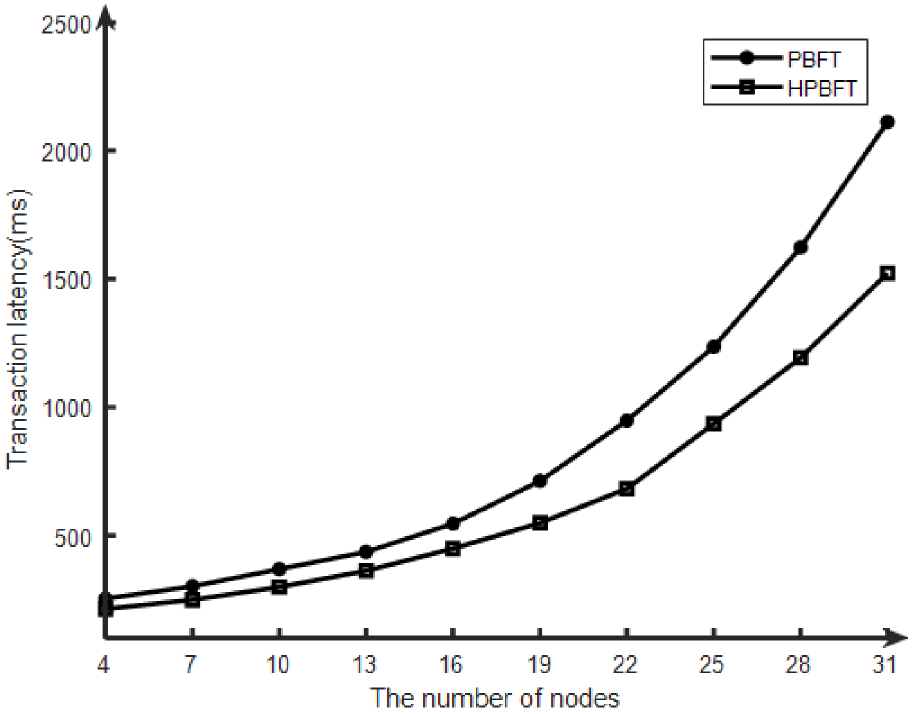

Figure 5. Comparison of Latency.

each node. The combination of blockchain and car-sharing systems provides a stable and efficient blockchain platform that solves the privacy and traceability of user data and features high stability, low latency, and Byzantine fault tolerance. Through experiments, the results show that the throughput of HPBFT is at least 36.52% higher than that of PBFT, and the response delay is at least 15.71% lower, which proves that the improved consensus algorithm and smart contract can be applied to the shared car blockchain system.

In future work, we will optimize the consensus algorithm, and the currently proposed scheme requires a polynomial level of communication between nodes for each transaction consensus. In this paper, personal credit and node honesty are bound as node evaluation conditions in the consortium blockchain. In addition, the consortium blockchain node authority has strict control management, so the nodes can be considered as honest nodes in theory. Therefore, it can also optimize the consistency protocol, reduce the response delay again, and provide a good experience for larger point-to-point network scenarios.

REFERENCES

[1] F. Ferrero, G. Perboli, M. Rosano and A. Vesco, "Car-sharing services: An annotated review," *Sustainable Cities and Society*, vol. 37, pp. 501-518, 2018.

[2] S.A. Shaheen and A.P. Cohen, "Car sharing and personal vehicle services: worldwide market developments and emerging trends," *International Journal of Sustainable Transportation*, vol.7, no.1, pp.5-34, 2013.

[3] M. Yavari, M.h Safkhani, S. Kumari, S. Kumar and C.M. Chen, "An Improved Blockchain-Based Authentication Protocol for IoT Network Management," *Security and Communication Networks*, vol.2020, 8836214, 2020.

[4] T.Y. Wu, Z.Y. Lee, L. Yang, J.N. Luo and R. Tso, "Provably Secure Authentication Key Exchange Scheme Using Fog Nodes in Vehicular Ad-Hoc Networks," *The Journal of Supercomputing*, vol.77, no.7, pp.6992–7020, 2021.

[5] C.M. Chen, X.T. Deng, W.S. Gan, J.H. Chen and SK Islam, "A secure blockchain-based group key agreement protocol for IoT," *The Journal of Supercomputing*, vol.8, no.77, pp.9046-9068, 2021.

[6] T.Y. Wu, X.L. Guo, Y.C Chen, S. Kumari and C.M. Chen, "SGXAP: SGX-Based Authentication Protocol in IoV-Enabled Fog Computing," *Symmetry*, vol.7, no.14, 1393, 2022.

[7] Q. Mei, H. Xiong, Y.C. Chen and C.M. Chen, "Blockchain-Enabled Privacy-Preserving Authentication Mechanism for Transportation CPS with Cloud-Edge Computing," *IEEE Transactions on Engineering Management*, 2022. [Online]. Available: https://doi.org/10.1109/ TEM.2022.3159311

[8] Y. Xu, X. Li, X. Zeng, J.K. Cao and W.B. Jiang, "Application of blockchain technology in food safety control: current trends and future prospects," *Critical reviews in food science and nutrition*, vol.10, no.62, pp.2800-2819, 2022.

[9] A. Dorri, M. Steger, S.S. Kanhere and R. Jurdak, "Blockchain: A distributed solution to automotive security and privacy," *IEEE Communications Magazine*, vol.55, no.12, pp.119-125, 2017.

[10] V. Valaštín, K. Košt'ál, R. Bencel and I. Kotuliak, "Blockchain based car-sharing platform," in *International Symposium ELMAR*, IEEE, 2019, pp.5-8.

[11] M. Kim, J. Lee, K. Park, Y. Park, K.H. Park and Y. Park, "Design of secure decentralized car-sharing system using blockchain," *IEEE Access*, vol.9, pp.54796-54810, 2021.

[12] C. Xu, K. Zhu, C. Yi and R. Wang, "Data pricing for blockchain-based car sharing: A Stackelberg game approach," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, IEEE, 2020, pp.1-5.

[13] B. Vaidya and H.T. Mouftah, "Security for shared electric and automated mobility services in smart cities," *IEEE Security & Privacy*, vol.19, no.1, pp.24-33, 2020.

[14] Q. Zhou, Z. Yang, K. Zhang, K. Zheng and J. Liu, "A decentralized car-sharing control scheme based on smart contract in internet-of-vehicles," in *IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, IEEE, 2020, pp.1-5.

[15] L. Li, J.Q. Liu, L.C. Cheng, Q. Shuo, W. Wang, X.L. Zhang and Z.H. Zhang, "Creditcoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol.19, no.7, pp.2204-2220, 2018.

[16] H. Su, B. Guo, Y. Shen and X.H. Suo. "Embedding Smart Contract in Blockchain Transactions to Improve Flexibility for the IoT," *IEEE Internet of Things Journal*, 2022. [Online]. Available: https://doi.org/ 10.1109/JIOT.2022.3163582

[17] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," *OsDI*, vol.99, no.1999, pp.173-186, 1999.

[18] H. Wang, K. Guo and Q, Pan, "Byzantine fault tolerance consensus algorithm based on voting mechanism," *Journal of Computer Applications*, vol.39, no.6, pp.1766-1771, 2019.

[19] E.K. Wang, R.P. Sun, C.M. Chen, Z.D. Liang, S. Kumari and M.K. Khan, "Proof of X-repute blockchain consensus protocol for IoT systems," *Computers & Security*, vol.95, 101871, 2020.

[20] Z. Li, S. Chen and B. Zhou, "Electric vehicle p2p electricity transaction model based on superconducting energy storage and consortium blockchain," in *2020 IEEE International Conference on Applied Superconductivity and Electromagnetic Devices (ASEMD)*, IEEE, 2020, pp.1-2.

[21] K. Lei, Q.H. Hang, L.M. Xu and Z.Y. Qi, "Reputation-based byzantine fault-tolerance for consortium blockchain," in *2018 IEEE 24th international conference on parallel and distributed systems (ICPADS)*, IEEE, 2018, pp.604-611.

[22] A. Lingayat, R.R. Badre and A.K. Gupta, "Performance evaluation for deploying docker containers on bare metal and virtual machine," in *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*. IEEE, 2018, pp.1019-1023.

[23] S.B. Atreyapurapu, K. Amarendra and M.M. Alishah, "Hyperledger Fabric based Medical Record Security," in *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE, 2022, pp.223-228.

[24] F.L. Wang, Y.P. Ji, M.S. Liu, Y.Y. Li, X. Li, X. Zhang and X.J. Shi, "An Optimization Strategy for PBFT Consensus Mechanism Based on Consortium Blockchain," in *Proceedings of the 3rd ACM International Symposium on Blockchain and Secure Critical Infrastructure*, 2021, pp.71-76.

[25] Hyperledger Blockchain Performance Metrics, 2018, [Online]. Available online: https://www.hyperledger.org/wp-content/uploads/2018/10/HL_Whitepaper_Metrics_PDF_V1.01.pdf.