An Efficient Supply Chain Traceability Architecture Based on Blockchain and a Dynamic-Credit PBFT Consensus Algorithm

Dong Wang[1], Xiang Huang[2*], Zhanpeng Jiang[3], Mingyi Wang[4], Hongying Zhou[1]
[1]School of Economics and Management, Hezhou University, Hezhou 542899, China
[2] Cheung Kong School of Journalism and Communication, Shantou University, Shantou 515000, China
[3]School of Economics and Management, Guangdong Songshan Polytechnic, Shaoguan 512126, China
[4]Guangzhou Research Institute of Optical, Mechanical and Electronical Technologies Co., Ltd., Guangzhou 510000, China
E-mail: wangdongmax@foxmail.com, hx838383@126.com, gdfsjzp@sina.com, wangmingyi920515@outlook.com, m13870015586@163.com
[*]Corresponding author

Keywords: Blockchain, supply chain traceability, PBFT, consensus algorithm, system stability

Received: September 16, 2025

In the context of global economic integration, supply chains are becoming increasingly complex and volatile. Traditional traceability systems are difficult to meet the requirements of large-scale data processing, tamper resistance, and scalability. To this end, a supply chain traceability system based on blockchain and improved Practical Byzantine Fault Tolerance consensus algorithm is proposed. The system optimizes the consensus process by introducing credit and voting mechanisms, improving the model performance under high-frequency trading and malicious attacks. Meanwhile, a decentralized architecture combining the blockchain technology and an improved PBFT algorithm for fast consistency model is designed. A supply chain traceability mechanism including data layer, network layer, consensus layer, contract layer, and view layer is constructed. The experiment was implemented on the Ethereum simulation platform and tested on 30 nodes, with metrics including throughput, transaction latency, view switching time, etc. Compared with GC-PBFT and LC-PBFT, IPBFT had an average throughput increase of 8.5%, transaction latency reduction of 12.7%, and fault tolerance increase of 5.3%. In terms of cost-effectiveness, the annual operation and maintenance cost of the proposed supply chain traceability system was about 60,000 RMB, and the upgrade and expansion cost was about 120,000 RMB, which was significantly reduced compared to traditional systems. The product recall rate decreased from 5.2% to 2.1%, which not only improved economic efficiency, but also enhanced system reliability. The research results provide new technological solutions for transparent management and traceability construction of complex supply chains, which have significant theoretical value and practical application significance.

Povzetek: Študija predlaga verigo blokov z izboljšanim PBFT za sledljivost oskrbovalnih verig, z večplastno arhitekturo in preizkusi na Ethereumu, ki prinesejo višji pretok, nižjo latenco, večjo odpornost ter nižje stroške in manj odpoklicev.

1 Introduction

With the increasing complexity of global supply chains, supply chain traceability has become an important means to ensure product quality and safety [1]. However, traditional traceability systems face many challenges in dealing with large-scale data management, data security, and system scalability. Especially when facing high-frequency transactions, the consensus mechanism of traditional systems is inefficient and difficult to meet the needs of modern supply chains. In recent years, blockchain technology has gradually become an ideal solution for supply chain traceability due to its decentralization, transparency, and data immutability [2]. Existing blockchain systems still face some challenges in handling supply chain traceability, especially on the efficiency and security [3]. Blockchain technology has decentralization, tamper proof data, transparency, and traceability, providing new technical support for supply chain traceability [4]. However, existing blockchain systems still encounter many technical bottlenecks in practical applications. Especially when dealing with high-frequency transactions in large-scale supply chain networks, the efficiency and security issues of consensus mechanisms are particularly prominent. The communication complexity of consensus mechanism is high, with malicious node interference and a lack of dynamic adaptation mechanism. These issues limit the widespread application of existing traceability systems in actual supply chains and urgently require improvement. Traditional blockchain consensus mechanisms, such as Proof of Work (PoW) and Proof of Stake (PoS), suffer from high energy consumption and low efficiency, making it difficult to meet the high efficiency and low consumption needs of modern supply chains. As one of the mainstream consensus algorithms, Practical Byzantine

Fault Tolerance (PBFT) consensus algorithm is extensively used in blockchain systems due to its low energy consumption and high throughput [5]. The traditional PBFT algorithm can maintain consensus in cases where some nodes are malicious, but it exposes limitations such as high communication complexity and insufficient node selection mechanisms in high concurrency or malicious attacks. Therefore, the study introduces a dynamic credit mechanism to improve PBFT and proposes a supply chain traceability system based on blockchain and IPBFT consensus. This algorithm significantly reduces communication overhead and improves system robustness and scalability through reputation weighted voting and adaptive master node election.

2 Literature review

Blockchain can upload real-time data information from various links in the supply chain, ensuring the authenticity and reliability of transaction records. Wu et al. optimized record search using bipartite graph and maximum matching algorithm, reducing time overhead by 85.1% [6]. Chen et al. explored the traceability strategies of competing manufacturers and demonstrated that joining a highly traceable third-party blockchain platform could increase profits for all parties involved [7]. Berkani et al. revealed the potential of blockchain in the sports industry and the current situation of prototype deficiency [8]. Tan et al. implemented a validated cargo tracking solution in the logistics field through the Hyperledger prototype system [9]. Privacy protection and data security are also important research directions in blockchain applications outside the supply chain. For example, Feng et al. designed the Internet of Vehicles for people Privacy-Preserving Blockchain-Based Authentication with Global-Updated Commitment (PBAG). The privacy authentication protocol reduced the verification cost of a single message to 0.36ms through a global update commitment mechanism, which was 63.7% lower than that of existing solutions [10]. He et al. proposed an incentive mechanism based on game theory for federated learning security in energy blockchain, which effectively defended against data poisoning and committee deception, and combined high robustness and low complexity in distributed energy environments [11]. Zhou et al. analyzed a differential game model that food supply chain enterprises could optimize traceability strategies based on information latency time, thereby improving food safety and commercial profits [12]. This series of studies not only demonstrates the multi-scenario application potential of blockchain technology, but also provides reference for security issues in the supply chain environment.

In blockchain, consensus algorithm is the core component for managing trust between participants in blockchain activities. As one of the mainstream consensus methods, PBFT has low energy consumption and high throughput [13]. However, traditional PBFT algorithms still have significant limitations in scalability and performance, and many studies are dedicated to optimizing traditional PBFT. Qin et al. proposed a weighted Byzantine fault-tolerant algorithm, which suppressed malicious nodes through a dynamic weighting mechanism and performed well in throughput, latency, and security [14]. Li et al. designed an adaptive PBFT framework based on multi-agent and deep reinforcement learning, which achieved efficient consensus in environments with incomplete information [15]. Ma et al. provided a universal performance evaluation method for various consensus mechanisms by establishing a stochastic performance model and analyzing the Quasi Bird and Path (QBD) process [16]. In terms of scalability optimization, Wang proposed a committee election mechanism based on bidirectional decay of credit values, which effectively improved node quality [17]. Li et al. improved the consensus efficiency in large-scale networks through a hybrid mechanism of group Raft and committee PBFT [18]. The Threshold Proxy PBFT (TP-PBFT) protocol proposed by Tang et al. combined threshold proxy signatures with two-step clustering to maintain stable system operation even when a large number of nodes were offline [19]. The above research has optimized the PBFT algorithm from different dimensions, providing theoretical and practical support for efficient consensus in supply chain blockchain systems. In addition to traditional PBFT optimization research, several improvements have recently been proposed and widely adopted as benchmarks for evaluating blockchain consensus performance, including Group Classification Practical Byzantine Fault Tolerance (GC-PBFT) [20], Layered Cross-Chain Practical Byzantine Fault Tolerance (LC-PBFT) [21], TP-PBFT, and Reputation-based PBFT algorithm with node grouping strategy (RGPBFT) [22]. These schemes respectively enhance node classification, cross-chain scalability, and signature security, forming representative directions for PBFT enhancement. Table 1 presents a comparative summary of these algorithms and the proposed IPBFT.

In Table 1, these representative PBFT variants have not only been widely discussed in the literature, but also used as benchmark models in the results analysis section to evaluate the performance of the proposed IPBFT. In summary, although current research has attempted to introduce weighted voting or view switching mechanisms, there are still three limitations: Firstly, the weight settings are mostly static or globally fixed, making it difficult to adaptively adjust with changes in node behavior or fluctuations in attack intensity, which can easily lead to selection bias and reduced system stability. Secondly, view switching typically relies on network wide broadcasting and centralized triggering, resulting in high recovery costs and limited availability in high concurrency or high failure rate scenarios. Finally, weight design and view switching are often optimized separately, lacking collaborative analysis and sensitivity testing of security, scalability, and communication complexity within the same framework. Therefore, although existing solutions can partially improve the single point bottleneck of PBFT, they have not achieved the organic fusion of dynamic reputation evolution and local adaptive view switching, and it is difficult to effectively reduce message complexity while ensuring security. Therefore, a supply chain traceability system based on IPBFT is established. By introducing dynamic weighted credit mechanism and voting mechanism, the shortcomings of traditional PBFT in communication complexity and node selection optimization are compensated. Combined with the decentralized characteristics of blockchain technology, the supply chain traceability process is optimized. The research aims to explore how to optimize the efficiency and security of supply chain traceability systems using IPBFT, and provide an economically viable traceability solution for the industry.

Table 1: Comparative analysis of improved PBFT algorithms.

Reference	Algorithm	Main mechanism	Main limitation
Reference [20]	GC-PBFT	Group classification for node credit	Group partition overhead
Reference [21]	LC-PBFT	Layered cross-chain consensus	Complex coordination
Reference [19]	TP-PBFT	Threshold proxy signature	Requires additional signature scheme
Reference [22]	RGPBFT	Reputation-governed PBFT	Reputation convergence & cold-start; parameter tuning
/	IPBFT (Proposed)	Dynamic weighted credit + voting + view optimization	Needs further validation in large-scale networks

3 Methods and materials

3.1 Optimization of PBFT consensus algorithm

PBFT is a fault tolerance consensus mechanism extensively applied in blockchain and distributed systems to ensure consistent decision-making even in malicious nodes. This algorithm aims to solve the Byzantine Generals problem, ensuring the security and reliability of distributed systems even in untrusted environments [23]. The PBFT algorithm achieves consensus through three stages: pre-preparation, preparation, and submission. In the pre-preparation stage, the master node receives client requests and broadcasts them to the backup node. Backup node records and forward messages. After collecting sufficient preparation messages, the node enters the preparation state to ensure that all honest nodes reach a consensus on the message. To reduce the risk of abnormal master nodes and communication overhead, this study proposes an improved IPBFT algorithm based on credit voting mechanism. This algorithm introduces a dynamic credit model to evaluate the credibility of nodes based on their historical behavior and real-time performance, and assigns weights and permissions accordingly. Nodes with good credit can participate in the election and voting of the master node, thereby optimizing the generation process of the master node. In the initial stage of the system, due to the lack of historical data, credit values are initialized based on observable indicators such as node network latency, providing a basis for subsequent consensus participation and weight allocation. The initial credit value is only calculated based on the node network status, as displayed in equation (1) [24].

$$V_0 = index = \begin{cases} \dfrac{100-l}{100-l_{\min}}, l < 100ms \\ -1 \quad , l \geq 100ms \end{cases} \tag{1}$$

In equation (1), V_0 represents the initial credit value of the system. l represents node network latency. $l_{\min}$ represents the minimum latency value in the node. $index$ represents the network latency index. After the consensus is completed, the system compares the consensus results with the confirmation messages sent by the nodes during the consensus process to decide the node integrity, as shown in equation (2).

$$compare = \begin{cases} 0, T_{result} \neq T_{commit} \\ 1, T_{result} = T_{commit} \end{cases} \tag{2}$$

In equation (2), $compare$ is a variable that depends on T_{result} and T_{commit}. T_{result} represents the consensus result. T_{commit} represents the confirmation information sent by each node in the consensus. When $compare$ is 0, the node has dishonest behavior or network transmission errors. When $compare$ is 1, the node behavior is honest. After combining $compare$ and $index$, the consensus credit value for this round is obtained, as shown in equation (3).

$$V_{once} = index + w(compare - index) \tag{3}$$

In equation (3), V_{once} represents the credit value of this round of consensus. w represents the weight of $compare$ during V_{once} normalization. After completing n rounds of consensus, when the node participates in consensus for the first time, i.e. $n=1$, the current credit value of the node is shown in equation (4).

$$V_{latest} = 0.8V_{once} + 0.2V_0 \tag{4}$$

In equation (4), V_{latest} represents the latest credit value of the node after the most recent consensus. At this point, V_{latest} is composed of the weighted sum of V_{once} and V_0 in the current consensus. V_{once} accounts for 80% of the weight, while V_0 accounts for 20% of the weight. Decentralized design can quickly adapt to changes in node performance. When $n \geq 2$, V_{latest} is shown in equation (5).

$$V_{latest} = 0.8V_{once} + 0.2V_{late} \tag{5}$$

In equation (5), V_{late} represents the global credit value accumulated by the node in previous consensus. In this case, V_{latest} is composed of the weighted sum of

V_{once} in the current consensus and the previous global credit value V_{late}. V_{once} still accounts for 80% of the weight, while V_{late} accounts for 20% of the weight. To make the current consensus behavior dominant in the credit value, w takes the minimum value of 0.65 that is easy to calculate, resulting in equation (6).

$$V_{once} = 0.65 compare + 0.35 index \tag{6}$$

According to equation (6), the integrity behavior and network status of a node can be comprehensively evaluated to determine the credit score in a single consensus. The credit score classification can be performed. The first method is shown in equation (7).

$$\begin{cases} V_{latest} > \sum_{i=0}^{n-1} 0.52 \times 0.2^{i} \\ \sum_{i=0}^{n-1} 0.52 \times 0.2^{i} > V_{latest} > \sum_{i=0}^{n-1} 0.52 \times 0.2^{i} + 0.28 \\ \sum_{i=1}^{n-1} 0.52 \times 0.2^{i} + 0.28 > V_{latest} \end{cases} \tag{7}$$

In the reputation update rule of equation (7), the first type implements rapid credit downgrade on Byzantine nodes to limit their impact. The second type adopts a mild adjustment and replacement mechanism for nodes with decreased network performance to ensure consensus continuity. The third type improves system robustness by dynamically weighting and switching views to replace failed nodes in a timely manner. In response to the inevitable Byzantine behavior, the system adopts a reputation weighted voting and dynamic view switching mechanism: The reputation mechanism marks malicious nodes as abandoned and removes them from the consensus, while selecting nodes to replace secondary nodes. The new voting algorithm combines reputation value and voting results to achieve adaptive master node election through state partitioning and permission allocation [25]. From a control perspective, reputation weighted voting is equivalent to adaptive feedback gain tuning, effectively suppressing disturbances and accelerating consensus convergence. View switching is similar to the rapid resynchronization process after a failure, shortening the system recovery time [26]. The low latency and fast recovery capability in attack scenarios can be attributed to the synergistic effect of "adaptive gain" and "local resynchronization", with the key being the balance between synchronization margin and anti-interference in parameter tuning. The final score of the candidate node is shown in equation (8).

$$S_n = V_n \cdot \lambda + (\sum\nolimits_{k=1}^{N} \frac{V_k}{150} \cdot vote_k)\mu \tag{8}$$

In equation (8), S_n represents the final score of the candidate node. V_n and V_k represent the credit values of candidate nodes n and k, respectively. $vote_k$ represents the voting of node k. λ and λ are weight coefficients, and $\lambda + \mu = 1$. Parameters λ and λ represent the weight coefficients of the node's own reputation factor and external voting feedback, respectively. To reasonably determine the values of both, parameter tuning and sensitivity analysis are conducted. In the initial setup, considering that the stability of node reputation in permissioned blockchain is higher than the randomness of voting feedback, the weights are set, and $\lambda > \mu$. Theoretical analysis shows that appropriately increasing λ helps maintain trust consistency, while retaining a moderate λ enhances the system's adaptability to Byzantine dynamic behavior. When $\lambda > 0.7$, the system adaptability decreases. When $\lambda < 0.5$, there is an increase in short-term misjudgments between nodes, leading to fluctuations in consensus efficiency. Therefore, $\lambda = 0.6$ and $\mu = 0.4$ are empirical optimal parameter configurations that strike a balance between stability and responsiveness. The PBFT algorithm achieves consensus through two stages. The master node sends requests to the backup node and verifies the information to form consensus. Subsequently, it is confirmed again to ensure that the majority of nodes have completed the verification. After verification, it indicates that enough backup nodes have confirmed the request from the primary node and reached consensus. During the confirmation phase, only the information verified during the preparation phase is reconfirmed to ensure that the number of valid nodes completes the request verification. Therefore, the optimized PBFT consensus process can be divided into two stages: consensus proposal and consensus confirmation. Figure 1 displays the IPBFT consensus process.

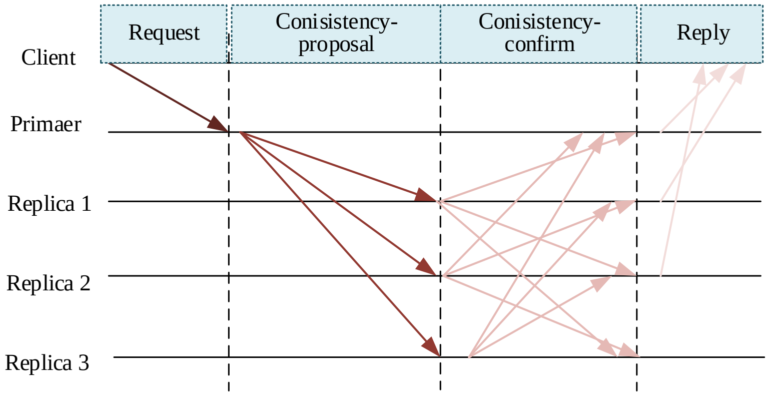


Figure 1: IPBFT consensus process

In Figure 1, even if the master node fails, the consensus system can continue to operate through the view switching protocol to prevent errors [27]. In the PBFT algorithm, once the backup node detects a master node failure or malicious behavior, it will broadcast a message and initiate the view switching process. The IPBFT algorithm classifies and processes view switching protocols based on the abnormal type of the master node. The pseudocode for IPBFT view switching protocol is shown in Algorithm 1.

Algorithm 1: View Change Protocol in IPBFT

```
Input: Timeout threshold T, Current view number v, Master node M
Output: Updated view and Master node
if M fails to respond within T then
    broadcast <ViewChange, v+1> to all nodes
    collect ViewChange messages from ≥ 2f+1 nodes
    select next Master = Master() based on highest Si
    broadcast <NewView, v+1, Master> to all nodes
    synchronize pending requests from old view
    execute consensus in new view
else
    continue normal consensus
```

In the view switching mechanism of IPBFT, the client sets a timeout threshold. If the master node times out, it may be due to system overload, malfunction, network issues, or intentional denial of service. This indicates that there is a problem with the master node. During the view switching process, whether the master node is abnormal is determined through voting. Once the master node times out, the system will enter the new view phase and select a new master node through the Master() algorithm [28]. The pseudocode for the Master() master node election function is shown in Algorithm 2.

Algorithm 2: View Change Protocol in IPBFT

```
Input: Node set N = {n1, n2, …, nk}, credit values Vn, votes voten
Output: Master node ID
for each node ni in N do
    compute score $Si = \lambda * Vn + \mu * (\Sigma(Vk / |N|) * votek)$
end for
sort nodes by Si in descending order
select node with max(Si) as Master
broadcast Master(ID) to all backup nodes
return Master
```

The new master node spread a new view message to continue processing unfinished transactions from the previous view. If the backup node believes that the master node is malicious, it will initiate a view switching process, including view switching, view switching-confirmation, and a new view phase. IPBFT classifies master node anomalies based on PBFT, reducing communication in timeout situations and improving efficiency. Figure 2 illustrates the final IPBFT algorithm flow.

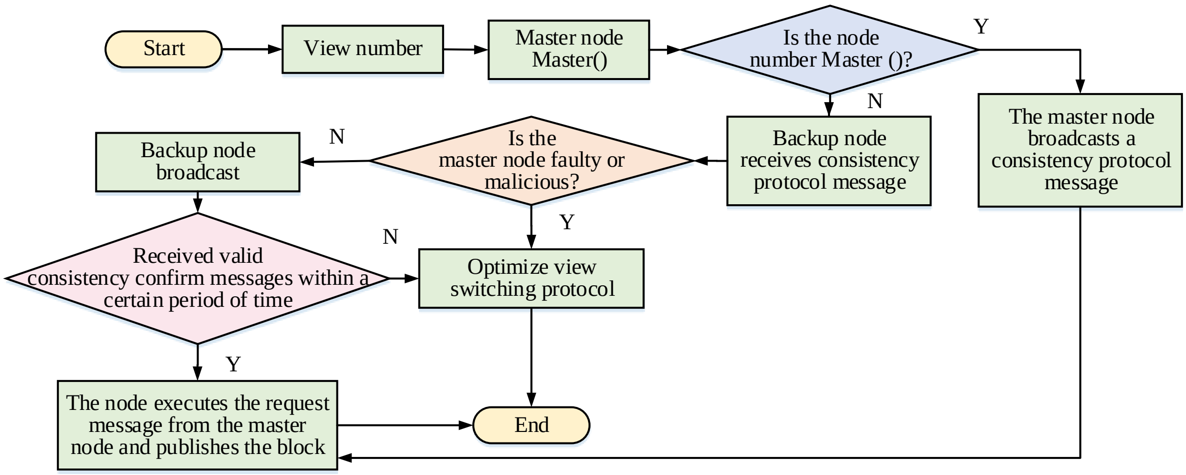

Start
View number
Master node Master()
Is the node number Master ()?
Y
N
The master node broadcasts a consistency protocol message
Backup node receives consistency protocol message
Is the master node faulty or malicious?
N
Y
Backup node broadcast
Received valid consistency confirm messages within a certain period of time
N
Y
Optimize view switching protocol
The node executes the request message from the master node and publishes the block
End

Figure 2: IPBFT algorithm process.

In Figure 2, the view number is initialized. The master node is set to Master() to determine whether the current node is the master node. If it is the master node, protocol messages are broadcasted consistently. Otherwise, the backup node receives consistency protocol messages and broadcasts them. Subsequently, any faults or malicious behavior on the master node are checked. If so, the view switching protocol has been optimized. On the contrary, valid consistency confirmation messages are received within a limited time. If consensus is reached, the node needs to execute the request message from the master node and publish the block. If no consensus is reached, the process is returned to wait and re-execute until the entire consensus process is completed.

3.2 Design of supply chain traceability mechanism based on blockchain and IPBFT consensus algorithm

To ensure the efficient operation of blockchain networks and the smooth execution of consensus algorithms, this study constructs an efficient, secure, and transparent supply chain traceability system. This system combines the decentralized characteristics of blockchain technology with the fast consistency of the IPBFT consensus algorithm. It can effectively record and verify various transactions and information flows in the supply chain, guaranteeing the authenticity and immutability of data. In the traceability system, from the perspective of system application, the block height increases correspondingly with each data on chain operation. Based on the development platform, the complete process of generating blocks is divided into four stages, including transaction initiation stage, block consensus stage, block verification stage, and block execution stage. Figure 3 illustrates the blockchain application process [29].

As shown in Figure 3, during the transaction initiation phase, a user initiates a transaction and a new message is broadcasted to the blockchain network. After entering the block consensus stage, the system uses the consensus algorithm of the traceability mechanism to consensus the previously initiated transaction information. In the block verification phase, the system schedules transactions and triggers the smart contract after reaching a consensus through the smart contract of the traceability system to verify the information accuracy. Finally, during the block execution phase, the blocks that have reached consensus and passed verification are submitted to the ledger for storage. In a blockchain system using the IPBFT algorithm, the master node is responsible for generating blocks and does not rely on incentive mechanisms. Therefore, the mining reward is not included in the block data. The system block only needs to contain the parent node, digest, and basic information, which not only reduces the storage requirements of the block, but also lowers the data consumption during transmission. On this basis, to enhance the application effect of blockchain in supply chain traceability, a five layer model is designed, including data layer, network layer, consensus layer, contract layer, and view layer. The model adopts a consortium chain form. The consensus algorithm is docked between the data layer and the contract layer. Figure 4 illustrates the architecture of the supply chain traceability system.

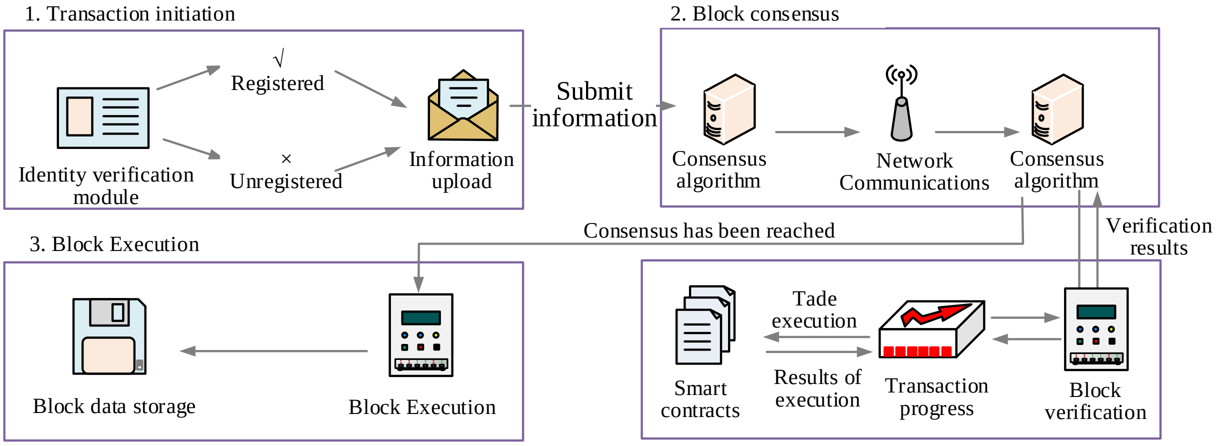


Figure 3: Blockchain application process

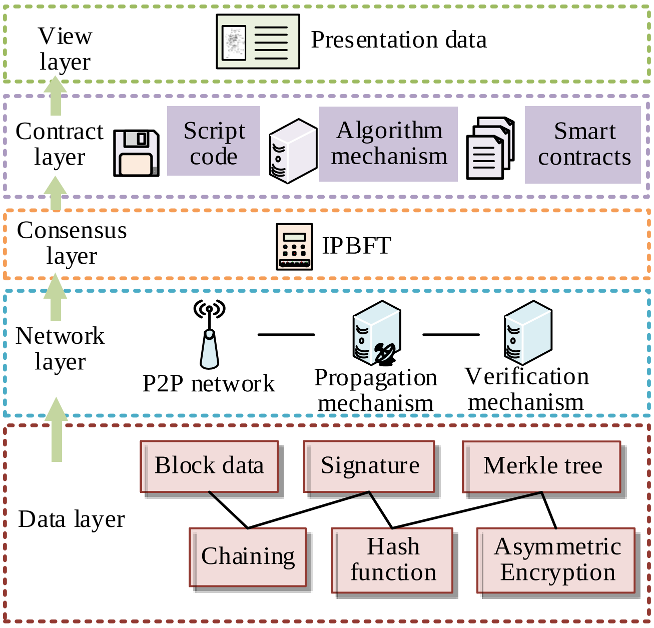


Figure 4: Architecture of supply chain traceability system

In Figure 4, the data layer uses Merkle trees and blockchain to store supply chain data, ensuring data security through digital signatures, hash functions, and asymmetric encryption [30]. A P2P distributed network is established at the network layer, responsible for node communication and block verification. The consensus layer adopts the IPBFT algorithm, which optimizes the consensus efficiency of traditional PoW and PoS algorithms [31, 32]. The contract layer deploys smart contracts for manufacturers, distributors, and retailers, recording digital signatures during transaction execution to achieve full traceability. The view layer provides a user interaction interface through a Graphical User Interface (GUI) [33]. The functions of smart contracts are mainly used for data processing, verification, and permission management, and do not directly participate in the execution of consensus logic. The IPBFT consensus process runs entirely within the consensus layer, with external engines implementing node communication, credit evaluation, and view switching operations. Figure 5 illustrates the final supply chain traceability system based on blockchain and IPBFT consensus algorithm.

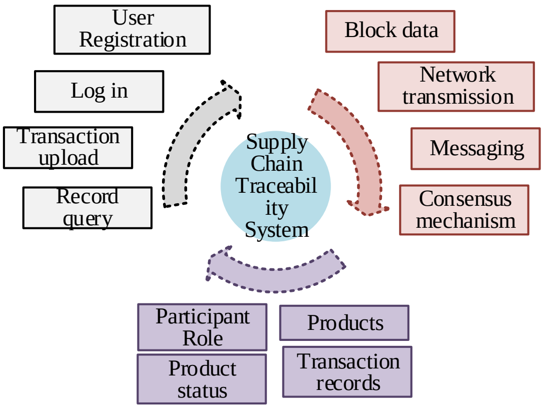


Figure 5: Supply chain traceability system

The system in Figure 5 consists of three core components: user interface module, blockchain coordination module, and smart contract processing module. The blockchain coordination module mainly handles information exchange and data upload between nodes. The underlying architecture of the system employs the Ethereum blockchain platform, deploying blockchain nodes with smart contract support on the Ethereum platform to build a decentralized network environment. The Ethereum client and IPBFT consensus module are integrated to ensure compatibility with existing blockchain architectures [34]. IPBFT adopts three-stage message passing to ensure block finality, and utilizes Ethereum smart contracts to manage node voting and verification processes, recording the entire consensus process. This system combines dynamic node management to maintain high fault tolerance in the event of network anomalies or attacks, while suppressing malicious behavior and incentivizing node participation through reward and punishment mechanisms. The system uses smart contracts to implement tamper proof contract publishing and role behavior constraints, reaching consensus through IPBFT. The user interaction module supports role registration, login, and transaction record upload queries. Among the three types of participants: manufacturers, distributors, and end-users, manufacturers and distributors need to undergo registration and review before uploading transactions. Manufacturer data is directly uploaded to the chain, and distributor data needs to be verified before uploading. All consensus data is stored in a database and queried through an open Application Programming Interface (API) [35]. Users can obtain transaction history sorted by time through the product number, or view detailed information in the blockchain browser through the product specific contract address.

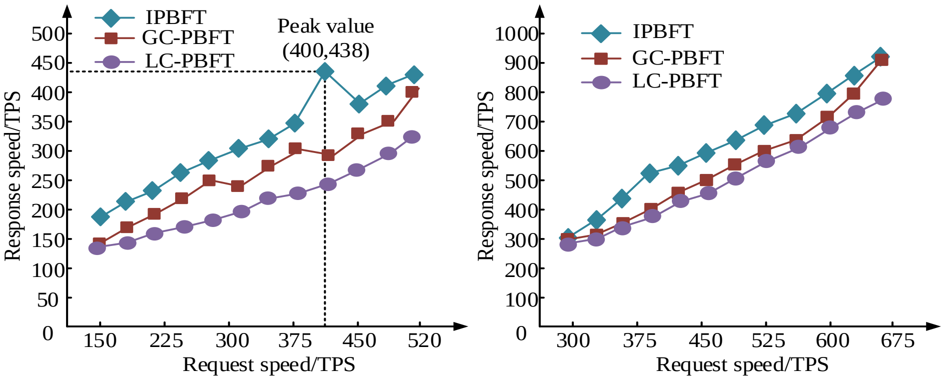
4 Results

4.1 IPBFT performance testing

To ensure the reproducibility of the experiment and comprehensively verify the robustness and anti attack performance of the proposed IPBFT algorithm, a controlled simulation environment was constructed. The blockchain network had a total of 30 nodes, of which 24 were normal nodes and 6 were Byzantine nodes, accounting for approximately 20%. The behavior of Byzantine nodes was simulated in three ways: One was message tampering, where nodes broadcast inconsistent block hashes or transaction results to different nodes to disrupt the consensus process. The second was packet loss behavior, where nodes selectively discarded some consensus messages, thereby increasing communication latency and reducing throughput. The third was latency injection, where nodes actively introduced the random communication latency to simulate unstable scenarios such as network congestion or denial of service attacks. The experimental simulation was based on the Ethereum platform and integrated with a custom IPBFT consensus engine. It was developed using Python 3.10 and PyTorch 2.1 language environments, and the underlying communication framework employed the Go Ethereum client. The smart contract module was deployed in a private Ethereum testing network, written in Solidity language, responsible for transaction recording and data verification. The consensus layer and message scheduling module ran in a Python parallel simulator to achieve synchronization and message interaction between nodes. To avoid accidental effects, each experiment was independently run 10 times under the same system configuration, and the resulting throughput, latency, response time, and communication volume were taken as arithmetic averages. The experimental running hardware environment is an Intel Core i5-9400F processor, 8GB DDR4 memory, and Ubuntu 16.04 operating system, which is consistent with the typical blockchain performance evaluation environment. All performance test results are based on independently running 10 simulations with the same system configuration, and the data obtained is the average of multiple experiments to eliminate possible random fluctuations in a single run. The variance of the experimental results is controlled within ± 3.5%, indicating that the system has good stability and repeatability under multiple rounds of testing. The

Byzantine fault testing is conducted to evaluate the performance changes under various conditions. The proposed IPBFT is compared and tested with GC-PBFT and LC-PBFT. When there are no node failures or attacks from Byzantine malicious nodes, the response speed of these three methods to Invoke and Query operations is shown in Figure 6.

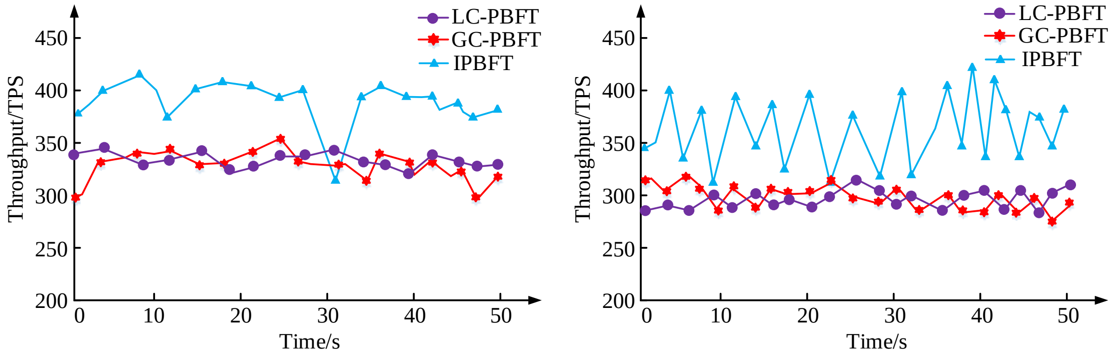
Figures 6 (a) and 6 (b) show the response speed changes of different algorithms in Invoke and Query operations, respectively. Overall, the data throughput continuously increased with the increase of operation requests. In Figure 6 (a), IPBFT always had the highest response speed as the request speed increased, reaching a peak of 438 at 400TPS. The response speed of GC-PBFT was lower than that of IPBFT, but it performed better than that of LC-PBFT. LC-PBFT had the lowest response speed among these three algorithms, indicating relatively poor performance. In Figure 6 (b), IPBFT exhibited the highest response speed at all request speeds. The response speed of LC-PBFT was still the lowest, indicating its lowest efficiency in processing requests. The dynamic weighted credit model and optimized view switching protocol in IPBFT reduce the computational complexity in the consensus process, enabling more reliable nodes to process requests more efficiently. Subsequently, whether there are random node failures and Byzantine malicious attacks is experimentally evaluated. The data throughput of Invoke operation is used as a metric, as shown in Figure 7.



(a) Invoke

(b) Query

Figure 6: The response speed of three algorithms to different operations

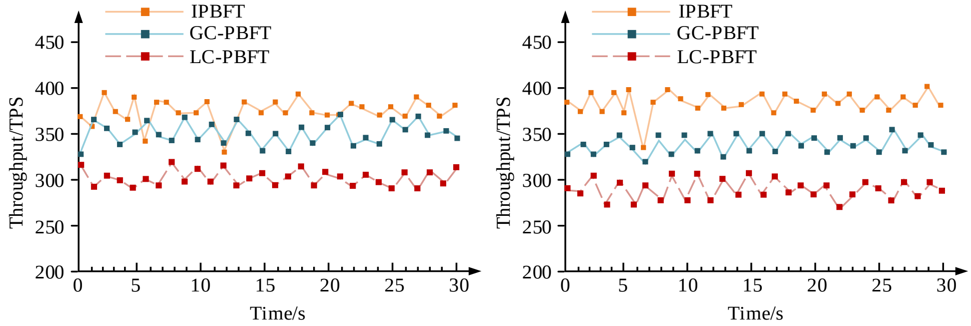


(a) Low Failure Probability

(b) High Failure Probability

Figure 7: Consensus efficiency changes over time under random faults

Figure 7 (a) shows the performance of throughput over time for LC-PBFT, GC-PBFT, and IPBFT under low failure probability conditions. The throughput of the IPBFT algorithm remains relatively stable, and its overall performance is better than LC-PBFT and GC-PBFT, but it does not show whether it is affected by node failures. After about 30s, the IPBFT algorithm showed that the consensus speed of the system slowed down due to node failures. Even after the speed decreased, the instantaneous processing capability of the system remained at normal operating efficiency, and the stability of the system was not affected. Figure 7 (b) shows the variation of data throughput over time for three algorithms under high failure probability. As the probability of failure continued to increase, the throughput of LC-PBFT and GC-PBFT remained at a relatively high level with relatively small fluctuations. The impact of IPBFT on data transmission speed was significant. Although this algorithm consumes a lot of resources in node replacement and view switching, it can cause performance fluctuations within a certain range. Overall, the efficiency of the IPBFT algorithm is still superior to the other two comparison algorithms, further demonstrating the stability advantage of the dynamic weighting mechanism and fault-tolerant protocol in fault prone environments. Finally, no node failures and continuous attacks from Byzantine malicious nodes are analyzed, and the Invoke operation is also selected. Figure 8 illustrates the data throughput results.



(a) Low probability of malicious attacks (b) High probability of malicious attacks

Figure 8: Consensus efficiency changes over time under random Byzantine attacks

Figure 8 illustrates consensus efficiency under random Byzantine attacks. In Figure 8(a), under low attack probability, due to malicious node replacement, the efficiency of IPBFT has doubled, but it quickly recovered, while other algorithms have not changed, indicating that they cannot detect attacks. In Figure 8(b), as attack probability increases, LC-PBFT and GC-PBFT experience severe efficiency decline and fluctuation, whereas IPBFT maintains higher and more stable throughput after brief interference. The results indicate that the dynamic credit model enables IPBFT to identify and isolate malicious nodes, minimizing the impact of attacks. The total communication volume of different algorithms in a single transaction is tested, as shown in Figure 9.

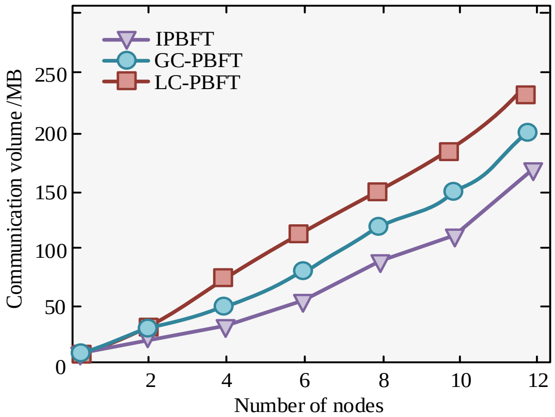


Figure 9: The total communication volume of different algorithms in a single transaction.

In Figure 9, overall, with the increase of nodes, the communication volume of the three methods showed an upward trend. The growth of communication volume was roughly linear, but the growth rate varied among different algorithms. As the node increased, the communication volume of LC-PBFT also increased, which may affect the overall communication efficiency. The communication volume of GC-PBFT was between IPBFT and LC-PBFT, and the communication overhead was higher than that of IPBFT. As the number of nodes increases, IPBFT shows slower growth in communication volume, demonstrating better scalability and communication efficiency. The communication volume of IPBFT is the lowest among these three algorithms. This indicates that IPBFT optimizes communication overhead by reducing redundant interactions in the consensus process. To verify the performance advantages of the IPBFT view switching mechanism, four algorithms, PBFT, GC-PBFT, LC-PBFT, and IPBFT, were compared in terms of view change time under different failure probabilities, as shown in Table 2.

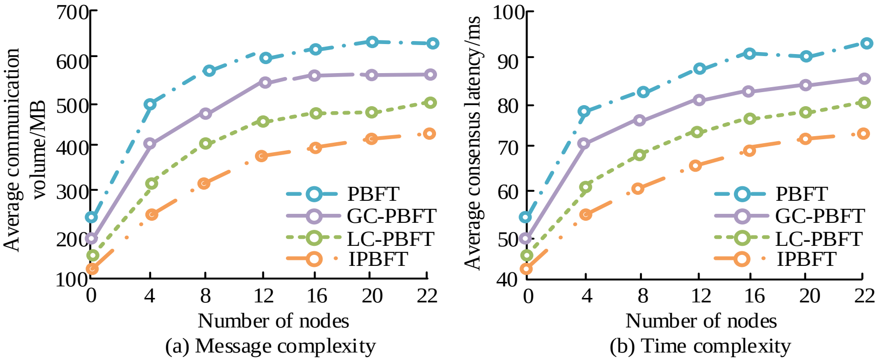
Table 2: Performance under view change time under different failure probabilities.

Scenario	Indicator	PBFT	GC-PBFT	LC-PBFT	IPBFT
Low-failure rate	Average time consumption/s	1.82	1.36	1.22	0.97
	Transaction throughput rate/TPS	365	412	398	438
	Average transaction latency/ms	145	132	128	115
High-failure rate	Average time consumption/s	3.45	2.91	2.63	1.88
	Transaction throughput rate/TPS	322	365	348	403
	Average transaction latency/ms	192	171	163	139

Table 2 showed that IPBFT outperformed PBFT, GC-PBFT, and LC-PBFT in both low- and high-failure environments, with lower time consumption, higher throughput, and shorter latency, demonstrating superior stability and robustness. Under low-failure rates, the average latency of PBFT was 1.82 seconds, 365 TPS, and 145 ms. GC-PBFT and LC-PBFT improved slightly through group validation and hierarchical topology. The average time of IPBFT was 0.97 seconds, which was 46.7% lower than that of PBFT, with a 20.7% reduction in latency and a 20% increase in throughput, indicating higher node collaboration efficiency. At high-failure rates, the time of PBFT increased to 3.45 seconds and the throughput decreased to 322 TPS, while IPBFT remained at 1.88 seconds and 403 TPS. The view switching efficiency increased by 45.5% and the latency decreased by 27.6%.

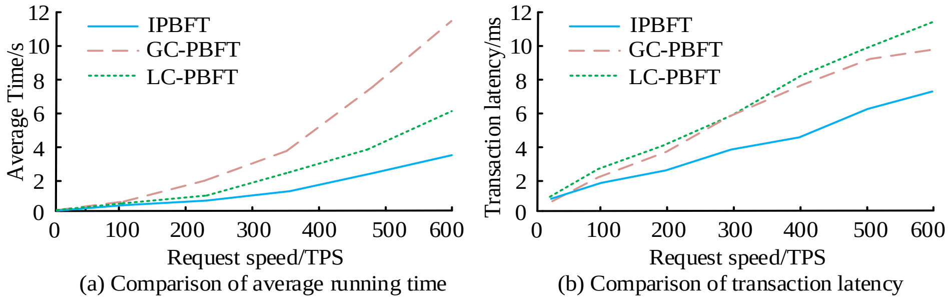
This confirms that IPBFT has strong fault tolerance and fast recovery capability through local broadcasting and isolation of low reputation nodes. Figure 10 presents the changes in message complexity for different algorithms.

In Figure 10 (a), as the number of nodes increases, the communication volume of each algorithm also increases, but the growth patterns differ significantly. The communication volume of PBFT increased from 250 MB to 640 MB. GC-PBFT used a packet broadcast mechanism, resulting in a final communication volume of approximately 550 MB. LC-PBFT adopted a hierarchical communication structure, further reducing the communication volume to 490 MB. In contrast, IPBFT had the slowest growth in communication volume, only increasing from 120 MB to 410 MB, showing an approximately linear growth trend. IPBFT effectively reduces message exchange redundancy through reputation filtering and local broadcast mechanism, and can maintain low communication load even when expanding in scale. In Figure 10 (b), the latency of PBFT increased from 55 ms at 4 nodes to 94 s at 22 nodes, indicating that its multi-node performance is significantly constrained by communication and synchronization. GC-PBFT and LC-PBFT achieved hierarchical and parallel optimization, with final latency controlled at 83 s and 79 s, respectively. IPBFT performed the best, with an average latency of only 71 s and a smooth growth curve, indicating that its dynamic reputation mechanism and asynchronous view switching strategy effectively alleviated the synchronization pressure caused by node expansion.



(a) Message complexity (b) Time complexity

Figure 10: Changes in message complexity of different algorithms.



(a) Comparison of average running time (b) Comparison of transaction latency

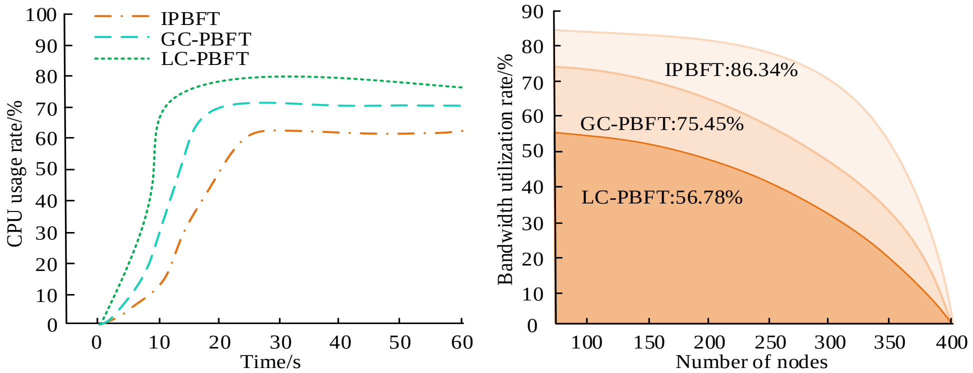
Figure 11: System average running time and time latency results.

4.2 Effect analysis of supply chain traceability system based on blockchain and IPBFT consensus algorithm

The study evaluates the efficiency of the designed system through actual operational testing to validate the practicality of the traceability system. Firstly, the blockchain technology is deployed on the virtual machine. The blockchain network is launched to conduct a series of tests on system runtime, transaction latency, and processing capabilities. After multiple tests, the system latency results are shown in Figure 11.

Figure 11 (a) showed that the average running time of IPBFT slowly increased with the increase of request speed, maintaining good efficiency, while GC-PBFT showed the fastest performance improvement above 300 TPS, with a sharp decline. Figure 11 (b) showed that the transaction latency of IPBFT was consistently lower than that of GC-PBFT and LC-PBFT, steadily but moderately increasing with load. In contrast, the latency of GC-PBFT increased rapidly above 400 TPS, while the latency of LC-PBFT was slightly lower but still high under heavy loads. Overall, IPBFT effectively reduces running time and latency under high loads by prioritizing reliable nodes through its credit model and minimizing latency during node changes via the view-switching protocol. Figure 12 displays the resource consumption of the platform in processing transactions under both high and low probability Byzantine malicious attacks.

Figure 12 (a) displays the CPU usage of the system based on different algorithms during operation. The CPU usage of GC-PBFT rapidly increased to nearly 80% in the first 10s, and then slowly increased to about 85%. The algorithm had a high demand for CPU resources. The CPU usage of LC-PBFT sharply increased to around 70% in the first 10s and eventually stabilized at around 75%. The CPU usage of IPBFT rapidly increased to about 60% in the first 10s and gradually stabilized, eventually stabilizing at around 60%. The low CPU usage of IPBFT indicates that the algorithm requires less CPU resources. Figure 12 (b) shows the bandwidth utilization of supply chain traceability systems based on different algorithms during operation. The bandwidth utilization rates of IPBFT, GC-PBFT, and LC-PBFT were 86.34%, 75.45%, and 56.78%, respectively. The bandwidth utilization of IPBFT gradually decreased with the increase of the node, but still maintained a high level and performed well in handling large amounts of data transmission. To further verify the applicability and comprehensive performance advantages of the proposed IPBFT algorithm under different blockchain architectures, RGBBFT and Polygon blockchain-based strategies were further introduced in the study [36]. Taking speed, security, scalability, and cost as indicators, the results are shown in Table 3.



(a) CPU usage rate (b) Bandwidth Utilization

Figure 12: Resource consumption results of different algorithms.

Table 3: Comprehensive performance comparison.

Algorithm	Throughput (TPS)	Byzantine fault tolerance (%)	Node capacity (piece)	Average cost (RMB)
PBFT	365	90.2	35	72,500
GC-PBFT	412	92.7	52	68,400
LC-PBFT	398	93.1	55	64,800
IPBFT	438	95.1	77	60,300
RGBBFT	421	94.3	64	62,800
Polygon blockchain	452	93.8	127	55,600

Table 3 showed that traditional PBFT faces a trade-off between speed, security, and scalability, with node capacity limited to 35 due to high communication complexity. GC-PBFT and LC-PBFT improved speed and stability through grouping and hierarchy but remained constrained in scalability. The proposed IPBFT, enhanced with dynamic reputation evaluation and asynchronous view switching, achieved 438 TPS throughput, 95.1% fault tolerance, scalability up to 77 nodes, and 16.9% lower maintenance costs, offering the most balanced performance. RGBBFT excels in trust and security through reputation governance, while Polygon's two-layer sidechain design provides excellent scalability and energy efficiency, supporting 127 nodes at the lowest cost, making it an ideal choice for large-scale, high concurrency supply chain networks. Finally, a simulation platform is established to apply the proposed supply chain traceability mechanism to a certain aquatic product supply chain. The cost-benefit analysis is compared with the supply chain traceability system that does not use the proposed mechanism. Table 4 displays the results.

Table 4 showed that both systems require an initial investment of 300,000 RMB. However, traditional systems had poor scalability, with annual maintenance costs of about 80,000 RMB and upgrade costs of 140,000 RMB. The proposed system, with higher automation and scalability, reduced these to about 60,000 and 120,000 RMB, respectively. Its product recall rate was only 2.1%, compared to 5.2% for traditional systems, greatly improving reliability and economic efficiency. The product data in Table 4 was based on simulated estimates, not real enterprise data. Cost is calculated theoretically based on market prices and experimental assumptions, while recall rate is determined based on the proportion of simulation errors in the traceability process.

Table 4: Cost benefit comparison of supply chain.

Traceability system	Initial investment cost (RMB)	Operation and maintenance costs (RMB/year)	Upgrade and expansion costs (RMB)	Product recall rate (%)
Traditional traceability system	300,000	80,000	140,000	5.2
The designed traceability system	300,000	60,000	120,000	2.1

5 Discussion

A blockchain-based supply chain traceability mechanism with an improved PBFT consensus is proposed to enhance product traceability and accountability. By introducing a dynamic weighted credit and voting model, it overcomes the performance and security limitations of traditional PBFT, resolving bottlenecks under high-frequency transactions, node failures, and Byzantine attacks. Experiments show that the IPBFT algorithm achieves optimal response and throughput at varying request rates. Under low-probability Byzantine attacks, despite brief efficiency drops, the system quickly restores stability through node replacement. As node scale increases, IPBFT shows the slowest communication growth and lower latency compared to other algorithms, demonstrating strong scalability. Even under high-probability attacks, system efficiency rapidly recovers after isolating malicious nodes. CPU utilization stabilizes around 60%, and bandwidth efficiency remains high as nodes increase. Economically, the system has maintained low annual maintenance and expansion costs, with a product recall rate of 2.1%, far lower than the 5.2% of traditional systems, improving reliability and cost-effectiveness. Compared to the group reputation clustering of GC-PBFT and the hierarchical consensus model of LC-PBFT, the innovation of IPBFT mainly lies in two aspects: One is the bidirectional dynamic reputation update mechanism. The credit attenuation parameter of IPBFT is not a preset constant, but varies proportionally with the transaction density and anomaly rate of nodes. When a node is in a high reputation state for a long time, the latency rate slows down to maintain system stability. Under continuous low reputation behavior, the attenuation coefficient rapidly increases to accelerate the isolation of malicious nodes. The second is the collaborative protocol for switching partial views. Traditional PBFT requires network wide broadcast triggering switching when the master node is abnormal, while IPBFT achieves local autonomous switching through reputation weighted selection, improving scalability and stability in large-scale node environments.

Although the IPBFT-based blockchain traceability system performs well under controlled attack scenarios, several challenges remain in real-world applications. Firstly, the system's scalability requires further verification. Current experiments involve only 30 nodes. Expanding to cross-regional networks with hundreds or thousands of nodes may sharply increase communication complexity and synchronization latency, necessitating optimization via message compression or hierarchical consensus. Secondly, integration with existing enterprise systems incurs high costs due to interface development and protocol adaptation, while regional differences in data privacy, regulation, and authentication hinder large-scale deployment. Finally, current security tests cover only moderate Byzantine and denial-of-service attacks, without addressing extreme cases such as multi-node collusion or network intrusions. Future work should strengthen robustness testing under such conditions. In terms of cost-effectiveness, although the aquatic product supply chain case shows that the system effectively reduces recall rates, lowers maintenance costs, and improves transparency, its cross-industry applicability still requires verification. Industries such as pharmaceuticals and food demand higher real-time traceability, compliance, and privacy protection, while electronic manufacturing is more sensitive to scalability and consensus latency. Thus, despite the system's general advantages in transparency, efficiency, and security, variations in regulation, data scale, and network complexity may affect deployment parameters and economic benefits. Future work should extend cost-benefit evaluations across industries and optimize adaptive designs for different application contexts.

6 Conclusion

The proposed IPBFT algorithm has shown significant performance advantages in multi-scenario experiments. Compared with GC-PBFT and LC-PBFT, IPBFT has a throughput increase of about 7%-10%, an average latency reduction of about 12%-18%, and a system availability improvement of over 15% under high attack probability. The dynamic weighted credit model quickly identifies and isolates malicious nodes by real-time evaluation of node behavior and feedback consistency. The parallel optimization of the view switching mechanism effectively reduces communication complexity and improves the scalability of the system under high concurrency conditions. Overall, IPBFT not only achieves efficiency and security optimization of traditional PBFT at the algorithm level, but also demonstrates strong universality and scalability potential in industrial applications. Its high concurrency, low latency, and robustness make it suitable for deployment in multiple industry scenarios such as food safety supervision, pharmaceutical anti-counterfeiting tracking, and high-value product authentication. It can support trusted traceability and cross-domain collaboration throughout the entire supply chain lifecycle, further verifying the applicability and promotional value of the proposed model in complex real-world systems. Future work will focus on cross-industry deployment and validation in large-scale node environments, further improving the adaptive scheduling and economic incentive mechanisms of algorithms to achieve a scalable and trustworthy blockchain traceability system.

Although the blockchain traceability system based on IPBFT has good performance in controlled attack scenarios, there are still several limitations and challenges that need further research in practical applications. Firstly, the scalability of the system needs to be verified. The current experiment is only completed at a scale of 30 nodes. If it is expanded to a cross regional supply chain network with hundreds or even thousands of nodes, the communication complexity and synchronization latency may significantly increase, and optimization through message compression or hierarchical consensus mechanisms is needed. Secondly, the integration cost between the system and the existing information system of the enterprise is high, requiring additional development of interfaces and adaptation protocols. Meanwhile, policy differences in data privacy, regulatory compliance, and digital authentication across regions also pose challenges for large-scale promotion. Finally, the existing security verification only covers moderate intensity Byzantine attacks and denial of service attacks, and has not yet simulated extreme scenarios such as multi-node collaborative attacks or underlying network intrusions. In the future, it is necessary to further strengthen the security robustness testing of the system.

Acknowledgement

This study was supported by the 2024 National Social Science Foundation Project "Policy Optimization Research on Digital Rural Development and Prevention of Large-Scale Poverty Relapse in Ethnic Minority Areas of Southwest China" (Project Number: 24XMZ032).

References

[1] L. Dong, P. Jiang, and F. Xu. Impact of traceability technology adoption in food supply chain networks. Management Science, 69(3), 1518-1535, March, 2023, DOI: 10.1287/mnsc.2022.4440.

[2] G. M. Razak, L. C. Hendry, and M. Stevenson. Supply chain traceability: A review of the benefits and its relationship with supply chain resilience. Production Planning & Control, 34(11), 1114-1134, August, 2023, DOI: 10.1080/09537287.2021.1983661.

[3] L. Chu. Optimization method of fresh agricultural products cross-border e-commerce supply chain based on blockchain technology. Pakistan Journal of Agricultural Sciences, 60(2), 415-423, June, 2023, DOI: 10.21162/PAKJAS/23.140.

[4] X. Zhang and L. Ling. A review of blockchain solutions in supply chain traceability. Tsinghua Science and Technology, 28(3), 500-510, June, 2022, DOI: 10.26599/TST.2022.9010030.

[5] H. Luo, X. Yang, H. Yu, G. Sun, B. Lei, and M. Guizani. Performance analysis and comparison of non-ideal wireless PBFT and RAFT consensus networks in 6G communications. IEEE Internet of Things Journal, 1(6), 9752-9765, March, 2023, DOI: 10.1109/JIOT.2023.3323492.

[6] H. Wu, S. Jiang, and J. Cao. High-efficiency blockchain-based supply chain traceability. IEEE Transactions on Intelligent Transportation Systems, 24(4), 3748-3758, April, 2023, DOI: 10.1109/TITS.2022.3205445.

[7] T. Chen, Y. Li, and F. Xu. Traceability strategy choice in competing supply chains based on blockchain technology. International Transactions in Operational Research, 31(6), 3873-3904, November, 2024, DOI: 10.1111/itor.13332.

[8] A. S. Berkani, H. Moumen, S. Benharzallah, S. Yahiaoui, and A. Bounceur. Blockchain use cases in the sports industry: a systematic review. International Journal of Networked and Distributed Computing, 12(1), 17–40, June, 2024. DOI: 10.1007/s44227-024-00022-3.

[9] J. Tan, W. P. Wong, C. K. Tan, S. Jomthanachai, and C. P. Lim. Blockchain-based Logistics 4.0: enhancing performance of logistics service providers. Asia Pacific Journal of Marketing and Logistics, 36(6), 1442–1463, June, 2024. DOI: 10.1108/APJML-07-2023-0650.

[10] X. Feng, K. Cui, L. Wang, Z. Liu, and J. Ma. PBAG: A privacy-preserving blockchain-based authentication protocol with global-updated commitment in IoVs. IEEE Transactions on Intelligent Transportation Systems, 25(10), 13524-13545, October, 2024, DOI: 10.1109/TITS.2024.3399200.

[11] Y. He, M. Luo, B. Wu, L. Sun, Y. Wu, Z. Liu, and K. Xiao. A game theory-based incentive mechanism for collaborative security of federated learning in energy blockchain environment. IEEE Internet of Things Journal, 10(24), 21294-21308, December, 2023, DOI: 10.1109/JIOT.2023.3282732.

[12] Y. P. Zhou, X. J. Zhao, and L. Sun. Research on traceability strategy of food supply chain considering delay effect. Journal of Food Science, 87(11), 4831-4838, November, 2022, DOI: 10.1111/1750-3841.16278.

[13] X. Wu, H. Ling, H. Liu, and F. Yu. A privacy-preserving and efficient byzantine consensus through multi-signature with ring. Peer-to-Peer Networking and Applications, 15(3), 1669-1684, May, 2022, DOI: 10.1007/s12083-022-01317-4.

[14] H. Qin, Y. Cheng, X. Ma, F. Li, and J. Abawajy. Weighted Byzantine Fault Tolerance consensus algorithm for enhancing consortium blockchain efficiency and security. Journal of King Saud University-Computer and Information Sciences, 34(10), 8370-8379, November, 2022, DOI: 10.1016/j.jksuci.2022.08.017.

[15] C. Li, W. Qiu, X. Li, C. Liu and Z. Zheng. A dynamic adaptive framework for practical byzantine fault tolerance consensus protocol in the internet of things.

IEEE Transactions on Computers, 73(7), 1669-1682, July, 2024, DOI: 10.1109/TC.2024.3377921.

[16] F. Q. Ma, Q. L. Li, Y. H. Liu, and Y. X. Chang. Stochastic performance modeling for practical byzantine fault tolerance consensus in the blockchain. Peer-to-Peer Networking and Applications, 15(6), 2516-2528, November, 2022, DOI: 10.1007/s12083-022-01380-x.

[17] Z. F. Wang, S. Q. Liu, P. Wang, et al. BW-PBFT: Practical byzantine fault tolerance consensus algorithm based on credit bidirectionally waning. Peer-to-Peer Networking and Applications, 16(6), 2915-2928, November, 2023, DOI: 10.1007/s12083-023-01566-x.

[18] C. Li, J. Zhang, and X. Yang. Scalable blockchain storage mechanism based on two-layer structure and improved distributed consensus. The Journal of Supercomputing, 78(4), 4850-4881, March, 2022, DOI: 10.1007/s11227-021-04061-3.

[19] F. Tang, T. Xu, J. Peng, and N. Gan. TP-PBFT: A scalable PBFT based on threshold proxy signature for IoT-blockchain applications. IEEE Internet of Things Journal, 11(9), 15434-15449, September, 2023, DOI: 10.1109/JIOT.2023.3347232.

[20] Y. Li, H. Huang, A. Lan, and Z. Huang. A practical byzantine fault tolerance improvement algorithm based on credit grouping-classification. The Journal of Supercomputing, 80(14), 20270-20301, September, 2024, DOI: 10.1007/s11227-024-06199-2.

[21] J. Li, L. Cao, S. Zhao, Wan, and J. Bai. LC-PBFT: Layered cross-chain consensus algorithm based on forest topology. The Journal of Supercomputing, 80(12), 17849-17873, August, 2024, DOI: 10.1007/s11227-024-06122-9.

[22] X. Zhu, X. Hu, W. Zhu. RGPBFT: A reputation-based PBFT algorithm with node grouping strategy. Arabian Journal for Science and Engineering, 50(15): 11837-11850, August, 2025. DOI: 10.1007/s13369-024-09614-1.

[23] H. Qushtom, J. Mišić, V. B. Mišić, and Chang. A two-stage PBFT architecture with trust and reward incentive mechanism. IEEE Internet of Things Journal, 10(13), 11440-11452, July, 2023, DOI: 10.1109/JIOT.2023.3243189.

[24] S. Alshihri and S. Park, "A Decentralized Lightweight Blockchain Nodes Architecture Based on a Secure OpenFlow Protocol Controller Channel," Tehnički vjesnik, vol. 30, no. 1, pp. 114–121, February 2023. DOI: 10.17559/TV-20220427051644.

[25] G. Rigatos, M. Abbaszadeh, B. Sari, P. Siano, G. Cuccurullo, and F. Zouari. Nonlinear optimal control for a gas compressor driven by an induction motor. Results in Control and Optimization, 11, 100226, June, 2023. DOI: 10.1016/j.rico.2023.100225.

[26] A. Boulkroune, F. Zouari, and A. Boubellouta. Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems. Journal of Vibration and Control, February, 2025, OnlineFirst, DOI: 10.1177/10775463251320258.

[27] J. Li, X. Li, H. Zhao, B. Yu, T. Zhou, H. Cheng, and N. Sheng. MANDALA: A scalable blockchain model with mesh-and-spoke network and H-PBFT consensus algorithm. Peer-to-Peer Networking and Applications, 16(1), 226-244, January, 2023, DOI: 10.1007/s12083-022-01373-w.

[28] D. Zhang, N. H. A. Wahab, and A. W. M. Zin. Optimizing blockchain consensus: Incorporating trust value in the practical Byzantine fault tolerance algorithm with Boneh-Lynn-Shacham aggregate signature. Baghdad Science Journal, 21(2), 0633-0633, February, 2024, DOI: 10.21123/bsj.2024.9735.

[29] M. Mohit, S. Kaur, and M. Singh. Design and implementation of transaction privacy by virtue of ownership and traceability in blockchain based supply chain. Cluster Computing, 25(3), 2223-2240, June, 2022, DOI: 10.1007/s10586-021-03425-x.

[30] J. Zhang, P. Zhou, J. Wang, O. Alfarraj, S. Singh, and M. Zhu. A novel high-efficiency transaction verification scheme for blockchain systems. CMES-Computer Modeling in Engineering & Sciences, 139(2), 60-66, February, 2024, DOI: 10.32604/cmes.2023.044418.

[31] G. A. F. Rebello, G. F. Camilo, L. C. B. Guimaraes, L. A. C. de Souza, G. A. Thomaz, and O. C. M. B. Duarte. A security and performance analysis of proof-based consensus protocols. Annals of Telecommunications, 77(7-8), 517-537, August, 2022, DOI: 10.1007/s12243-021-00896-2.

[32] C. Mueller-Bloch, J. V. Andersen, J. Spasovski, and J. Hahn. Understanding decentralization of decision-making power in proof-of-stake blockchains: an agent-based simulation approach. European Journal of Information Systems, 33(3), 267-286, May, 2024, DOI: 10.1080/0960085X.2022.2125840.

[33] Q. Fan, Y. Xin, B. Jia, Y. Zhang, and P. Wang. COBATS: A novel consortium blockchain-based trust model for data sharing in vehicular networks. IEEE Transactions on Intelligent Transportation Systems, 24(11), 12255–12271, November, 2023. DOI: 10.1109/TITS.2023.3286432.

[34] A. Purusottama, Y. Sunitiyoso, and T. M. Simatupang. Exploring the potential of blockchain adoption for promoting value innovation: A case of the halal industry. Business Process Management Journal, 29(7), 2034–2058, October, 2023. DOI: 10.1108/BPMJ-04-2023-0267.

[35] Y. Zhou, X. Gao, and J. Nie. Value of blockchain-enabled supply chain traceability under competition. International Transactions in Operational Research, 31(6), 3669-3703, November, 2024, DOI: 10.1111/itor.13295.

[36] P. Nowvaratkoolchai, N. Thawesaengskulthai, W. Viriyasitavat, and P. Rangsunvigit. Blockchain-based cannabis traceability in supply chain management. International Journal of Advanced Computer Science and Applications, 15(2), 75–85, February, 2024. DOI: 10.14569/ijacsa.2024.0150210.