information

MDPI

Article

5G-Practical Byzantine Fault Tolerance: An Improved PBFT Consensus Algorithm for the 5G Network

Xin Liu [1], Xing Fan [1], Baoning Niu [1,*] and Xianrong Zheng [2,*]

[1] College of Computer Science and Technology (College of Data Science), Taiyuan University of Technology, Jinzhong 030600, China
[2] Information Technology & Decision Sciences Department, Old Dominion University, Norfolk, VA 23529, USA
* Correspondence: niubaoning@tyut.edu.cn (B.N.); x1zheng@odu.edu (X.Z.)

Abstract: The consensus algorithm is the core technology of blockchain systems to maintain data consistency, and its performance directly affects the efficiency and security of the whole system. Practical Byzantine Fault Tolerance (PBFT) plays a crucial role in blockchain consensus algorithms by providing a robust mechanism to achieve fault-tolerant and deterministic consensus in distributed networks. With the development of 5G network technology, its features of high bandwidth, low latency, and high reliability provide a new approach for consensus algorithm optimization. To take advantage of the features of the 5G network, this paper proposes 5G-PBFT, which is an improved practical Byzantine fault-tolerant consensus algorithm with three ways to improve PBFT. Firstly, 5G-PBFT constructed the reputation model based on node performance and behavior. The model dynamically selected consensus nodes based on the reputation value to ensure the reliability of the consensus node selection. Next, the algorithm selected the primary node using the reputation model and verifiable random function, giving consideration to the reliability of the primary node and the randomness of the selection process. Finally, we take advantage of the low latency feature of the 5G network to omit the submission stage to reduce the communication complexity from $O(N^2)$ to $O(N)$, where N denotes the number of nodes. The simulation results show that 5G-PBFT achieves a 26% increase in throughput and a 63.6% reduction in transaction latency compared to the PBFT, demonstrating significant performance improvements.

Keywords: blockchain; consensus algorithm; PBFT; 5G network

check for updates

Academic Editors: Kostas Vergidis and George Tsakalidis

Received: 16 January 2025
Revised: 21 February 2025
Accepted: 26 February 2025
Published: 5 March 2025

Citation: Liu, X.; Fan, X.; Niu, B.; Zheng, X. 5G-Practical Byzantine Fault Tolerance: An Improved PBFT Consensus Algorithm for the 5G Network. *Information* **2025**, *16*, 202. https://doi.org/10.3390/info16030202

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Blockchain is a decentralized distributed ledger technology that combines distributed storage [1], peer-to-peer network [2], consensus algorithm [3], cryptography [4], and other technologies to ensure data security and transparency by recording transaction information through a chain structure. The core issue of blockchain technology lies in ensuring the consistency of block data and block link relationships stored across all nodes. Consensus algorithms are precisely the key to solving this problem.

The consensus algorithm in blockchain ensures the validity of transactions among the participants, maintaining the security and consistency of the blockchain system [5]. Currently, the most widely used consensus algorithms in blockchain include Proof of Work (PoW) [6], Proof of Stake (PoS) [7], Delegated Proof of Stake (DPoS) [8], and Practical Byzantine Fault Tolerance (PBFT) [9]. Compared to PoW, PoS, and DPoS, PBFT is characterized by its absence of forks, high throughput, and ability to tolerate up to one-third of Byzantine nodes, making it the most prevalent consensus algorithm in consortium blockchains.

However, PBFT employs a fixed rotation mechanism to select primary nodes, which cannot guarantee the reliability of the selected nodes. Additionally, the consensus process involves two-phase network-wide broadcasting, resulting in high consensus response latency and significant communication overhead.

With its features such as high bandwidth, low latency, and high reliability, the 5G network has been widely applied in multiple fields, including the industrial Internet [10], smart cities [11], autonomous driving [12], and telemedicine [13]. The widespread adoption of the 5G network presents the possibility of leveraging its features to enhance PBFT performance. Specifically, the high bandwidth facilitates the faster synchronization of block and transaction information within the blockchain network, laying a solid foundation for handling large-scale and high-frequency transaction demands. The low latency feature can significantly reduce the communication time between nodes. Meanwhile, the high reliability can effectively minimize the risks of communication failures and data loss between nodes, ensuring the accuracy and consistency of information among nodes and the smooth execution of the consensus algorithm.

To address the above problems, this paper proposes the improved PBFT consensus algorithm 5G-PBFT for the 5G network. The main contributions and novelties of this work are as follows:

1. We propose a reputation model based on node performance and behavior to ensure the reliability of consensus node selection. This model allows for the dynamic adjustment of node reputation values, which not only motivates nodes to actively participate but also mitigates the risk of Byzantine behavior among nodes.
2. We select primary nodes based on the proposed reputation values of nodes and the verifiable random functions, which can improve the unpredictability of primary node selection and thereby guarantee the reliability of the system.
3. We eliminate the commit stage, simplifying the PBFT five-stage consensus process with four stages, which reduces communication overhead and shortens the consensus achievement time by leveraging the low latency features of the 5G network.

2. Related Work

This section first introduces the implementation principle of PBFT and then describes the consensus algorithms derived from PBFT from two aspects, i.e., primary node selection and reducing communication complexity. Finally, we summarize the shortcomings of the existing schemes and discuss some ideas to improve them.

2.1. Overview of PBFT

PBFT is a fault-tolerant algorithm applied in distributed systems, designed to address the issue of potential Byzantine nodes within the system. This algorithm ensures that all honest nodes reach consensus on a particular transaction or state, even in the presence of Byzantine nodes. The PBFT consensus process consists of views, with each round of consensus involving one primary node and multiple replica nodes. The primary node packages transactions into blocks and leads the block consensus process, serving in a leadership role, while the replica nodes participate in the block consensus. View changes are triggered when the primary nodes act maliciously.

The selection of the primary node is determined jointly by the view number and the total number of nodes, according to Equation (1), where p represents the node ID, v represents the view number, and N represents the total number of nodes.

$$p = v \bmod N \tag{1}$$

This fixed rotation mechanism implies that the ID of the primary node selected in the next round is predictable. Attackers can exploit this predictability of the primary node ID to launch a Denial-of-Service attack by sending a large number of invalid or malicious requests to exhaust the resources of this node, severely impeding the normal execution of the consensus algorithm. Thus, it can be seen that the traditional method of selecting the primary node in PBFT is unreliable and poses potential security risks.

The five phases of the PBFT are request, pre-prepare, prepare, commit, and reply. The execution process is shown in Figure 1, and the specific steps are as follows.

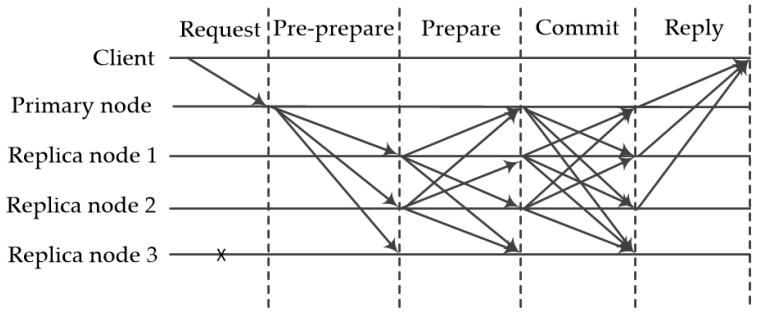


Figure 1. PBFT algorithm workflow. Replica node 3 is a Byzantine node.

The request phase: The client initiates a transaction request to the primary node, which is responsible for receiving and processing the request from the client.

The pre-prepare phase: The primary node numbers the request and broadcasts the pre-prepare message to all replica nodes to ensure the uniqueness and consistency of the message after it receives the request from the client.

The prepare phase: All nodes start to verify the message signature digest after receiving the information broadcast by the primary node, and if correct, start to broadcast the prepare message to all other nodes.

The commit phase: The node starts to verify the message signature digest after receiving the prepare message. A node will broadcast the commit message to the entire network when it has received $2f + 1$ prepare messages correctly and cumulatively, because it confirms that the request has been agreed by all nodes.

The reply phase: A node executes the client request and sends a reply to the client when it collects $2f + 1$ verified commit messages.

From the execution process of PBFT, it is evident that achieving consensus in PBFT requires two rounds of many-to-many broadcast across the entire network. Blockchain technology encourages as many nodes as possible to participate in consensus. As the number of nodes increases, so does the communication overhead, which inevitably leads to a decrease in consensus efficiency. Therefore, optimizing the consensus algorithm and reducing communication overhead are crucial for enhancing PBFT performance.

2.2. Improvement of PBFT

To address the two issues of unreliable primary node selection and high communication complexity in PBFT, many scholars have proposed optimized PBFT for selecting primary nodes and improving communication efficiency. In terms of selecting primary nodes, Credit-Layered Byzantine Fault Tolerance (CLBFT) [14] proposes a credit rating division mechanism to select high-credit nodes as primary nodes, while screening out low-credit nodes to avoid Byzantine nodes becoming primary nodes. Practical Byzantine Fault Tolerant Algorithm Based on Clustering (C-PBFT) [15] divides the network nodes into different classes by using the clustering method. The node with the highest reputation value in each class is responsible for generating new blocks, and its reputation value is

updated in a timely manner. If the node chooses to act badly, its reputation value will decrease due to system punishment. Improved Practical Byzantine Fault Tolerance Based on Comprehensive Evaluation (CE-PBFT) [16] weights the performance and reputation values of nodes before consensus to obtain a comprehensive evaluation, and filters out nodes that meet the needs of specific scenarios. Based on the comprehensive evaluation, the nodes are K-means clustered to form a two-layer network architecture. Selection-based Byzantine Fault Tolerance (SBFT) [17] randomly selects the primary node based on the node contribution value and the verifiable random function, combined with the password lottery, to ensure the unpredictability of the selection process.

The above studies can avoid the selection of Byzantine nodes as primary nodes through the reputation evaluation of nodes, but relying on the reputation model alone can only reduce the possibility of nodes acting badly in the network. In real situations, the performance of nodes also has an important impact on their consensus participation. Therefore, we propose a comprehensive measurement method combining node performance, which is very critical for us to improve the reliability of the consensus algorithm.

In terms of improving communication efficiency, Geographic-PBFT (G-PBFT) [18] selects the representatives of the primary nodes to implement PBFT to simplify the process and reduce overhead based on the mobility features of industrial Internet nodes. It reduces the latency in information confirmation by reducing the scope of internal communication, and solves the security risks of Sybil attacks through special geographic location identification. Double-layer PBFT [19] optimizes the complex internal communication in traditional PBFT, so that the PBFT process is carried out in a group composed of a fixed number of nodes. Also, different groups can reach a consensus at the same time, which significantly reduces the communication overhead. Dynamic Grouping Byzantine Fault Tolerance Mechanism (DGBFT) [20] divides nodes into groups based on trust values and assigns consensus tasks to each group based on the idea of grouping, thereby reducing the traffic in the network and improving the efficiency of consensus. Zyzzyva consensus [21] is a light version of the PBFT. The process of mutual confirmation by internal nodes is ignored and the results are directly returned to the client, assuming that the network nodes are honest and reliable. In the presence of Byzantine nodes, broadcasting is needed again only after receiving the results, which greatly reduces the complexity of internal communication.

The majority of the above studies are based on the hierarchical concept. In these studies, the consensus task is divided into various layers to reduce the communication overhead of PBFT. The use of the 5G network makes it possible to simplify the PBFT consensus process. In this paper, we aim to leverage the high bandwidth, low latency, and high reliability features of the 5G network to propose a novel PBFT optimization algorithm tailored for the 5G network.

3. The Overview of 5G-PBFT

To address the issues of unreliable primary node selection and high communication overhead in the existing PBFT and its related algorithms, this paper proposes an efficient consensus algorithm, 5G-PBFT, which leverages the reputation model, verifiable random function, and the features of the 5G network. First, we construct a reputation model based on node performance and behavior. The model enables the dynamic selection of consensus nodes based on their reputation values. Also, we propose a dynamic adjustment mechanism for the reputation value. Next, the primary node is selected based on the nodes' reputation values and a verifiable random function. It enhances the unpredictability of primary node selection. Finally, we leverage the reputation model and the features of the 5G network to optimize the consensus process. The five-stage consensus process

is streamlined into four stages, which reduces communication overhead and improves consensus efficiency.

3.1. The Reputation Model

Reputation is a way to represent the credibility of a node. The larger the reputation value, the higher the credibility of the node, and the lower the probability of it becoming a Byzantine node. Most of the existing reputation models determine the quality of nodes based on whether they have successfully completed the consensus process. However, they do not consider the resources, and may select the primary node with limited hardware resources, which could make it unable to carry out the consensus process. To solve this issue, this paper proposes a reputation model based on node performance and behavior. To ensure the reliability of the consensus node selection, the model dynamically evaluates the node behavior after each round of consensus and combines the performance of the node to calculate the reputation value of the node.

We classify nodes into three categories based on their reputation values. The first category is the primary node, which is responsible for collecting transaction information and packaging it to generate blocks. The reputation value of the primary node is higher than the initial value preset by the system; the higher the reputation value of the node, the greater the probability of it being selected as the primary node. The second category is the replica node, which receives the consensus message broadcast by the primary node, and verifies and participates in the consensus process. The reputation value of the replica node is greater than the lower limit of the reputation value set by the system. The third category is the nodes that are prohibited from participating in the consensus, which can only passively receive the blocks that reach consensus. The reputation value of the nodes is lower than the lower limit of the reputation value set by the system.

The node reputation is evaluated by the node performance and behavior. The node's performance reflects its ability to stably run the blockchain. The higher the node's performance, the more durably and stably it can participate in the blockchain consensus process. The performance of a node is determined by its hardware capacity. Considering that the node needs to generate blocks, forward messages, and store blocks, we assume in the paper that we can measure the node's performance by its computing power and storage capacity. The ratio of CPU computing power (MH/s) and storage space (MB) to the current highest standard represents the node's computing power and its available storage capacity, respectively. For example, if the maximum storage standard is 1 TB and the available storage space of the node is 512 GB, we calculate the storage capacity value as 0.5. Node i has computing power C_i^t and storage capacity S_i^t in round t. The performance score A_i^t of node i in round t is calculated by Equation (2).

$$A_i^t = \alpha_1 C_i^t + \alpha_2 S_i^t \tag{2}$$

In Equation (2), the parameters α_1 and α_2 represent the weight of computing power and storage capacity, respectively. $\alpha_1 + \alpha_2 = 1$, $0 < \alpha_1, \alpha_2 < 1$. The values of α_1 and α_2 are set, according to the requirements of the system.

Moreover, the paper proposes to construct a reputation model and calculate its corresponding reputation value based on the behavior of nodes in the network. If node i successfully participates in the consensus process and broadcasts the correct message in round t, the node is considered to behave normally. If node i packs invalid blocks in round t or fails to lead the whole network nodes to complete consensus, and the final synchronization result of the replica node is inconsistent with the majority of nodes, the corresponding primary node and replica node are considered to be abnormal. In particular, a node that does not send messages due to failure or downtime is considered as an offline node. If a

node sends inconsistent messages to the rest of the nodes or sends wrong messages so that the message cannot pass the verification, it is considered as a Byzantine node.

The blockchain system updates the node reputation value according to the reputation model after completing a round of consensus. $RV_i(t)$ is the reputation value of node i in round t consensus. If the node behaves normally, the reputation value $RV_i(t+1)$ of node i in round $t+1$ consensus is calculated using Equation (3).

$$RV_i(t+1) = RV_i(t) \times \left(1 + s_i \times A_i^t\right) \tag{3}$$

In Equation (3), the parameter s_i represents whether node i is a primary node, and the value of the replica node is 1, and the value of the primary node is 2. The parameter is set to make the reputation value of the primary node increase more than that of replica node when consensus is successful, and the reputation value of the primary node decrease more than that of the replica node when consensus fails.

Only when the nodes correctly participate in the consensus can they obtain higher reputation values based on higher performance scores, thus preventing high-performance Byzantine nodes from obtaining higher reputation values.

If the node is offline, the reputation value of node i in the $t+1$ round of consensus is calculated as shown in Equation (4).

$$RV_i(t+1) = RV_i(t) \times e^{-\Delta H} \tag{4}$$

In Equation (4), the parameter ΔH represents the difference between the current block height when the node is online again and the block height when it is offline. The longer the node is offline, the greater the ΔH, and the greater the reduction in its reputation value.

If the node is a Byzantine node, its reputation value is set to zero in the next round. Consequently, it is ineligible to participate in consensus in the network.

3.2. Reputation Dynamic Adjustment Mechanism

On one hand, motivating nodes with low reputation values to participate in the blockchain consensus process prevents nodes from being unable to re-participate in the blockchain consensus process. On the other hand, if the reputation value of a node accumulates without an upper limit, it may make its reputation value too high. It will cause the node to serve as the primary node with a probability much higher than that of other nodes, which in turn leads to the node serving as the primary node for a longer time. This will increase the risk of blockchain attacks. Therefore, the paper proposes a mechanism to dynamically adjust the node's reputation value. The initial reputation value of the node is RV_{init}. When the reputation value of the node is higher than the upper limit RV_{high} or lower than the lower limit RV_{low} (no less than zero), the reputation value will be dynamically adjusted as follows:

When the reputation value of a node exceeds the upper limit, we adjust its reputation value according to the activity rate. The activity rate AC_i of the node i is calculated using Equation (5).

$$AC_i = B_i / A_i \tag{5}$$

In Equation (5), B_i is the number of blocks generated in the blockchain within the time from the start of node i participating in the consensus to its reputation value exceeding the upper limit. A_i is the time that node i participates in the consensus.

The node activity rate represents the number of times that a node serves as the primary node. The higher the value of the activity rate, the more times the node serves as the primary node, and more blocks are packaged and generated by the node. Therefore, the reputation value of the node will be dynamically adjusted, after the end of the next

consensus process when its reputation value exceeds the upper limit. The equation for calculating the reputation value $RV_i(t+1)$ of node i in the $t+1$ round of consensus is as follows:

$$RV_i(t+1) = max\{RV_i(t) \times (1 - AC_i,\ RV_{\text{init}})\} \tag{6}$$

In Equation (6), the higher the activity rate AC_i of the node, the lower its reputation value in the next round, which reduces the probability of the node serving as the primary node in the next round. The equation ensures that the adjusted reputation value is no less than the minimum reputation value for participating in the selection of the primary node. The system adjusts the reputation value $RV_i(t+1)$ of the node in the $t+1$ round of consensus to RV_{low}, when its reputation value drops to the lower limit.

3.3. The Primary Node Selection Model

Through the model, the nodes, whose reputation value is higher than the initial value set by the system, are screened out. They have the opportunity to become primary nodes in the next round of consensus process. Then, Verifiable Random Function (VRF) [22] is used to select the primary node of the next round from the set of the candidate primary nodes. The selection process introduces randomness to ensure the reliability of the primary node and the safety of the system. The process is as follows:

Verifiable Random Function (VRF): The input of the function is the private key sk_i of node i and the *seed*, and the output is the two values VRF_hash_i and VRF_proof_i. For the same sk_i and *seed*, VRF will produce the same VRF_hash_i and VRF_proof_i, where VRF_hash_i is a pseudo-random string sequence of fixed length, and VRF_proof_i is used to prove whether VRF_hash_i is generated by VRF$(sk_i, seed)$. In the paper, the *seed* value is the ID of the latest block in the blockchain, which ensures that all nodes can use the same *seed* that can be updated dynamically.

When a node discovers that the current primary node has malfunctioned, for example, by not forwarding messages for a long time or forwarding incorrect messages, it will construct a $VIEW\text{-}CHANGE < VIEW - CHANGE, v+1, n, m, i >$ message locally and then broadcast it to the network. In the message, v represents the number of the previous view; n represents the maximum request sequence number of the consensus completed by node i; m represents the content of the primary node's malfunction, including timeout or forwarding messages that fail to pass verification; and i represents the node number of the node sending the message. When $2f+1$ nodes receive the message, it will trigger the replacement of the primary node. Then, the consensus nodes will execute the VRF and broadcast the calculated random number along with the corresponding proof to the network.

The nodes will conduct verification after receiving the random number and the corresponding proof. Once the verification is passed, the random number will be cached locally. Then, the nodes will perform a modulo operation on the largest random number. The node j will construct a *NEW-VIEW* message and broadcast it to the network if it is the next primary node. They will carry out verification after other nodes receive the message. The replacement of the primary node is completed if the verification is passed.

The reputation model mentioned above screens some nodes to become candidate primary nodes. To make the node with a high reputation value more likely to be selected as the primary node of the next round of consensus, the paper proposes to divide the node's reputation value into units according to a fixed value. For example, if the current reputation value of a node is 5 and the unit is set to 1, the node has 5 reputation value units. Each time a node has a reputation value unit, it can be considered as having a child node. The higher the reputation value of a node, the more reputation value units it contains, and the more child nodes it has. It should be noted that the child node is a fictitious node to help select

the primary node, not the real one in the system. When the primary node is selected, it is mapped to the node itself through the set of child nodes. The process is shown in Figure 2.

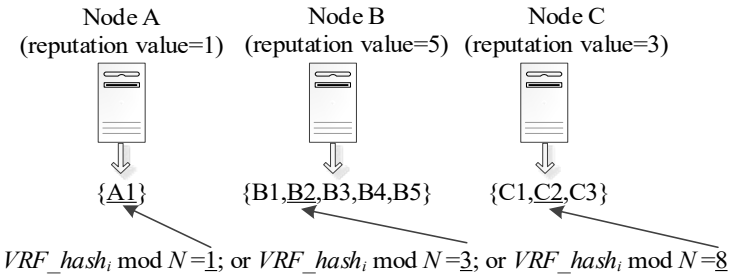


Figure 2. The chart of primary node selection.

As shown in Figure 2, there are three nodes in the current network, the names of the three nodes are A, B, C, the reputation values of the nodes are 1, 5, and 3, and the number of child nodes are one, five, and three, respectively. When using VRF to select the primary node, the value of N is the number of child nodes in the network. Here, the total number of nodes $N = 9$. When using the value of modulo operation for mapping, the nodes are sorted according to the node number size. The node A contains one child node {A1}. The node B contains five child nodes {B1, B2, B3, B4, B5}. The node C contains three child nodes {C1, C2, C3}. The modulo operation with a value of 1 is mapped to node 1, the modulo operation with a value of 3 is mapped to node 2, and the modulo operation with a value of 8 is mapped to node 3. With this scheme, it is easier for nodes with a high reputation value to be selected as primary nodes. The algorithm of the primary node selection is as depicted in Algorithm 1.

Algorithm 1. Primary node selection

Input : sk_i, $seed$, $\lfloor \sum_i^n RV_i \rfloor$
Output : VRF_hash_i, VRF_proof_i, num
1 : VRF_hash_i, $VRF_proof_i \leftarrow \mathrm{VRF}(sk_i, seed)$
2 : $N = \lfloor \sum_i^n RV_i \rfloor$
3 : $H = max(VRF_hash_i)$
4 : num $\leftarrow H$ mod N
5 : return VRF_hash_i, VRF_proof_i, num

3.4. Consensus Process of 5G-PBFT

In the PBFT prepare phase, the nodes need to communicate with each other to ensure that the received message is the same one sent by the primary node. The purpose of the commit phase is to ensure the state synchronization among nodes and prevent the received messages coming from different nodes due to the network latency issue. The paper proposes the following improvements since the two phases require many-to-many broadcasting with extremely high communication overhead:

First, the nodes do not need to perform the complex process of communicating with each other in the prepare phase. In 5G-PBFT, the primary node is more secure, because of the use of the node reputation model. The replica nodes do not need to communicate with each other to confirm the consistency of the message, but only need to communicate with the primary node. When the primary node receives more than $2f + 1$ prepare messages, it is assumed that most nodes in the network are ready to execute client requests. At this point, the primary node can broadcast the message of executing the client request to the remaining nodes. Next, the nodes will not fail to receive prepare messages from other

nodes due to the low latency environment of the 5G network. Also, the nodes do not need to communicate with each other in the commit phase. When a node receives the execution message from the primary node, it means that the rest of the nodes can receive the same execution message. At this point, all nodes in the network execute the client request. Therefore, eliminating the commit phase can guarantee state consistency among nodes too. The evaluation of the network latency where the commit phase can be omitted will be conducted in the simulation experiment section. The specific consensus process is shown in Figure 3.

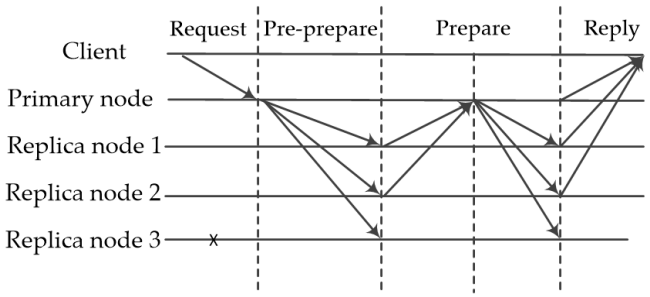


Figure 3. The workflow of the 5G-PBFT algorithm. Replica node 3 is a Byzantine node.

The request phase: the client initiates a request and sends the request message to the consensus network.

The pre-prepare phase: The primary node packages the transactions published by the client in the blockchain, generates the corresponding block, and broadcasts the pre-preparation message in the network. The specific format is <*Pre-Prepare*, *v*, *n*, *d(b)*, *t*, *b*>, where *v* is the current view number, *n* is the request number, *d(b)* is the block summary, *t* is the timestamp, and *b* is the block.

The prepare phase: The node verifies after receiving the pre-preparation message. After the verification is passed, the replica node sends a prepare message to the primary node in the format <*Prepare*, *v*, *n*, *d*, *i*>, where *v*, *n*, and *d* are the same as in the pre-prepare message, and *i* is the ID of the node. The primary node assumes that most nodes in the network are ready to execute the client's request, when it receives $2f + 1$ prepare messages. At this time, the primary node broadcasts a prepare message to all consensus nodes.

The reply phase: all nodes will receive the prepare message from other nodes within a certain time frame. After the verification is passed, which means that the prepare stage has been completed, the nodes will execute the transaction and return a reply message to the client. The transaction has been confirmed when the client receives more than $f + 1$ reply messages.

We present below the theoretical analysis of communication complexity between 5G-PBFT and PBFT. Assuming that the total number of nodes in the system is N. In the PBFT consensus process, the number of communications required in the pre-prepare phase, prepare phase, and commit phase are $N - 1$, $(N - 1)^2$, and $N(N - 1)$, respectively, and the total number of communications is $2N(N - 1)$. However, in the 5G-PBFT consensus process, the number of nodes participating in the consensus is $hN(0 < h \leq 1)$, the number of communications in the pre-prepare phase and the prepare phase are $(hN - 1)$ and $2(hN - 1)$, respectively, and the total number of communications is $3(hN - 1)$. It can be seen that 5G-PBFT simplifies the PBFT consensus process, reduces the communication complexity from $O(N^2)$ to $O(N)$, and improves the performance significantly.

4. Simulation Results and Analysis

In this paper, we conduct experiments on an Inspur server, which is manufactured by a company from Jinan, China, equipped with an Intel (R) Core (TM) i7-13600KF processor

running at 2.50 GHz, 32 GB of memory, and a 2 TB hard disk. We build a 5G network simulation environment based on OMNeT++ 4.6. We use docker to simulate communication between multiple nodes and implement 5G-PBFT using Python 3.6. Each block contains 1000 transactions. The parameters of the experiment are set as follows:

In the reputation model, the initial value of the reputation value $RV_{\text{init}} = 5$; the upper limit of the reputation value $RV_{\text{high}} = 10$; the lower limit $RV_{\text{low}} = 2$; and the value of s_i is {1, 2}, where one indicates that the node is the primary node, and two is the opposite. The weight of computing power and the weight of storage power are both set to 0.5.

To demontrate the effectiveness of the 5G-PBFT algorithm, we compare PBFT [9] with SBFT [17] in terms of communication overhead, throughput, and transaction latency.

4.1. Network Latency Evaluation

To address the high communication cost in the PBFT, we propose the 5G-PBFT, which leverages the low latency feature of 5G networks to eliminate the commit phase. To ensure the feasibility of this approach, we conduct simulation experiments to evaluate the network latency where consensus can still be reached even the commit phase is eliminated. In the experiment, we set the total number of nodes to 300. As the network latency increases, we calculate the ratio of the number of transactions that successfully achieve consensus to the total number of transactions submitted to the network for consensus processing within 30 min. The ratio represents the consensus success rate. The results are shown in Figure 4.

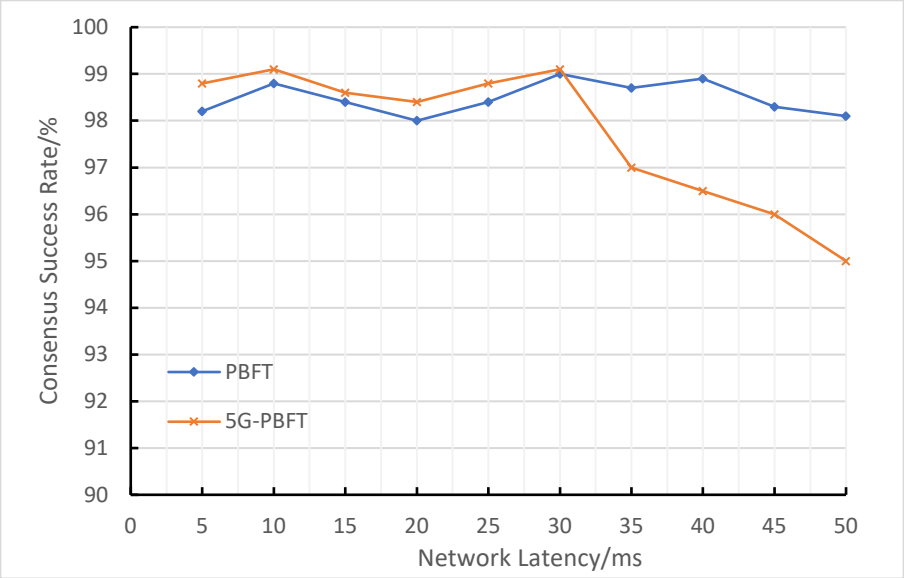


Figure 4. Comparison chart of consensus success rates.

From the experimental results, it can be seen that when the network latency is below 30 milliseconds (ms), the 5G-PBFT has a similar or even higher consensus success rate compared with PBFT. However, when the latency exceeds 30 ms, the consensus success rate of 5G-PBFT starts to decline. In contrast, the consensus process of PBFT features strong stability. Even if the network latency increases to some extent, it can still ensure that all nodes successfully reach a consensus. As the 5G-PBFT eliminates the commit phase to improve efficiency, its nodes cannot reach a consistent state, when the network latency exceeds 30 ms. Therefore, the 5G-PBFT is feasible, when the network latency is below 30 ms.

4.2. Communication Overhead Analysis

Studying the change in the communication overhead with nodes can determine the scalability of the blockchain. If the increase in the number of nodes has a small impact on the communication overhead, then the blockchain scales well. In the experiment, the average number of communications required to generate an effective block is used to measure the communication overhead of 5G-PBFT, SBFT, and PBFT. When there are Byzantine nodes in the system, the probability of view changes increases, which will bring additional communication overhead. Therefore, this paper conducts multiple experiments to calculate the average value in the presence of Byzantine nodes. The number of Byzantine nodes is set to be $\lfloor (N-1)/3 \rfloor$, where N is the maximum number of nodes that the system can allow.

The results are shown in Figure 5. The x-axis represents the number of nodes, and the y-axis represents the communication overhead. When the number of consensus rounds $M = 10$, the communication overhead of the three algorithms varies with the number of nodes in the system.

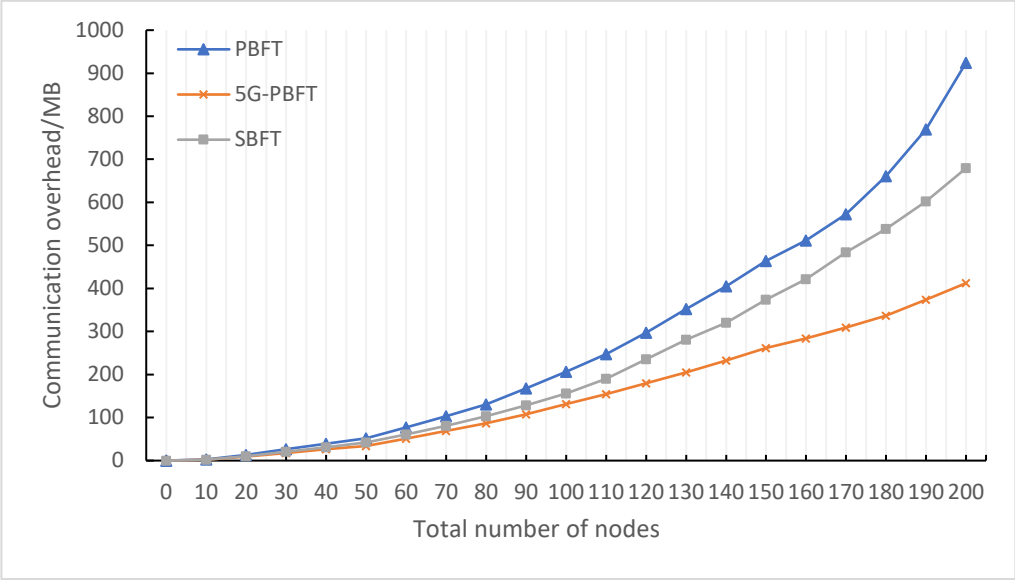


Figure 5. Comparison of communication overhead.

From the experimental results, we can see that PBFT has the largest communication overhead. The theoretical analyses are as follows: Under the same view, the communication complexity of SBFT and 5G-PBFT in the consensus process is $O(N)$, while the communication complexity of PBFT is $O(N^2)$. The primary node selected by 5G-PBFT and SBFT is more reliable, because the reputation model is established and the probability of view changes is reduced. Therefore, the communication overhead of 5G-PBFT and SBFT is lower than that of PBFT.

Compared with SBFT, 5G-PBFT reduces the number of communications, since there is no commit phase in the consensus process. Moreover, 5G-PBFT selects the primary node according to the reputation value and VRF, which can greatly reduce the probability of Byzantine nodes becoming primary nodes. So, the communication overhead of 5G-PBFT is superior to SBFT. For instance, when the number of nodes is 100, the communication overhead of PBFT and SBFT is about 1.5 times and 1.2 times that of 5G-PBFT, respectively. When the number of nodes is 200, the communication overhead of PBFT and SBFT is about 2 times and 1.5 times that of 5G-PBFT, respectively. With the increase in the number of nodes, this gap will continue to expand. This indicates that 5G-PBFT is more scalable. The communication overhead of 5G-PBFT is better than that of PBFT and SBFT.

4.3. Throughput Analysis

In the consensus field, the throughput can directly reflect the performance of the algorithm. The higher the throughput, the better the algorithm's performance. The Transactions Per Second (TPS) measure the throughput of 5G-PBFT, SBFT, and PBFT. In the experiment, we consider both the absence and presence of Byzantine nodes. In the case of Byzantine nodes, the number of Byzantine nodes is set to be $\lfloor (N-1)/3 \rfloor$, where N is the maximum number of nodes that the system can allow. The comparison results of the throughput of the three algorithms with the number of nodes in the network are shown in Figures 6 and 7.

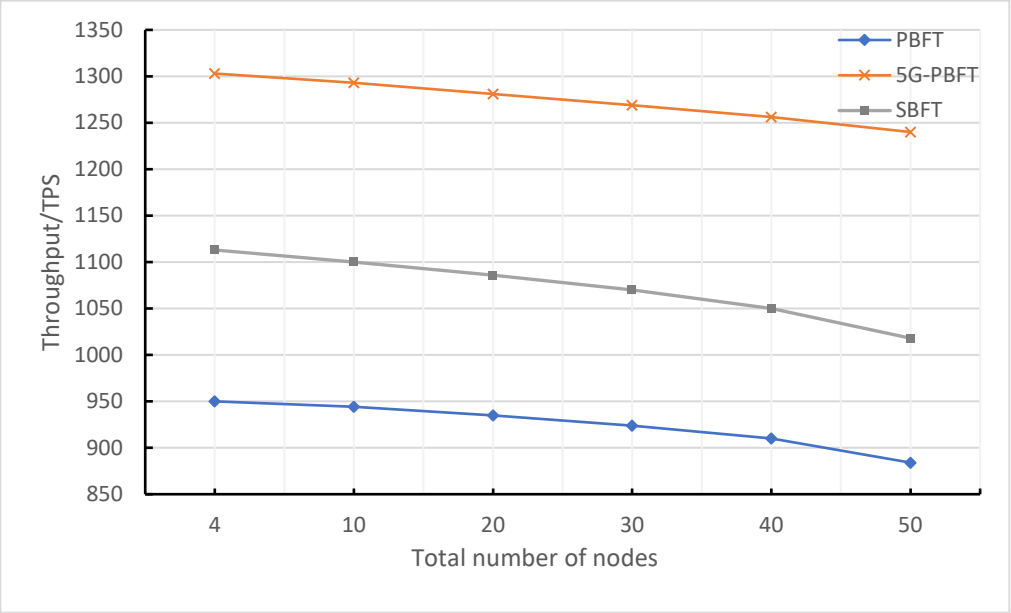


Figure 6. Comparison of throughput without Byzantine nodes.

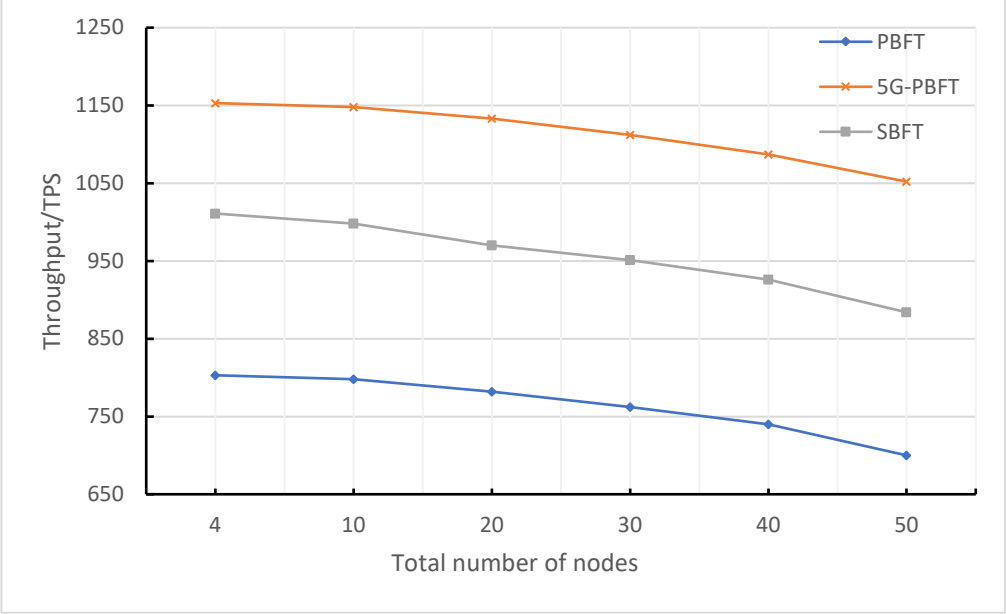


Figure 7. Comparison of throughput with Byzantine nodes.

With the increase in the number of nodes, the traffic in the network and the time it takes for nodes to process transactions will increase, which will lead to the decline in the consensus efficiency and the reduction in throughput. Therefore, the throughput of the three algorithms shows a downward trend in the two figures.

From the comparison between Figures 6 and 7, it can be observed that the throughput of the system without Byzantine nodes is higher than that of the system with Byzantine nodes. This is because the probability of view changes increases when the system has Byzantine nodes. The blockchain consensus process is interrupted during view changes. As a result, the throughput is lower. Also, the throughput of 5G-PBFT is better than that of PBFT and SBFT.

Overall, regardless of whether there are Byzantine nodes or not, 5G-PBFT outperforms PBFT and SBFT in terms of throughput. This is because 5G-PBFT optimizes the consensus process, and its primary node selection model is more reliable, which has the advantage of quickly reaching consensus. So, the throughput is higher.

4.4. Transaction Latency Analysis

The transaction latency is an important indicator that measures the efficiency of the blockchain. A lower transaction latency means that transactions can be confirmed faster, which saves time and reduces the risk of blockchain bifurcation. The measurement of transaction latency in the experiment is the time from the submission of a transaction by a client to the confirmation of the transaction. The experiments are carried out under cases with the presence and absence of Byzantine nodes, where the number of Byzantine nodes is set as the maximum value allowed by the system. The transaction latency from multiple experiments is averaged, and the results without and with Byzantine nodes are shown in Figures 8 and 9, respectively.

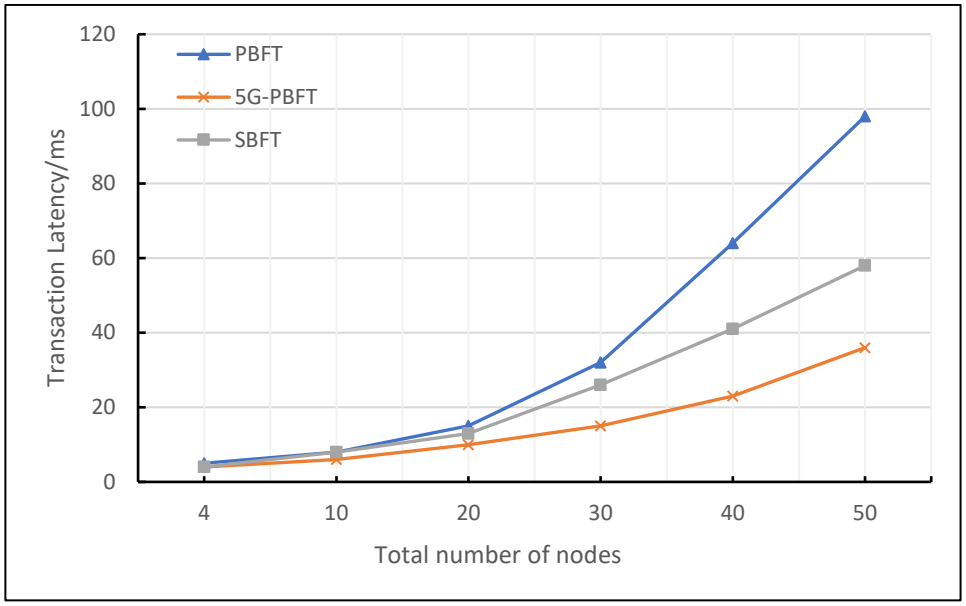


Figure 8. Comparison of transaction latency without Byzantine nodes.

It can be concluded that regardless of whether there are Byzantine nodes in the blockchain system, the transaction latency of PBFT and SBFT is higher than that of 5G-PBFT. The reasons are as follows: In the PBFT consensus process, nodes need to communicate with each other in the prepare phase and commit phase, and the time complexity is $O(N^2)$. SBFT optimizes the communication process compared to PBFT. However, the commit phase still needs to be performed, which greatly increases the time required for transaction confirmation. In 5G-PBFT, nodes do not need to communicate with each other, but through the primary node. The time complexity can be reduced to $O(N)$ without view changes. So, the commit phase is omitted to further reduce the time required for transaction confirmation. For instance, in the case of no Byzantine nodes, when the number of nodes

is 50, the transaction latency of 5G-PBFT is 63.6% lower than that of PBFT and 36% lower than that of SBFT.

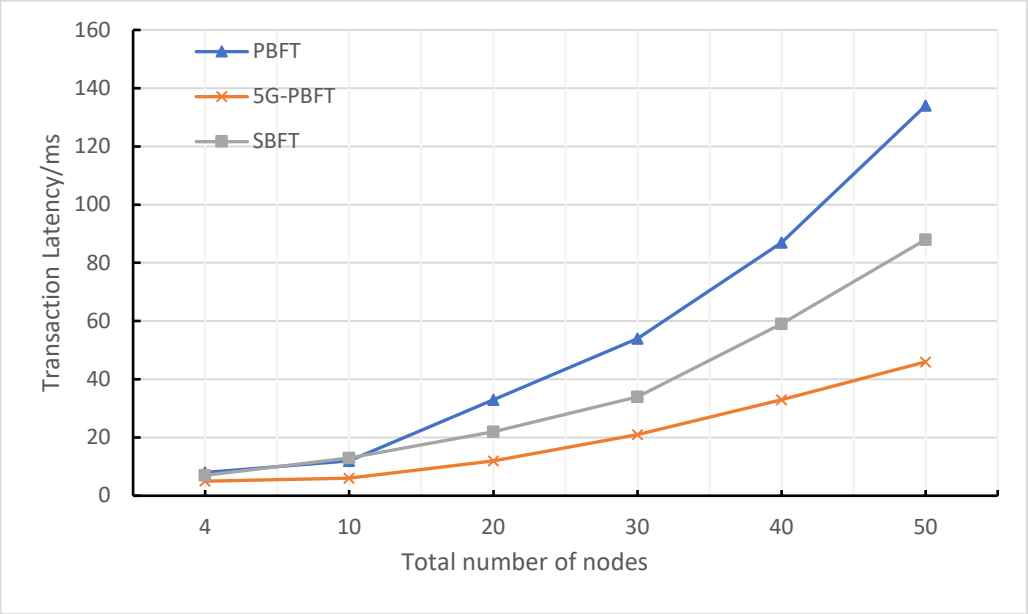


Figure 9. Comparison of transaction latency with Byzantine nodes.

Regardless of whether there are Byzantine nodes or not, the transaction latency of the three algorithms increases as the number of nodes becomes larger. This is because when the number of nodes increases, the number of communications between nodes during consensus also becomes larger. In addition, the 5G-PBFT transaction latency has the smallest increase as the number of nodes becomes larger. This shows that the proposed algorithm is highly scalable.

When there are Byzantine nodes in the system, they may act as primary nodes. The transaction confirmation time will be delayed. Thus, the transaction latency of PBFT, SBFT, and 5G-PBFT increases compared to the case without Byzantine nodes.

5G-PBFT uses the VRF and reputation model to select primary nodes, which greatly reduces the probability of Byzantine nodes becoming primary nodes. The probability of triggering view changes is reduced. The growth rate of transaction latency is less than that of PBFT and SBFT. When the number of nodes increases from 10 to 50, the transaction latency of PBFT is up 16.2 times, SBFT 8 times, and 5G-PBFT 6 times. Therefore, the performance of 5G-PBFT is better than that of PBFT and SBFT in terms of transaction latency.

5. Conclusions and Future Work

This paper proposes an improved PBFT consensus algorithm, 5G-PBFT, for the 5G network, which solves the problems of lacking security in PBFT primary node selection and the high network communication overhead. Firstly, 5G-PBFT proposes a reputation model based on node performance and behavior, which comprehensively evaluates the node according to its hardware capacity and behavior in the consensus process. The nodes whose reputation values are too low will not be able to participate in the consensus process. After each round of consensus, the reputation value is dynamically adjusted to balance the number of times that a node is elected as the primary node, which takes into account the fairness and security of the primary node selection. Secondly, 5G-PBFT uses a verifiable random function to select the primary node among the nodes that meet the conditions. The node with a higher reputation value has a higher probability of being selected as the primary node. The algorithm avoids the risk of selecting the primary node based

on the node number sequence in PBFT and reduces the probability of Byzantine nodes being selected as the primary node. Finally, 5G-PBFT optimizes the consensus process of PBFT, which reduces the communication complexity from $O(N^2)$ to $O(N)$. The speed of the nodes in verifying transactions is improved, which increases the system throughput. The simulation results show that 5G-PBFT is superior to the PBFT and SBFT consensus algorithms, in terms of communication overhead, throughput, and transaction latency. However, the algorithm may not be able to prevent external malicious attacks, while 5G provides low latency and high reliability. Improving the system security while ensuring the efficiency of the system is an important research direction in the future.

Author Contributions: Conceptualization, X.L., X.F. and B.N.; methodology, X.L. and B.N.; validation, X.L., X.F. and B.N.; formal analysis, X.L., X.F., B.N. and X.Z.; writing—original draft preparation, X.L.; writing—review and editing, X.L., X.F., B.N. and X.Z.; supervision, X.F., B.N. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Shanxi Major Science and Technology Special Project under grants 202301020101001, Key R&D Program of Shanxi Province under grants 202302010101004, and Shanxi Provincial Applied Basic Research Program under grants 202203021222093.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Apat, H.K.; Sahoo, B. A Blockchain assisted fog computing for secure distributed storage system for IoT applications. *J. Ind. Inf. Integr.* **2024**, *42*, 100739. [CrossRef]
2. Wang, S.; Ma, Z.; Wang, R. Performance analysis of a queueing system based on working vacation with repairable fault in the P2P network. *J. Supercomput.* **2023**, *79*, 12902–12923. [CrossRef]
3. Biswas, S.; Sharif, K.; Li, F. PoBT: A lightweight consensus algorithm for scalable IoT business blockchain. *IEEE Internet Things J.* **2019**, *7*, 2343–2355. [CrossRef]
4. Karthik, P.; Srinivasan, S.; Jayaprakash, J. An empirical evaluation of SHA-256 on the efficacy of random oracle for forestalling the security threats of blockchain-based cyber-physical systems. In Proceedings of the Congress on Smart Computing Technologies, Singapore, 2–3 December 2023.
5. Zheng, Z.; Xie, S.; Dai, H.; Chen, X. An overview of blockchain technology: Architecture, consensus, and future trends. In Proceedings of the IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 25–30 June 2017.
6. Lan, X.; Tang, X.; Zhang, R.; Li, B. Distributionally robust mining for proof-of-work blockchain under resource uncertainties. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Dubai, United Arab Emirates, 21–24 April 2024.
7. Semantic Scholar. Available online: https://people.cs.georgetown.edu/~clay/classes/fall2017/835/papers/peercoin-paper.pdf (accessed on 11 November 2024).
8. Liu, M.; Tan, M.; Lei, P. Research on DPOS consensus mechanism improvement. In Proceedings of the 2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Dali, China, 12–14 October 2022.
9. Castro, M.; Liskov, B. Practical Byzantine fault tolerance. In Proceedings of the 3rd Symposium on Operating Systems Design and Implementation (OSDI), New Orleans, LA, USA, 22–25 February 1999.
10. Peter, O.; Pradhan, A.; Mbohwa, C. Industrial internet of things (IIoT): Opportunities, challenges, and requirements in manufacturing businesses in emerging economies. *Procedia Comput. Sci.* **2023**, *217*, 856–865. [CrossRef]
11. Wang, M.; Mao, J. Smart city transportation: A VANET edge computing model to minimize latency and delay utilizing 5G network. *J. Grid Comput.* **2024**, *22*, 25. [CrossRef]
12. Zoghlami, C.; Kacimi, R.; Dhaou, R. 5G-enabled V2X communications for vulnerable road users safety applications: A review. *Wirel. Netw.* **2023**, *29*, 1237–1267. [CrossRef]
13. Liang, W.; Chan, S.; Lee, H. Feasibility and safety of 5G-based telerobotic abdominal ultrasonography. *J. Ultrasound Med.* **2024**, *43*, 355–360. [CrossRef] [PubMed]
14. Ding, T.; Chen, S. Improved PBFT consensus mechanism based on credit-layered mechanism. *Comput. Syst. Appl.* **2020**, *29*, 255–259.
15. Liu, W.; Ruan, M.; She, W. PBFT optimized consensus algorithm for the internet of things. *Comput. Sci.* **2021**, *48*, 151–158.

16. Qi, W.; Shi, P.; Gu, C. Improved PBFT consensus algorithm for multi-scenario internet of things. *Appl. Res. Comput.* **2024**, *41*, 676–682, 687.
17. Huang, B.; Peng, L.; Zhao, W.; Chen, N. Practical Byzantine consensus algorithm based on verifiable random functions. *Comput. Sci.* **2023**, *50*, 727–732.
18. Lao, L.; Dai, X.; Xiao, B. G-PBFT: A Location-based and scalable consensus protocol for IoT-blockchain applications. In Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS), New Orleans, LA, USA, 27–31 May 2024.
19. Li, W.; Feng, C.; Zhang, L. A scalable multi-layer PBFT consensus for blockchain. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *32*, 1146–1160. [CrossRef]
20. Yu, G.; Wu, B.; Niu, X. Improved blockchain consensus mechanism based on PBFT algorithm. In Proceedings of the 2020 2nd International Conference on Advances in Computer Technology, Information Science and Communications (CTISC), Suzhou, China, 20–22 March 2020.
21. Kotla, R.; Alvisi, L.; Dahlin, M. Zyzzyva: Speculative Byzantine fault tolerance. In Proceedings of the 21st ACM Symposium on Operating Systems Principles, Stevenson, WA, USA, 14–17 October 2007; Volume 27, pp. 1–39.
22. Yao, S.; Zhang, D. An anonymous verifiable random function with applications in blockchain. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 6467866. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.