

Received 12 October 2023, accepted 25 October 2023, date of publication 30 October 2023, date of current version 6 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3328539

## RESEARCH ARTICLE

# Improved PBFT Algorithm Based on K-Means Clustering for Emergency Scenario Swarm Robotic Systems

YI SUN<sup>1,2</sup> AND YING FAN<sup>1</sup> 

<sup>1</sup>College of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an 710054, China

<sup>2</sup>Xi'an Key Laboratory of Heterogeneous Network Convergence Communication, Xi'an 710054, China

Corresponding author: Ying Fan (ffany99@163.com)

**ABSTRACT** In order to solve the problem of data security and sharing efficiency caused by the complex and uncertain environment faced by swarm robotic systems in emergency scenarios, this paper designs a data security consensus algorithm based on blockchain technology. Aiming at the problems of high communication overhead, long consensus delay and low throughput when the traditional Practical Byzantine Fault Tolerance (PBFT) algorithm is applied to this scenario, a grouped practical Byzantine Fault Tolerance consensus algorithm based on K-means clustering is proposed. Firstly, the K-means clustering algorithm is used to group robotic nodes according to the location distribution of robotics in the emergency scenario. Secondly, a reputation mechanism is designed to dynamically evaluate the behaviour of robotic nodes in each group during the consensus process. Each consensus node is divided into master and slave chains according to its reputation score. The consensus task is firstly decomposed, and slave chains participate in the consensus in parallel. Finally, the master chain completes the global consensus, so as to reduce the communication times between nodes. The experimental results show that compared with the traditional PBFT, SG-PBFT and P-PBFT consensus algorithm, the proposed consensus algorithm effectively improves the system throughput, reduces the delay, reduces the communication overhead, and has a higher success rate of consensus.


**INDEX TERMS** Swarm robotics, blockchain, PBFT, K-means clustering, master-slave multichain.

## I. INTRODUCTION

In recent years, natural disasters and sudden accidents have occurred frequently all over the world. Due to the suddenness, complexity and uncertainty of various accidents, disasters or events, it is easy to cause serious casualties and economic losses [1], [2], [3]. In the emergency scene, the use of rescue personnel to detect dangerous and hazardous areas, organize the rescue of victims, eliminate the harmful consequences of post-disaster accidents, and restore the scene [4]. On the one hand, it will make them face a great risk of casualties and hinder the rescue work. On the other hand, many spaces cannot be reached by manpower in the emergency scene, which will make it difficult to implement rescue activities. Therefore, in this context, the use of robotics to replace staff

to perform tasks is of great significance to reduce the danger of rescue tasks and improve rescue efficiency.

Robotics replace rescuers in emergency scenarios to monitor the environment, provide communication support when communications are disturbed or interrupted, search and rescue places that are inaccessible or dangerous to rescuers, thereby reducing rescue costs, casualties, and improving rescue efficiency. The initial robotic system was mainly the single robotic [5], it designed to perform specific tasks. In the face of complex tasks and large-scale scenarios, the capabilities of single robotics are limited, and the work efficiency is generally not high. Especially in emergency scenarios, due to the changing environment and unknown scene information, there will be many complicated situations. Relying on a single robotic can no longer meet the task requirements in this scenario. Compared with the single robotic, the mode of the swarm robotic system [6] has

The associate editor coordinating the review of this manuscript and approving it for publication was Mueen Uddin .

gradually become an effective way to improve the efficiency of task execution by decomposing complex tasks and then cooperating among multiple robotics. At the same time, the swarm robotic system is much better than the single robotic in terms of robustness, fault tolerance, perception ability, and task execution strength.

Most of the current swarm robotic systems adopt centralized data processing and storage modes [7]. In the face of emergency scenarios, traditional centralized swarm robotic systems will face many challenges, such as insufficient fault tolerance, and individual failures may lead to system failures and data loss. In the face of a large number of task requirements, the calculation cost of the system is huge, and it is difficult to respond in time. The dynamic adjustment capability of the system is limited, and it is difficult to make adaptive adjustments according to changes in the system scale. Therefore, adopting the distributed network structure [8] has become an effective measure for the swarm robotic system to solve such problems in this scenario. For distributed swarm robotic systems in emergency scenarios, swarm robotic collaboration relies on reliable communication between them. Therefore, how to ensure the data security and information integrity of the system to ensure the efficiency of the swarm robotic system to perform tasks is the focus of this research.

As a new distributed computing paradigm [9], blockchain provides a decentralized, tamper-proof, transparent and traceable distributed database solution. Applying blockchain technology to the swarm robotic system can effectively solve the problems of poor scalability and unreliable central nodes of the traditional swarm robotic system in emergency scenarios, so as to ensure the data security of the entire system. As the core technology of the blockchain, the consensus algorithm [10] can improve the collaboration, fault tolerance ability and decision-making efficiency of robotics, thereby enhancing the efficiency of task execution. As one of the classic algorithms in the field of distributed systems, the PBFT algorithm has a high throughput and a response time of seconds [11], [12], [13]. It achieves consensus in the blockchain system through voting. Even if there are Byzantine error nodes, it can still ensure the efficiency and security of the consensus process. Therefore, the PBFT consensus mechanism is widely used in blockchain application scenarios. The application of the PBFT algorithm to swarm robotic systems in emergency scenarios can ensure data consistency between robotic nodes, thereby improving the reliability and stability of the entire system. However, PBFT requires multiple rounds of communication between nodes to reach a consensus. When the number of robotic nodes in the system increases, the communication traffic between nodes increases sharply, which puts huge pressure on the network bandwidth and leads to a rapid decline in system performance, and it is difficult to meet the real-time and accurate data requirements of the communication process of the swarm robotic system in large-scale emergency scenarios. At the same time, there is only one primary node in the

PBFT algorithm. If this node fails, the view needs to be changed frequently, which will affect the performance of the whole system and increase the communication overhead of the system.

Therefore, based on the swarm robotic system in large-scale emergency scenarios, given the shortcomings of the above PBFT consensus algorithm and in order to improve the applicability of the PBFT algorithm in this scenario, this paper proposes an optimized PBFT consensus algorithm based on K-means clustering. Specifically, the main contributions of this study are as follows:

- 1) Each robotic node is grouped according to the region based on K-means clustering. The communication between robotic nodes with distance advantage reduces the consensus delay and energy consumption between each group of nodes, and improves the stability of the consensus process.
- 2) We propose a reputation scoring mechanism to dynamically evaluate the behaviour of nodes in each region, select nodes with better performance in each region to form the master chain to increase the reliability of the primary node and solve the single point failure problem in the traditional PBFT consensus algorithm.
- 3) Based on the results of the above K-means consensus algorithm and the reputation scoring mechanism, we introduce a master-slave multi-chain data storage and sharing structure. On the one hand, this structure ensures the security of the data in the system, on the other hand, the system consensus task is decomposed into groups of local consensus, which reduces the overall communication overhead and consensus delay and improves the consensus efficiency. At the same time, the scalability of the system is enhanced, and new nodes can be managed and introduced more easily.
- 4) Experimental analysis and evaluation verify that our algorithm has fewer communication times, and has obvious advantages in throughput, transaction delay, stability and consensus success rate, indicating that the system has high consensus efficiency and execution speed, which is very suitable for this scenario.

The rest of this paper is organized as follows. Section II discusses the related work, Section III describes the system model and the various parts of the algorithm before and after the improvement, Section IV is the performance analysis and testing of the algorithm, and Section V concludes this paper.

## II. RELATED WORK

Blockchain technology has been widely used in many fields [14], and it can improve data transparency, security and sharing efficiency. In the following, we provide a detailed comparison of the existing research works, as shown in Table 1. Alsamhi et al. [15] applied blockchain technology to swarm robotic systems to improve work efficiency and fight against epidemics. Hamledari and Fischer [16] combined smart contracts with robotics to achieve distributed storage of progress data and efficient payment management. Malsa et al.

**TABLE 1. Comparison of existing research works.**

methods	contributions	limitations
[15]–[22]	It avoids the risk of centralization and ensures the security of system data.	The impact of the consensus algorithm on the whole distributed system is not considered.
[23], [24]	It provides tamper-evidence and non-repudiation features with high performance.	It is more suitable for applications that require fine-grained control and authority management.
[25]–[30]	The consensus efficiency is improved.	It can not realize parallel processing of multi-tasks, and the selection method of primary node is not safe enough.
[31]–[39]	The reliability of the primary node is improved.	The primary node is held by a fixed node, which reduces the degree of decentralization.

[17] proposed a blockchain-based certificate verification technique for robotic systems to check the authenticity of certificates. Salimi et al. [18] utilize blockchain technology to establish trusted data sharing. Alladi et al. [19] applied blockchain technology to the Unmanned Aerial Vehicle (UAV) network to realize the safe storage and management of UAV data. Vasyukovskiy et al. [20] designed a blockchain-based data access control model to control each user's data in a distributed manner to ensure data privacy. Gupta et al. [21] combined smart contracts and the Inter Planetary File System (IPFS) to solve the data security and privacy issues of intelligent telesurgery systems. Allouch et al. [22] proposed a lightweight security solution based on blockchain to ensure data security when UAV systems are connected to the Internet. At the same time, LedgerDB [23] and VeDB [24], as the latest ledger databases, provide tamper-evidence and non-repudiation features with high performance. It has strong auditability and wide verification scope, and can flexibly support tamper-proof applications.

The above data security solutions based on blockchain technology can effectively avoid the single-point risk of the traditional method and ensure the data security of the system, but the impact of the consensus algorithm on the whole distributed system is not taken into account. PBFT is a commonly used algorithm to solve the consensus problem of distributed systems, but it is limited in practical applications due to its poor scalability, low efficiency of multi-node consensus, and simple selection of master nodes. Therefore, many researchers have improved and innovated PBFT. Srinivas Aditya et al. [25] proposed a hybrid consensus algorithm combining PBFT and Proof of Stake (POS), which reduced the number of consensus nodes to a constant value through pseudo-random ordering. Li et al. [26] proposed a scalable multi-layer PBFT consensus algorithm to allocate nodes to different layers to reduce the number of communications between nodes. The NBFT proposed by Yang et al. [27] uses the hash algorithm to group nodes to reduce the communication complexity of traditional PBFT.

Sun et al. [28] proposed an improved algorithm based on network partitioning, which combines the Raft mechanism with PBFT to achieve node consensus. Ling et al. [29] designed a node management module, level division module and consensus module to optimize PBFT layered and reduce the number of consensus in the system. Chen et al. [30] adopted a multi-layer PBFT Consensus protocol to solve the scalability problem of the process of sharing IoT data. Although the above methods have improved the throughput and scalability of the system to a certain extent by reducing the number of consensus nodes and adopting hierarchical and grouped consensus methods, all nodes in this scheme can only complete one task in a specific time, which cannot realize multi-tasking, and the way to select the master node is not safe enough. If the master node is attacked or faulty one after another, the view needs to be replaced frequently, which seriously affects the consensus efficiency and threatens the system security.

Xie et al. [31] used probabilistic language terminology to set confidence intervals for the selection of master nodes to improve the reliability of master nodes. Zhang et al. [32] proposed a node reliability quantification mechanism to select highly reliable nodes to participate in block production. Pang et al. [33] proposed a PBFT algorithm that can check the status of nodes, and reduce the impact of malicious nodes on the system by checking and evaluating the status of master nodes. Liu et al. [34] evaluate the reliability of users through the credit model and voting mechanism, and use this as the basis for selecting the master node. Jiang et al. [35] proposed a comprehensive evaluation model combining the entropy method, TOPSIS method and Borda counting method to select the highest-ranking node as the master node to ensure the security and stability of the blockchain network. Jun et al. [36] use the proposed credit model to select the master node by voting to ensure the reliability of the master node. Qushtom et al. [37] use credit scoring and reward mechanisms to motivate nodes in the system to produce correct behaviour. Li et al. [38] proposed a scalable hierarchical Byzantine fault tolerance algorithm, in which master node selection and impeachment mechanisms are set to ensure the safety of master nodes. Zheng et al. [39] combined the C4.5 algorithm with PBFT, used the decision tree to evaluate the node credit, and introduced the integral voting mechanism to determine the leader node. Although the above methods have improved the reliability of the master node to a certain extent by adding node evaluation models, voting mechanisms, and reliability quantification mechanisms, the master nodes are often served by a few fixed nodes, which reduces the degree of decentralization of the system.

The improved PBFT algorithm mentioned above mainly has the disadvantages that unsafe selection of primary nodes, degraded distributed performance of the system, and only processing a single consensus task per unit time, which cannot meet the requirements of multi-task application scenarios. In addition, it can only guarantee the reliability of the master node or the consensus efficiency of the system, but

cannot achieve both. In view of the above problems, in order to realize the safety and efficiency of the consensus process of the swarm robotic system in emergency scenarios, this paper improves the PBFT algorithm from two aspects of consensus efficiency and node security, grouping nodes based on the distance between nodes, and combining dynamic reputation model to constitute a master-slave multi-chain data storage and consensus structure with high-performance multi-master nodes.

### III. THE PBFT OPTIMIZED CONSENSUS ALGORITHM

In this section, we will detail the design of the improved PBFT algorithm. First, we describe the business model of the swarm robotic system in emergency scenarios and the master-slave multi-chain data storage and consensus model established in this paper, and propose our design goals. We will then focus on the various parts of the improved algorithm and the final consensus process.

#### A. SYSTEM MODEL

##### 1) SWARM ROBOTIC BUSINESS MODEL

This paper aims at a large-scale emergency rescue swarm robotic system, which consists of the base robotic, communication robotics, and business robotics, and its topology is shown in Figure 1. Among them, the distributed self-organizing wireless communication network built by base robotic and communication robotics provides communication services for business robotics. Due to the complexity of the environment in emergency scenarios, the communication between swarm robotics will be affected by the network structure and geographical location. This paper uses blockchain as the data storage method and mainly focuses on how to quickly achieve data consistency for a large number of business robotics in this scenario. The nodes in the blockchain are composed of two types: full nodes and light nodes.

- The full node will record the complete blockchain information, so all transactions on the blockchain can be verified independently, which will be executed by nodes with better performance.
- Light nodes do not store or maintain complete blockchain information, but only store a minimal amount of state to send or transmit transaction messages.

In this paper, business robotics in emergency scenarios are divided into full node robotics and light node robotics. These two types of robotics cooperate to deal with the transactions in the system according to their characteristics. The light node business robotics only processes transaction information in its group, while the full node robotics should cache all the transaction content received in the system locally for verification.

##### 2) DATA STORAGE AND CONSENSUS MODEL OF SWARM ROBOTICS

This paper adopts the master-slave multichain architecture to store data and builds a master chain composed of multiple

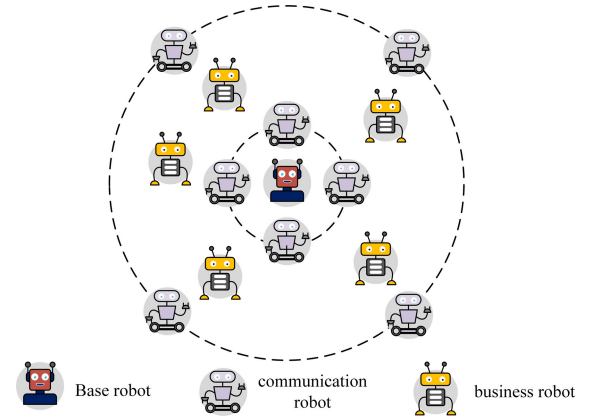


FIGURE 1. The topological structure of the emergency rescue swarm robotic system.

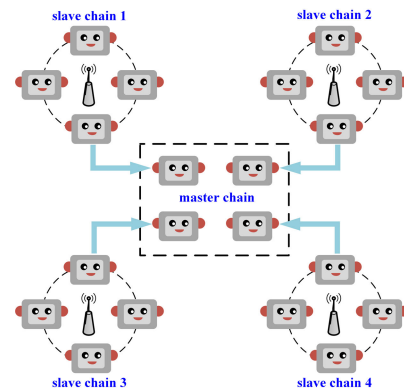


FIGURE 2. The master-slave multichain storage structure.

full-node robotics and multiple slave chains composed of sets of light node robotics. The specific consensus node hierarchy is shown in Figure 2. The light node set is generated according to the real-time position clustering of each robotic, while the full node robotics is served by the nodes with a high reputation in each region. Since the swarm robotic system needs different consensus requirements and time in each area in an emergency scenario, the master-slave chain structure can process various and complex consensus transactions in parallel, effectively improving the throughput of the system and reducing the risk that a single primary node poses to the system.

A channel is composed of a single slave chain and its corresponding primary node, in which only the transaction information in the corresponding region is stored and maintained. Each slave chain jointly maintains a master chain, and nodes on the master chain store all transaction information for verification by other nodes. Using multiple channels to process transactions in each region in parallel, solves the problems of low throughput and high transaction latency of the traditional PBFT consistency algorithm.

#### B. TRADITIONAL PBFT CONSENSUS ALGORITHM

PBFT is a distributed system consensus algorithm that can tolerate Byzantine errors and can guarantee the security



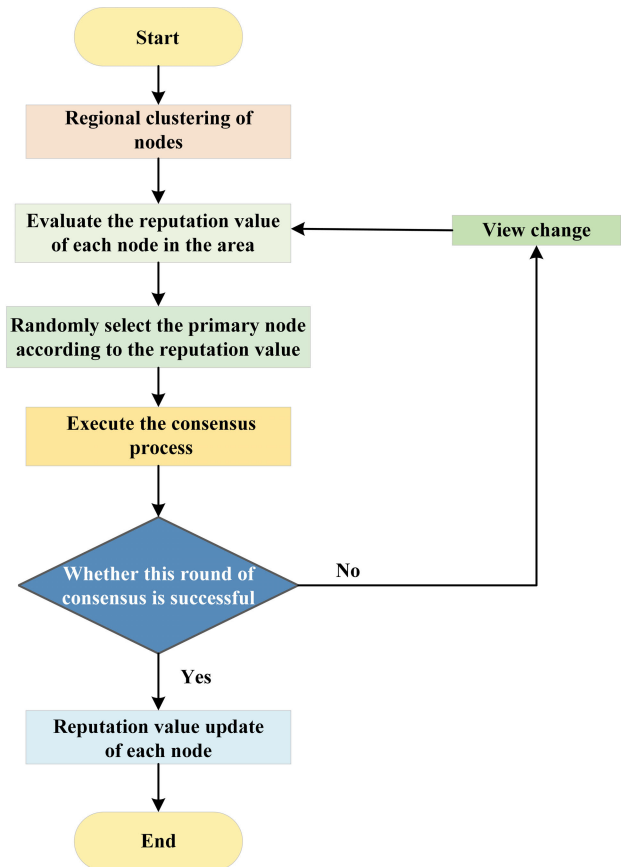
**Algorithm 1** Swarm Robotic Data Storage and Consensus Model Pseudocode**Input:** Swarm robotic system node set**Output:** data storage and consensus success or failure

- 1: Slave chains  $\leftarrow$  Location clustering.
- 2: The master chain  $\leftarrow$  Reputation mechanism.
- 3: A channel  $\leftarrow$  slave chain and its corresponding primary node.
- 4: The master chain  $\leftarrow$  multiple slave chains jointly maintain.
- 5: **if** The client sends the consensus transaction **then**
- 6:     The primary node sends the transaction information to the corresponding slave chain.
- 7: **end if**
- 8: **if** Complete the consensus from the slave chain **then**
- 9:     **while** multiple slave chains completing consensus **do**
- 10:         the master chain is merged to complete consensus.
- 11:     **end while**
- 12: **end if**
- 13: Slave chains store and maintain only the current channel transaction information.
- 14: The master chain stores all the transaction information.

and validity of the system consensus when the number of Byzantine nodes in the system does not exceed one-third. It mainly includes three stages: pre-preparation stage, preparation stage and submission stage. In this consensus process, the client first sends the transaction to the master node, and the master node is responsible for numbering the received transaction, then sending these transactions to other consensus nodes, and finally returning the confirmation information of the consensus node to the client. Although the PBFT consensus algorithm is a Byzantine fault-tolerant algorithm, it can solve the problem of data consistency caused by possible failures or malicious behaviours of nodes in a distributed system. However, there are still some deficiencies in applying it to swarm robotic systems in emergency scenarios:

- 1) Multiple messages passes and calculations are required to reach consensus. This may lead to a high delay for swarm robotic systems in emergency scenarios, thus affecting the response speed and efficiency of robotics.
- 2) There is a single point of failure problem, that is, if the master node fails or is attacked, the operation of the entire system will be affected. In the emergency scene swarm robotic system, due to the large number of robotics, the problem of single point failure may be more prominent. Therefore, it is necessary to design a mechanism to solve the single point of failure problem and ensure the stability and reliability of the entire system.

To sum up, the application of the PBFT consensus algorithm to the swarm robotic system in the emergency scene, it is necessary to solve the performance problem of the

**FIGURE 3.** Optimized single-round consensus algorithm process.

algorithm and the single point of failure problem, improve the stability, reliability and security of the system, so as to ensure the execution efficiency of the whole system.

**C. IMPROVED CONSENSUS ALGORITHM DESIGN FOR SWARM ROBOTICS**

In this paper, aiming at the problems existing in traditional PBFT consensus algorithm in emergency scenarios, such as random selection of primary nodes, high communication overhead, long consensus delay and low throughput, a PBFT optimization consensus algorithm based on K-means clustering is proposed. The consensus nodes in the system are grouped by the K-means clustering algorithm [40] to form multiple slave chains. In addition, master nodes of each region are selected according to the reputation mechanism to form the master chain, and the master chain and multiple slave chains jointly complete the system consensus.

**1) IMPROVED ALGORITHM FLOW**

According to the communication characteristics of the swarm robotic system in emergency scenarios, this paper puts forward a PBFT optimization consensus algorithm given K-means clustering. The main process of this algorithm consists of three sections, and the specific flow is illustrated in Figure 3.

- 1) Slave Chain Formation Stage:

Firstly, the location information of each robotic node is acquired, and it is clustered according to the latitude and longitude of the robotic to form k node clusters of different regions.

2) Main Chain Formation Stage:

Dynamically and comprehensively estimate the performance of every robotic node during the consensus process, and generate corresponding reputation value. In the area formed by each cluster, a node with a high reputation is randomly chosen as the primary node within the threshold. A master chain is formed by the primary nodes of each slave chain.

3) Consensus Execution Stage:

The consensus within the system is composed of two parts: the slave chain consensus and the master chain consensus. After the master-slave chain verifies and votes on the new block, it synchronizes the new block to the local.

2) GROUPING MODEL BASED ON K-MEANS CLUSTERING

Because the situations in emergency scenarios are complex and often involve a wide range, selecting the nodes close to each other for consensus will effectively decrease the communication delay, and also avoid the problems of signal interruption and high communication overhead caused by distance factors. Therefore, in this paper, each node in the swarm robotic system is grouped according to its geographical location and selects the normalized value of the longitude and latitude of each robotic node as the input of K-means clustering, to divide the space. Suppose there is a sample set  $L = \{l_1, l_2, \dots, l_n\}$ , where each sample represents the coordinates of the robotic  $l_i = (y_i, z_i)$ ,  $y_i$  is the longitude of the i-th robotic,  $z_i$  is the latitude of the i-th robotic, and the sample set L includes a total of n robotic samples. Before K-means clustering, first normalize the coordinates of all robotics, assuming that the abscissa set  $Y = \{y_1, y_2, \dots, y_n\}$ , and the ordinate set  $Z = \{z_1, z_2, \dots, z_n\}$ , so the coordinate normalization formula is

$$y'_i = \frac{y_i - y_{\min}}{y_{\max} - y_{\min}} \tag{1}$$

$$z'_i = \frac{z_i - z_{\min}}{z_{\max} - z_{\min}} \tag{2}$$

where  $y_{\max}$ ,  $y_{\min}$  and  $z_{\max}$ ,  $z_{\min}$  is the maximum value and minimum value in the set Y and Z respectively.  $y_i$  and  $z_i$  are original values.  $y'_i$  and  $z'_i$  are the normalized value. The normalized robotic coordinate is  $l'_i = (y'_i, z'_i)$ .

Suppose that according to the normalized coordinate values of each robotics, this paper adopts the K-means algorithm to group it into k clusters  $K = \{K_1, K_2, \dots, K_k\}$ , the specific steps are as follows:

- Randomly select k samples from the sample set L as the initial cluster centers  $\{\mu_1, \mu_2, \dots, \mu_k\}$ .
- Calculate the Euclidean distance  $\lambda_{ij} = \|l_i - \mu_j\|_2$  from each node in the sample set to each cluster centre in

turn, and assign it to the nearest cluster centre, thereby obtaining k clusters.

- Calculate the new cluster centre  $\mu'_i = \frac{1}{|K_i|} \sum_{l \in K_i} l$  in each cluster, where  $K_i$  is the i-th cluster, and  $l$  is the coordinate of the sample robotic node in the cluster  $K_i$ .
- Iterate procedures 2 and 3 until the cluster centre remains unchanged.

The most critical part of the above steps is to determine the value of k, which influences the clustering effect. The sum of the squared errors (SSE), which is an indicator for evaluating the effectiveness of clustering quality, which is defined as follows:

$$SSE = \sum_{i=1}^k \sum_{l \in K_i} \|l - \mu_i\|_2^2 \tag{3}$$

The smaller the SSE value, the better the sample classification effect is. When k is less than the most suitable number of clusters, the aggregation degree of every cluster will increase significantly as k increases and SSE will decrease rapidly. The rate of increase of the aggregation degree of each cluster will slow down, and the decline of SSE will be flat when k is greater than the most suitable number of clusters. Therefore, the most appropriate clustering number is the k value when SSE from a rapid decline to a gentle decline. However, it is necessary to introduce the silhouette coefficient (SC) as another indicator to determine the validity of the clustering quality when the SSE decline trend is not obvious. In the sample set L, the SC of the i-th node is defined as:

$$SC(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{4}$$

Among them:  $a(i)$  and  $b(i)$  are the average distances from the i-th robotic node to other robotic nodes in the same group  $K_i$  and the nearest group  $K_j$ , respectively. The closest group  $K_j$  is defined as:

$$K_j = \arg \min_{K_i \subset \kappa} \frac{1}{|K_i|} \sum_{l \in K_i} \|l - \mu_i\|_2 \tag{5}$$

The SC of the entire sample set L is

$$SC = \frac{1}{n} \sum_{i=1}^n SC(i) \tag{6}$$

The value boundary of SC is  $[-1, 1]$ , and the closer SC is to 1, the higher the clustering quality is. Compared with the previous clustering result, when the position of more than half of the nodes in the system changes, it needs to be clustered again. After each slave chain completes the previous round of consensus, it will participate in the next round of consensus according to the latest clustering results.

3) REPUTATION MECHANISM

In emergency scenarios, the node failure problem caused by an unstable network and communication environment must be considered. Therefore, it is necessary to appraise the

behaviour of every node during realizing data consistency to select the robotic nodes with stable functions and good performance in each area. The selected nodes as primary nodes in their corresponding region accomplish the master chain consensus to enhance the consensus efficiency of the entire system. The dynamic reputation mechanism can also improve the enthusiasm of robotic nodes to take part in the consensus at the same time.

The improved algorithm divides robotic nodes into 3 states according to their behaviour during the consensus process: Normal nodes, Abnormal nodes, and Byzantine nodes, and the reputation score of nodes is increased or decreased differently in each state. The reputation score of the robotic node during the consensus is evaluated as follows:

$$X_i(s+1) = \begin{cases} X_i(s) + (1 - \frac{w}{T}) X_i(s), & \text{Normal nodes} \\ (1 - \frac{t}{T}) X_i(s), & \text{Abnormal nodes} \\ 0, & \text{Byzantine nodes} \end{cases} \quad (7)$$

Among them,  $X_i(s)$  and  $X_i(s+1)$  are the reputation score of each robotic node after the  $s$ -th and the  $s+1$ -th round of consensus respectively,  $w$  represents the time when the node completes the consensus,  $t$  is the time that the robotic node was not involved in the consensus process, and  $T$  is the total time of the current round of consensus participated by the robotic node. For the nodes in the swarm robotic system, the corresponding  $w$  value is, the faster it accomplishes the consensus, the higher its reputation score. The longer the node delay time  $t$ , the greater the influence for the consensus speed, and the lower the reputation score.

As shown in Equation (7), the state of each node is divided into three kinds based on its performance during the consensus process: Normal, Abnormal and Byzantine node state. If the node behaves normally in the consensus process, the system will recognize it as a Normal node and score it according to the rules in the first row. If the node does not join the consensus on time due to its own parts failure or network reasons, the system will score the node according to its delay time and identify it as an Abnormal node. If the node affects the system consensus by malicious acts such as data tampering during the consensus process, its reputation score will be set to 0, and it will be judged as a Byzantine node to exit the system and then checked. In the initial case, a primary node is randomly selected from each area, and then the client sends a test transaction to evaluate the consensus effectiveness of each robotic node, at the same time, generate the initial reputation value of each node. The reputation score of every node is sorted and the selected threshold  $D$  is set before the start of a new round of consensus. The threshold  $D$  is determined according to the reputation value of each node in the slave chain after sorting. Select several nodes with higher reputation value as candidate primary nodes to determine the threshold range of the final

primary node selection. Randomly selecting a robotic node from the threshold range as the primary node in the area, then the primary node of every area together forms the master chain. Accordingly, after the end of each consensus round, the reputation score of each robotic node is updated.

The status of each robotic node determines its role in the next round of consensus, and different roles exercise different powers and take different responsibilities in the consensus process. As shown in Table 2, the roles of each node and its reputation status are defined here.

TABLE 2. Node permission settings.

Reputation value	Reputation status	Node role
Within the threshold $D_q$	Superior	candidate primary node
Outside the threshold $D_q$	Normal	slave node
	Abnormal node with $t < \frac{T}{2}$	
	Abnormal node with $t \geq \frac{T}{2}$ Byzantine node	no participation in nodes

The behaviour of each robotic node is restricted according to the performance in the consensus process. Nodes with excellent performance can be selected into the set of preliminary primary nodes, and nodes with poor performance must exit the system and cannot participate in the consensus. After the robotic node is checked and repaired, its reputation value will be reset to 1, and it will re-enter the system to take part in the consensus. The dynamic reputation model algorithm is shown in Algorithm 2.

**Algorithm 2** The Dynamic Reputation Model

**Input:** ( $X_i, D_q, w, t, Nr_i$ )

**Output:** ( $X_i, Nr_i$ )

```

1:  $i \in N$ 
2: if the node completes this round of consensus normally
   then
3:    $X_i \leftarrow X_i + (1 - \frac{w}{T}) X_i$ ;
4: else  $X_i \leftarrow (1 - \frac{t}{T}) X_i$ ;
5:   if node behaves maliciously then
6:      $X_i \leftarrow 0$ ;
7:   end if
8: end if
9: if  $X_i$  is within  $D_q$  then
10:   $Nr_i \leftarrow$  candidate primary node;
11: end if
12: while  $X_i$  is outside  $D_q$  do
13:  if normal node or Abnormal node with  $t < \frac{T}{2}$  then
14:     $Nr_i \leftarrow$  slave node;
15:  else[Byzantine node or Abnormal node with  $t \geq \frac{T}{2}$ ]
16:     $Nr_i \leftarrow$  no participation in nodes;
17:    after the node is checked and repaired, it re-
    engages in the consensus;
18:  end if
19: end while
20: return ( $X_i, Nr_i$ )

```

4) CONSENSUS EXECUTION PROCESS

1) Slave chain consensus stage:

In this stage, the client first sends the transaction to the primary node in the corresponding area, and then primary nodes broadcast the transaction to each node on its corresponding slave chain. Each slave node confirms the message and then returns the confirmation result to its corresponding primary node. The specific slave chain consensus procedure is as below:

- After the client  $c$  obtains the transaction list, it sorts and packages the transactions and sends them to the primary nodes in the corresponding regions for slave chain consensus. On this basis, the client will also send the packaged transaction to other nodes in the master chain to prepare for the future master chain consensus.
- The primary node  $p_i$  first numbers the data package after it receives the package and sets its number to  $n$ , then appends the view number  $v$  and signature to form a pre-prepare stage message packet  $\langle (pre - prepare, v, n, d(m), \lambda), \sigma_{p_i}, m \rangle$ , and broadcasts the packet to each node in the corresponding slave chain, where  $d(m)$  and  $\sigma_{p_i}$  are the digest of the message  $m$  and the signature of the primary node respectively.
- Each slave node verifies the packet information in the view after receiving the pre-prepare data packet  $\langle (prepare, v, n, d(m), t), \sigma_t \rangle$  sent by the primary node. If the information is correct, the preparation phase begins. Every slave node will send a prepare data packet to other nodes in this slave chain, where  $t$  and  $\sigma_t$  represent the  $t$ -th slave node of the slave node set and its signature respectively.
- The nodes from the slave chain check up the received prepare packets and send a confirmation information to other slave nodes after passing. Meanwhile, it will collect  $2w + 1$  ( $w$  is the number of Byzantine nodes that can be tolerated in the slave chain) commit packets sent by other nodes. If the verification passes and a corresponding number of confirmation information is received, the reply phase begins.
- The primary node  $p_i$  will bear out that the transaction is effective after it receives more than  $2w + 1$  replies, and broadcasts this message to other primary nodes in the master chain. If there is a problem with the primary node, the nodes will send a view change message. The slave chain will replace the view, update the reputation score, and reselect the primary node in this region.

2) Master chain consensus stage:

The primary node in each area feeds back the message of successful consensus to other primary nodes in the master chain for verification after the consensus of the slave chain is accomplished. After the verification is passed, the result is feedback to the client. This round

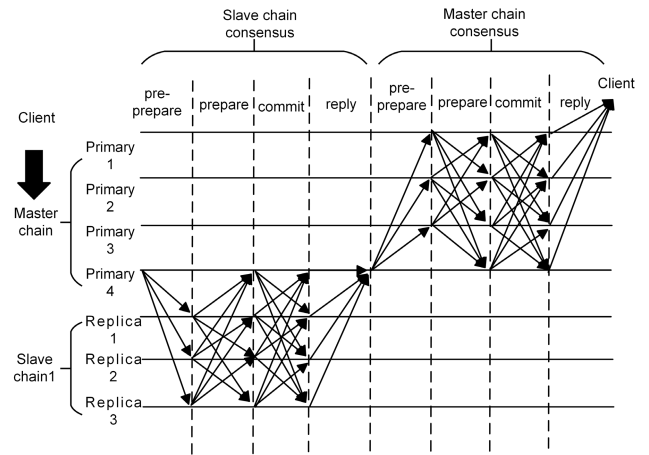


FIGURE 4. The improved PBFT consensus process.

TABLE 3. Description of relevant symbols in this section.

Symbol	Description
$N$	the total number of nodes in the system
$m_i$	the number of nodes in the $i$ -th slave chain
$T_{c_{max}}$	maximum communications times of slave chain
$T_{z_{max}}$	maximum communications times of master chain
$T_{max}$	total maximum times of communications
$B$	the ratio of PBFT to the maximum communications times after optimization

of consensus is successful when the client receives  $2o + 1$  ( $o$  is the number of Byzantine nodes that can be tolerated in the master chain) reply messages from the master chain. Since each slave chain performs different transaction processing, the processing time of the transaction may be different. When conducting consensus in the master chain stage, if multiple slave chains have master chain consensus requirements at the same time, the master chain consensus tasks of these slave chains can be packaged and merged to jointly complete the master chain consensus, to reduce the consensus frequency of the master chain and the communication cost of the system. The specific process of the improved PBFT consensus process is illustrated in Figure 4.

IV. ALGORITHM ANALYSIS AND SIMULATION VERIFICATION

In this paper, theoretical analysis and simulation verification are carried out for the traditional PBFT consensus algorithm and the optimized algorithm. The theoretical analysis includes two sections: the communication overhead and the stability of the swarm robotic system. The simulation verification part is to test and compare the transaction delay, throughput, and stability of different algorithms under the same hardware and software environment. For the sake of clarity, the relevant symbols involved in this section are explained as follows in Table 2.



**TABLE 4. Communication times of PBFT and improved PBFT.**

	PBFT	Improved PBFT	
		slave chain consensus	master chain consensus
pre-prepare	$N - 1$	$m_i - \sum_{i=1}^k m_i$	$k(k - 1)$
prepare	$(N - 1)^2$	$m_i^2 - \sum_{i=1}^k m_i^2$	$k(k - 1)^2$
commit	$N(N - 1)$	$m_i(m_i - 1) \rightarrow \sum_{i=1}^k m_i(m_i - 1)$	$k^2(k - 1)$

**A. ALGORITHM ANALYSIS**

1) COMMUNICATION OVERHEAD ANALYSIS

For a swarm robotic system, the communication overhead is determined by the quantity of communications between robotics and the time required for communication. The improved algorithm uses the geographic location of the robotics to cluster, which shortens the communication distance of each robotic in the subcluster, reduces the communication delay, and reduces the consensus frequency between nodes through the master-slave chain storage structure. The communication times required for PBFT and the improved consensus algorithm in this paper to accomplish a round of consensus are listed in Table 4.

During the traditional PBFT consensus process, the specific calculation process consists of three stages. The communication times in the pre-preparation stage and the preparation stage are  $(N - 1)$  and  $(N - 1)^2$  times respectively. All slave nodes validate the received prepare data packet during the commit stage. If the validation result is correct, the slave node will broadcast a confirmation message to residual nodes. The communication times at the stage is  $N(N - 1)$ . So the total communication times of the PBFT consensus algorithm is  $2N(N - 1)$ .

In the improved consensus algorithm, assuming that the number of slave chains after clustering is  $k$ , the corresponding quantity of primary nodes in the master chain is also  $k$ , and the quantity of robotic nodes in every area determines the number of nodes  $m_i$  in each slave chain, which is generally not less than 4.

Due to the different consensus requirements of robotics in each area, when robotic nodes in all areas participate in the consensus, the required communication times are the most, and the maximum communication times of the slave and master chain consensus phase are respectively:

$$T_{c_{max}} = \sum_{i=1}^k m_i + \sum_{i=1}^k m_i^2 + \sum_{i=1}^k m_i(m_i - 1) \quad (8)$$

$$T_{z_{max}} = 2k^2(k - 1) \quad (9)$$

Then the maximum times of communication after improvement is:

$$T_{max} = T_{c_{max}} + T_{z_{max}} = N - k + 2k^2(k - 1) + \sum_{i=1}^k m_i^2 + \sum_{i=1}^k m_i(m_i - 1) \quad (10)$$

$$B = \frac{2N(N - 1)}{N - k + 2k^2(k - 1) + \sum_{i=1}^k m_i^2 + \sum_{i=1}^k m_i(m_i - 1)} \quad (11)$$

The ratio of the maximum communication times between PBFT and the improved algorithm is listed in Equation (11). When  $k = 1$  and the number of summary points  $N$  remains unchanged, the total communication times of the improved algorithm remain unchanged. When  $k$  continues to increase, the value of  $B$  will also increase, but it will start to decrease when it increases to an extreme point. Therefore, the value of  $k$  should not be too large, and the extreme point and the  $k$  value near the extreme point should be used for clustering.

2) STABILITY ANALYSIS

First of all, before the swarm robotics start a consensus, the reputation of each robotic will be evaluated. The nodes in the master chain are composed of multiple nodes with high reputations, and the multiple primary nodes supervise each other, which reduces the risk of the system caused by primary node failure. At the same time, owing to the strong mobility of robotics, the position of robotic nodes may change after each task is accomplished. Re-clustering based on the position of each node in the system will increase the randomness of the primary node election in each region. Moreover, the primary node is randomly selected within the set threshold, which tremendously increases the difficulty of malicious attacks on the system and enhances the robustness of the system. Finally, using the distance advantage of each node in the slave chain to set up micro base stations or ad hoc networks in each area can improve the stability of signal transmission and reduce the transmission delay of signals.

**B. SIMULATION VERIFICATION**

In order to evaluate the performance of the various algorithms before and after optimization, we use the Go language to simulate the consensus process. The experimental environment is Intel I7-8550U CPU, 8GB memory, 64-bit Win11 operating system, and Go language version is 1.18.3.

The experiment mainly tests the throughput, transaction delay, and stability of PBFT and the improved consensus algorithm. In the transaction delay and throughput tests, the quantity of primary nodes  $k$  is set to 4, 6, and 9 in this paper. The throughput and latency of the four consensus algorithms are compared under different total node numbers. Then, different numbers of Byzantine nodes were set to test the stability of the two algorithms. To decrease the experimental error, each experiment takes an average of 20 experimental results in the same software and hardware environment as the final result.

1) TRANSACTION LATENCY TEST

The time from the client sending the transaction task to the system completing the consensus and finally returning the confirmation result to the client is defined as transaction

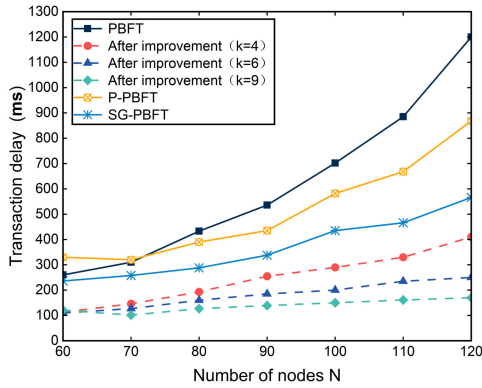


FIGURE 5. Time delay comparison of different algorithms.

latency [41]. The lower the transaction delay, the faster the consensus speed of the nodes. Correspondingly, the higher the consensus efficiency of the system. The experiment compares the latency of PBFT, P-PBFT [42], SG-PBFT [43] and the improved consensus algorithm with  $k$  values of 4, 6, and 9 in this paper. The experimental results are illustrated in Figure 5.

The experimental results in Figure 5 show that the improved algorithm in this paper is better than the other three algorithms in terms of transaction delay, and the difference between the improved algorithm and the other three algorithms also increases gradually with the increase of the total number of nodes in the system. It is mainly because the improved algorithm divides the nodes into  $k$  slave chains, each slave chain processes transactions in parallel, reducing the consensus times between nodes. Most of the consensus processes in the system are completed by the nodes close together, so the communication time between them is short and relatively stable. Thus, greatly reducing the transaction delay. With the increase of  $k$  value, the transaction delay also decreases and gradually tends to balance.

## 2) THROUGHPUT TEST

The quantity of transaction tasks that the system can tackle in a unit of time is called throughput [44]. The higher the throughput, the higher the transaction processing efficiency of the system. The experiment compares the throughput of the four algorithms under different total node numbers of the system. The experimental results are illustrated in Figure 6.

The experimental results in Figure 6 show that with the rise of the total number of nodes, the throughput of the four algorithms is also decreasing, but the improved algorithm proposed in this paper changes gently and is always higher than the other three algorithms. At the same time, along with the increase of the  $k$  value, the number of transactions that the system can handle is also rising. Due to the master-slave chain structure, multiple slave chains process different transactions in parallel, which reduces the times of communication between nodes and enhances the speed of transaction processing, so the swarm robotic system can complete more transactions per unit of time. And before

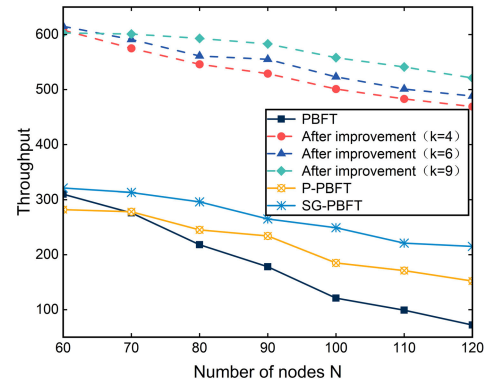


FIGURE 6. Throughput comparison of different algorithms.

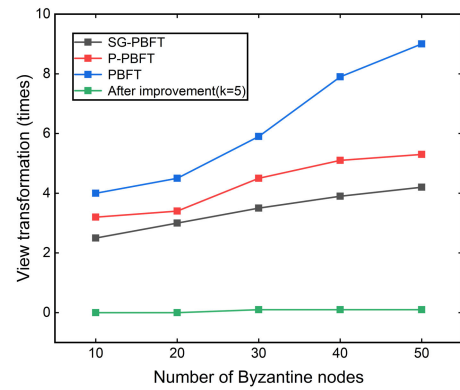


FIGURE 7. Comparison of the number of view transitions.

the  $k$  reaches the extreme value, with the increase of  $k$  value, the consensus efficiency of the system continues to increase.

## 3) STABILITY TEST

Stability is one of the indicators to measure the performance of the consensus algorithm. In the system with Byzantine nodes, the stability of the system can be expressed by the number of view transitions and the size of the system transaction delay under different numbers of Byzantine nodes. Therefore, the total number of nodes in this group of experiments is set to 200 to test the performance of several algorithms with different numbers of Byzantine nodes included in the system, and the experimental results are shown in Figure 7 and Figure 8.

There is only one primary node in the traditional PBFT consensus algorithm. If there is a problem with the primary node, the view needs to be changed frequently, which affects the stability and consensus efficiency of the system. As shown in Figure 7, as the number of Byzantine nodes in the system increases, its view transformation grows faster, followed by the other two algorithms. The number of view transformations of the improved algorithm is the least, and it can keep the number of view transformations very low even in the case of more Byzantine nodes. This is because the improved algorithm randomly selects the high-performance primary nodes through the reputation evaluation model,

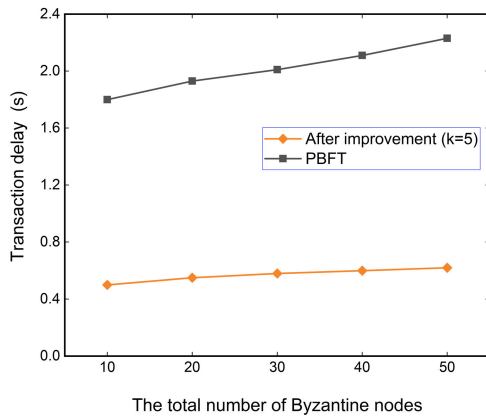


FIGURE 8. Comparison of transaction delay.

which greatly reduces the possibility of the primary node being a Byzantine node. At the same time, it also avoids the view transformation caused by malicious nodes predicting the identity of the primary node to launch attacks and enhances the robustness of the system.

From Figure 8, information can be obtained, that when the system contains the same number of Byzantine nodes, the optimized algorithm reduces transaction delay by 68.4% compared with the PBFT algorithm. As the quantity of Byzantine nodes increases, the improved algorithm is relatively stable. It is mainly because the traditional PBFT consensus algorithm needs large amounts of consensus times to reach an agreement, while the improved algorithm first conducts consensus in each area, and after the consensus is successful, the master chain consists of a small number of high-performance primary nodes confirms the result. The adoption of a reputation mechanism greatly reduces the error probability in consensus. At the same time, the faulty node only needs to be responsible for the corresponding area. Even if the view needs to be changed, only the nodes in this area participate, and other nodes do not need to participate, which effectively reduces the transaction delay.

#### 4) CONSENSUS SUCCESS RATE TEST

In the PBFT consensus algorithm, the number of Byzantine nodes has a direct impact on the consensus success rate, which is not more than 1/3 in general. In particular, if the Byzantine node participates in the consensus as the primary node, it will have a serious impact on the security of the consensus and may lead to the failure of the consensus and waste a lot of communication resources. Therefore, we set  $N$  as 100 and take  $k$  as 5 to compare and test the consensus performance of the improved algorithm with the other three PBFT under different numbers of Byzantine nodes. The consensus success rate of each algorithm with different Byzantine nodes is shown in Figure 9.

The final experimental results are shown in Figure 9. It can be seen that with the increase of the number of Byzantine nodes in the system, the success rate of the improved algorithm in this paper is always higher than that of the other

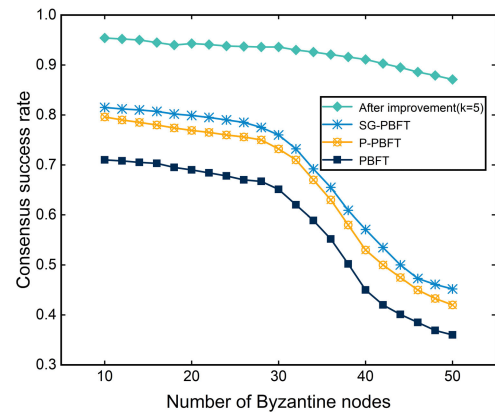


FIGURE 9. Comparison of the consensus success rate of PBFT before and after improvement.

three algorithms, and the success rate of the PBFT, P-PBFT and SG-PBFT algorithms is increased by 37.05%, 22.97% and 18.36% respectively. At the same time, the upper limit of fault tolerance for Byzantine nodes is more than 1/3 of the total number of system nodes. When the number of Byzantine nodes is more than 1/3, the consensus success rate of the other three algorithms decreases rapidly, but the success rate of the improved algorithm is relatively stable. Therefore, it can be proved that the fault tolerance rate and consensus performance of the proposed algorithm are higher. This is because the primary node is selected through the reputation scoring mechanism, which ensures the reliability of the primary node and reduces the degree of node centralization and the possibility of Byzantine nodes becoming primary nodes. In addition, due to the grouping model, Byzantine nodes only have an impact on the corresponding group, which effectively reduces their impact on the overall consensus efficiency of the system.

As can be seen from the above, the selection of  $k$  value determines the communication times between nodes, thus affecting the communication time and throughput of the system, and at the same time, the selection of  $k$  value also greatly affects  $SSE$  (clustering effect).  $SSE$  determines the communication distance between nodes and the communication environment, thus affecting the communication time and communication stability between nodes. Therefore, it is necessary to comprehensively consider the situation of the two. If there is little difference between the optimal  $k$  values of the two, the integer approach to the average value of the  $k$  values of the two is taken as the final clustering number. However, if there is a large difference between the optimal  $k$  values of the two, the discussion can be divided into two cases: if the distance between nodes in this scenario is far, emphasis should be placed on choosing the  $k$  value that is close to  $SSE$  to reach the optimal value, because compared with the number of communication times, the communication distance between nodes and the communication environment have a much greater impact on the system consensus. If the distance between nodes is relatively close, the  $k$  value that makes the communication frequency reach the optimal is

preferred, because the distance between nodes has very little influence on the system consensus at this time.

## V. CONCLUSION

In order to solve the problems of high communication overhead, long consensus delay, and low throughput that exist in the swarm robotic system using the traditional practical Byzantine fault-tolerant algorithm to achieve distributed consensus in emergency scenarios, this paper proposes a grouping practical Byzantine fault-tolerant quadratic consensus algorithm based on K-means clustering. Each node is clustered into several slave chains based on the real-time geographic location. Because each slave chain has the advantage of distance, the consensus delay is shorter, and each slave chain completes different transactions in parallel, which reduces the communications times between nodes and improves the efficiency of consensus.

Using the reputation scoring mechanism to elect nodes with high reputations in each area as the primary node to accomplish the master chain consensus, avoiding the waste of communication resources caused by the failure of a single primary node.

The experimental results make known, that the performance of the improved consensus algorithm is better than the PBFT consensus algorithm in transaction delay, system throughput, and communication overhead. With the continuous application of swarm robotics, in the following work, further research will be done on the clustering method of consensus nodes to achieve a more efficient consensus of the swarm robotic system.

## REFERENCES

- [1] R. Nelson and E. Lima, "Effectuations, social bricolage and causation in the response to a natural disaster," *Small Bus. Econ.*, vol. 54, no. 3, pp. 721–750, Mar. 2020.
- [2] A. Khan, S. Gupta, and S. K. Gupta, "Multi-hazard disaster studies: Monitoring, detection, recovery, and management, based on emerging technologies and optimal techniques," *Int. J. Disaster Risk Reduction*, vol. 47, Aug. 2020, Art. no. 101642.
- [3] H. Lu, M. Chen, and W. Kuang, "The impacts of abnormal weather and natural disasters on transport and strategies for enhancing ability for disaster prevention and mitigation," *Transp. Policy*, vol. 98, no. 1, pp. 2–9, Nov. 2020.
- [4] Y. Feng and S. Cui, "A review of emergency response in disasters: Present and future perspectives," *Natural Hazards*, vol. 105, no. 1, pp. 1109–1138, Jan. 2021.
- [5] J. Ribeiro, R. Lima, T. Eckhardt, and S. Paiva, "Robotic process automation and artificial intelligence in industry 4.0—A literature review," *Proc. Comput. Sci.*, vol. 181, pp. 51–58, Jan. 2021.
- [6] M. Schranz, M. Umlauf, M. Sende, and W. Elmenreich, "Swarm robotic behaviors and current applications," *Frontiers Robot. AI*, vol. 7, no. 2, p. 36, Apr. 2020.
- [7] Z. Feng, G. Hu, Y. Sun, and J. Soon, "An overview of collaborative robotic manipulation in multi-robot systems," *Annu. Rev. Control*, vol. 49, no. 1, pp. 113–127, 2020.
- [8] S. Singh, A. S. M. S. Hosen, and B. Yoon, "Blockchain security attacks, challenges, and solutions for the future distributed IoT network," *IEEE Access*, vol. 9, pp. 13938–13959, 2021.
- [9] R. Huo, S. Zeng, Z. Wang, J. Shang, W. Chen, T. Huang, S. Wang, F. R. Yu, and Y. Liu, "A comprehensive survey on blockchain in industrial Internet of Things: Motivations, research progresses, and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 88–122, 1st Quart., 2022.
- [10] L. Cao, "Decentralized AI: Edge intelligence and smart blockchain, metaverse, web3, and DeSci," *IEEE Intell. Syst.*, vol. 37, no. 3, pp. 6–19, May 2022.
- [11] X. Zheng and W. Feng, "Research on practical Byzantine fault tolerant consensus algorithm based on blockchain," *J. Phys., Conf. Ser.*, vol. 1802, no. 3, pp. 032022–032031, 2021.
- [12] J. Li, X. Li, H. Zhao, B. Yu, T. Zhou, H. Cheng, and N. Sheng, "MAN-DALA: A scalable blockchain model with mesh-and-spoke network and H-PBFT consensus algorithm," *Peer-Peer Netw. Appl.*, vol. 16, no. 1, pp. 226–244, Jan. 2023.
- [13] J. Zhang, Y. Yang, D. Zhao, and Y. Wang, "A node selection algorithm with a genetic method based on PBFT in consortium blockchains," *Complex Intell. Syst.*, vol. 9, no. 3, pp. 3085–3105, Jun. 2023.
- [14] A. Thakur, S. Sahoo, A. Mukherjee, and R. Halder, "Making robotic swarms trustful: A blockchain-based perspective," *J. Comput. Inf. Sci. Eng.*, vol. 23, no. 6, pp. 60803–60813, Dec. 2023.
- [15] S. H. Alsamhi and B. Lee, "Blockchain-empowered multi-robot collaboration to fight COVID-19 and future pandemics," *IEEE Access*, vol. 9, pp. 44173–44197, 2021.
- [16] H. Hamledari and M. Fischer, "Construction payment automation using blockchain-enabled smart contracts and robotic reality capture technologies," *Autom. Construct.*, vol. 132, Dec. 2021, Art. no. 103926.
- [17] N. Malsa, V. Vyas, J. Gautam, R. N. Shaw, and A. Ghosh, "Framework and smart contract for blockchain enabled certificate verification system using robotics," in *Machine Learning for Robotics Applications*, M. Bianchini, M. Simic, A. Ghosh, and R. N. Shaw, Eds. Singapore: Springer, 2021, pp. 125–138.
- [18] S. Salimi, J. P. Queralta, and T. Westerlund, "Hyperledger fabric blockchain and ROS2 integration for autonomous mobile robots," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Atlanta, GA, USA, Jan. 2023, pp. 1–8.
- [19] T. Alladi, V. Chamola, N. Sahu, and M. Guizani, "Applications of blockchain in unmanned aerial vehicles: A review," *Veh. Commun.*, vol. 23, Jun. 2020, Art. no. 100249.
- [20] V. Vasylykovskiy, S. Guerreiro, and J. S. Sequeira, "BlockRobot: Increasing privacy in human robot interaction by using blockchain," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Rhodes, Greece, Nov. 2020, pp. 106–115.
- [21] R. Gupta, U. Thakker, S. Tanwar, M. S. Obaidat, and K.-F. Hsiao, "BITS: A blockchain-driven intelligent scheme for telesurgery system," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Hangzhou, China, Oct. 2020, pp. 1–5.
- [22] A. Allouch, O. Cheikhrouhou, A. Koubâa, K. Toumi, M. Khalgui, and T. N. Gia, "UTM-chain: Blockchain-based secure unmanned traffic management for Internet of Drones," *Sensors*, vol. 21, no. 9, p. 3049, Apr. 2021.
- [23] X. Yang, Y. Zhang, S. Wang, B. Yu, F. Li, and Y. Li, "LedgerDB: A centralized ledger database for universal audit and verification," in *Proc. 46th Int. Conf. Very Large Data Bases*, Tokyo, Japan, 2020, vol. 13, no. 12, pp. 3138–3151.
- [24] X. Yang, R. Zhang, C. Yue, Y. Liu, B. Ooi, Q. Gao, Y. Zhang, and H. Yang, "VeDB: A software and hardware enabled trusted relational database," in *Proc. ACM SIGMOD/PODS Int. Conf. Manage. Data*, Seattle, WA, USA, 2023, vol. 1, no. 2, pp. 1–27.
- [25] U. S. P. Srinivas Aditya, R. Singh, P. K. Singh, and A. Kalla, "A survey on blockchain in robotics: Issues, opportunities, challenges and future directions," *J. Netw. Comput. Appl.*, vol. 196, Dec. 2021, Art. no. 103245.
- [26] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A scalable multi-layer PBFT consensus for blockchain," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 5, pp. 1146–1160, May 2021.
- [27] J. Yang, Z. Jia, R. Su, X. Wu, and J. Qin, "Improved fault-tolerant consensus based on the PBFT algorithm," *IEEE Access*, vol. 10, pp. 30274–30283, 2022.
- [28] G. Sun, K. Xu, J. Wu, X. Wang, and Z. Liu, "Optimization of PBFT consensus algorithm in piecewise blockchain," in *Proc. 2nd Int. Conf. Big Data, Artif. Intell. Risk Manage. (ICBAR)*, Xi'an, China, Nov. 2022, pp. 20–23.
- [29] H. Ling, F. Wu, J. Chang, H. Liu, and X. Wu, "Alliance chain management system and methods for personal files based on improved multi-layer PBFT," in *Proc. 4th Int. Conf. Blockchain Technol. Appl.*, Xi'an, China, Dec. 2021, pp. 31–38.
- [30] X. Chen, J. Yang, and J. Qiu, "Blockchain-empowered high-frequency spectrum management IoT: A multilayer PBFT consensus perspective," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–17, Apr. 2022.



- [31] M. Xie, J. Liu, S. Chen, G. Xu, and M. Lin, "Primary node election based on probabilistic linguistic term set with confidence interval in the PBFT consensus mechanism for blockchain," *Complex Intell. Syst.*, vol. 9, no. 2, pp. 1507–1524, Apr. 2023.
- [32] Z. Zhang, D. Zhu, and W. Fan, "QPBF: Practical Byzantine fault tolerance consensus algorithm based on quantified-role," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Guangzhou, China, Dec. 2020, pp. 991–997.
- [33] Z. Pang, Y. Yao, Q. Li, X. Zhang, and J. Zhang, "Electronic health records sharing model based on blockchain with checkable state PBFT consensus algorithm," *IEEE Access*, vol. 10, pp. 87803–87815, 2022.
- [34] S. Liu, R. Zhang, C. Liu, and D. Shi, "P-PBFT: An improved blockchain algorithm to support large-scale pharmaceutical traceability," *Comput. Biol. Med.*, vol. 154, Mar. 2023, Art. no. 106590.
- [35] W. Jiang, X. Wu, M. Song, J. Qin, and Z. Jia, "Improved PBFT algorithm based on comprehensive evaluation model," *Appl. Sci.*, vol. 13, no. 2, p. 1117, Jan. 2023.
- [36] J. Chen, X. Zhang, and P. Shangguan, "Improved PBFT algorithm based on reputation and voting mechanism," *J. Phys., Conf. Ser.*, vol. 1486, no. 3, Apr. 2020, Art. no. 032023.
- [37] H. Qushtom, J. Mišić, V. B. Mišić, and X. Chang, "A two-stage PBFT architecture with trust and reward incentive mechanism," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11440–11452, Jul. 2023.
- [38] Y. Li, L. Qiao, and Z. Lv, "An optimized Byzantine fault tolerance algorithm for consortium blockchain," *Peer-Peer Netw. Appl.*, vol. 14, no. 5, pp. 2826–2839, Sep. 2021.
- [39] X. Zheng, W. Feng, M. Huang, and S. Feng, "Optimization of PBFT algorithm based on improved C4.5," *Math. Problems Eng.*, vol. 2021, pp. 1–7, Mar. 2021.
- [40] D. Yu, H. Xu, C. L. P. Chen, W. Bai, and Z. Wang, "Dynamic coverage control based on K-means," *IEEE Trans. Ind. Electron.*, vol. 69, no. 5, pp. 5333–5341, 2022.
- [41] A. S. M. S. Hosen, S. Singh, P. K. Sharma, U. Ghosh, J. Wang, I.-H. Ra, and G. H. Cho, "Blockchain-based transaction validation protocol for a secure distributed IoT network," *IEEE Access*, vol. 8, pp. 117266–117277, 2020.
- [42] G. Xu, H. Bai, J. Xing, T. Luo, N. N. Xiong, X. Cheng, S. Liu, and X. Zheng, "SG-PBFT: A secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent Internet of vehicles," *J. Parallel Distrib. Comput.*, vol. 164, no. 1, pp. 1–11, Jun. 2022.
- [43] Y. Wu, P. Song, and F. Wang, "Hybrid consensus algorithm optimization: A mathematical method based on POS and PBFT and its application in blockchain," *Math. Problems Eng.*, vol. 2020, pp. 1–13, Apr. 2020.
- [44] D. Xenakis, A. Tsiota, C.-T. Koulis, C. Xenakis, and N. Passas, "Contract-less mobile data access beyond 5G: Fully-decentralized, high-throughput and anonymous asset trading over the blockchain," *IEEE Access*, vol. 9, pp. 73963–74016, 2021.



**YI SUN** was born in March 1972. He received the Ph.D. degree in electrical engineering from Xi'an Jiaotong University, in May 2001. He has been a Professor with the School of Information and Communication Engineering, Xi'an University of Science and Technology, mainly engaged in the teaching and research of distributed systems and swarm intelligence technology. In recent years, he has presided over and participated in more than 20 scientific research projects at all levels,

published more than 30 papers in core journals, edited and published five textbooks and monographs, authorized more than ten invention patents, and guided undergraduate and graduate students to win national and provincial science and technology competition awards for many times. His main research interests include the network structure and evolution mechanism of group robot systems in emergency scenarios, such as the research on resource allocation game strategy of swarm robot cooperation and the research on network game mechanism of group robots.



**YING FAN** received the bachelor's degree in communication engineering and the master's degree in electronic information from the Xi'an University of Science and Technology, in July 2021 and September 2021, respectively. Her main research interests include consensus algorithms, consensus efficiency, and data security of swarm robot systems in emergency scenarios.

...