Copyright ⓒ2017 KSII

A Novel Multi-link Integrated Factor Algorithm Considering Node Trust Degree for Blockchain-based Communication

Jiao Li [1,2], Gongqian Liang[1], Tianshi Liu[2]
[1] School of Management, Northwestern Polytechnical University
Xi'an, China
[e-mail: lijiao@xsyu.edu.cn , lgqian@nwpu.edu.cn]
[2] School of Computer Science, Xi'an Shiyou University
Xi'an, China
[e-mail: liutianshi@ xsyu.edu.cn]
*Corresponding author: Jiao Li

Received November 1, 2016; revised March 10, 2017; accepted April 17, 2017; published August 31, 2017

Abstract

A blockchain is an underlying technology and basic infrastructure of the Bitcoin system. At present, blockchains and their applications are developing rapidly. However, the basic research of blockchain technology is still in the early stages. The efficiency and reliability of blockchain communication is one of the research problems that urgently need to be studied and addressed. Existing algorithms may be less feasible for blockchain-based communication because they only consider a single communication factor (node communication capability or node trust degree) and only focus on a single communication performance parameter(communication time or communication reliability). In this paper, to shorten the validation time of blockchain transactions and improve the reliability of blockchain-based communication, we first establish a multi-link concurrent communication model based on trust degree, and then we propose a novel integrated factor communication tree algorithm (IFT). This algorithm comprehensively considers the node communication link number and the node trust degree and selects several nodes with powerful communication capacity and high trust as the communication sources to improve the concurrency and communication efficiency. Simulation results indicate that the IFT algorithm outperforms existing algorithms. A blockchain communication routing scheme based on the IFT algorithm can increase communication efficiency by ensuring communication reliability.

Keywords: Blockchain, trust degree, multi-link, communication tree, communication performance parameters

This research was supported by Natural Science Foundation of Shaanxi Province (2012JQ8040, 2012JM8037), Project of Science and Technology Bureau of Shaanxi Province(2016GY132) and Scientific Research Foundation of Education Bureau of Shaanxi Province(15JK1571, 15JK1586).

https://doi.org/10.3837/tiis.2017.08.001 **ISSN : 1976-7277**

1. Introduction

Blockchain technology has various advantageous characteristics, such as decentralization, non-forgery, tamper-resistance, low operating costs, etc., and thus blockchain technology has attracted wide attention from the academic research community and industry fields. Blockchain is essentially a type of decentralized P2P computing model. Blockchain can allow nodes to exchange information freely and collaborate flexibly without a central node's participation [1-4]. Blockchain has wide application prospects in finance, security, global payment, trading, etc. [5]. However, the basic research of blockchain technology has lagged. By investigating the literature, we find that there are few entries on blockchain, and thus we consider blockchain to be a new technology that is still in its infancy.

The Bitcoin system is an example of a blockchain application. Compared with traditional bank transactions, Bitcoin's blockchain does not require a third party to finish management, settlement and audit, and a transaction generated in a node must be distributed to other nodes for validation and confirmation [6-8]. At present, transactions in blockchain can be validated and confirmed once every 10 minutes, and blockchain can process only 8 transactions per second, so blockchain is unsuitable for the high-frequency trades of many business applications [9]. To shorten the validation time of transactions and improve the business processing capability of blockchain, how to organize all nodes to finish validation quickly is one of the key problems in blockchain research. Due to blockchain using a P2P network, the traditional line or star topology relies on a single communication source, and in the process of communication, a bottleneck may result because of breakdowns or failures of the central node. The tree topology belongs to a hierarchical structure in which the failures of individual nodes cannot cause communication interruptions [10-13]. In actual communication, according to the physical network topology, a tree rooted from a communication source node is established by all other nodes in blockchain to finish the validation of transactions. Therefore, communication tree algorithm research is the key to addressing the efficiency of blockchain-based communication.

This paper proposes the integrated factor communication tree algorithm, named IFT for short, which comprehensively considers communication factors based on the multi-link concurrent communication model. This algorithm provides a routing scheduling scheme for blockchain-based communication, and we wish to complement the existing basic blockchain theory. The concrete contributions of this paper are described below.

(1) The multi-link concurrent communication model based on trust degree is established. The proposed model has various characteristics. (i) It introduces a concurrent communication mechanism, which is meant to assign proper tasks to other nodes besides the communication source node. The node must quickly send data to other nodes without intermittence or delay over the next communication stage, provided that it has received data. Thus, the communication concurrency is high. (ii) Using a tree topology for blockchain-based communication, this model avoids the bottleneck effect, which will occur when a failure befalls individual nodes or their data is polluted. Thus, the proposed model has good malfunction isolation and pollution isolation. (iii) Considering the nodes' difference in communication-forwarding capacity, the model belongs to multi-link communication. Finally, (iv) by introducing the concept of node trust degree, the model has high reliability because honest nodes are selected to undertake forwarding tasks.

(2) An integrated factor communication tree algorithm is proposed, which comprehensively considers node communication link number and node trust degree. This algorithm gives priority to nodes with powerful communication capability and high trust degree as the communication source to shorten the validation time of blockchain transactions and to improve the reliability of communication links.

(3) The proposed algorithm is compared with existing algorithms (LFT, TFT) through theoretical proofs and simulation. The theoretical analysis and simulation results show that the proposed algorithm performs better in overall performance parameters.

The remainder of this paper is organized as follows. In Section 2, we describe the process of blockchain transaction, and then we find and analyze the efficiency problems that exist in blockchain-based communication. In Section 3, we establish the multi-link concurrent communication model based on trust degree. Based on the model, the IFT algorithm is proposed in Section 4. Section 5 first gives theorems for the communication tree, then analyzes and evaluates the communication performance of the proposed algorithm through extensive simulations, and finally compares the proposed algorithm with the other existing algorithms. We conclude this paper in Section 6.

2. Related Work

The aim of this paper is to maximize the communication efficiency of block data validation on the basis of guaranteeing the security and reliability of blockchain transactions.

A blockchain adopts P2P technology in the view of network structure. Each node in a blockchain is a service requester and a service provider. The block data is generated by one node and then must be forwarded one by one to all other nodes for validation of the correctness, so most nodes may be service providers during transaction validation. It is not guaranteed that the block data provided by nodes are real and reliable, so we have to consider node reliability during the entire process of blockchain communication. For this reason, the concept of a trust degree for nodes is introduced in this paper. In addition, at present, it takes 10 minutes to validate and confirm the transactions in blockchain, and new block data is produced every 10 minutes. However, this is too long for business applications with frequent transactions. Thus, shortening the validation time is worth studying. Next, for the entire process of a blockchain transaction, we analyze the reason that it takes so long to finish validation and find the most time-consuming steps in blockchain transactions.

Fig. 1 shows the entire transaction process of a blockchain from generating a transaction to finishing validation. The steps are as follows.

Step 1: collect non-validated transactions in the blockchain and get a transaction set. Then, encrypt the transaction set by a digital signature and calculate a unique hash to identify the specific transaction by a hash function. Finally, obtain block data by working out block header elements and encapsulating transactions to form a block body.

Step 2: distribute the hash to all other nodes in the blockchain; the hash contains a math problem.

Step 3: all other nodes maximize their potential computing power to solve the math problem. The node that first obtains the answer owns the right of accounting and earns Bitcoins automatically generated by the blockchain system. This node is regarded as the accounting node indicated by v_0.

Step 4: the accounting node v_0 distributes the block data to all other nodes in the blockchain.

Step 5: all other nodes confirm the correctness of the block data and link it to the end of the blockchain.

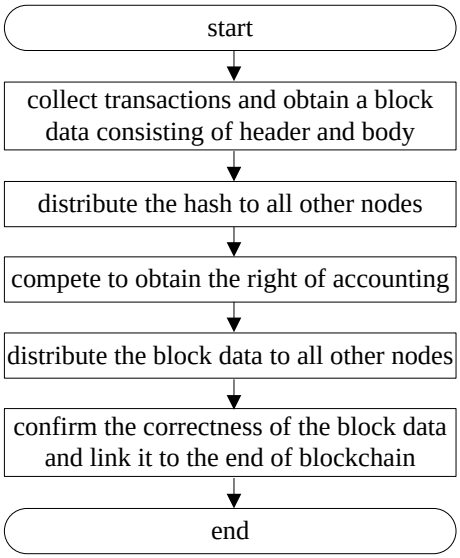

Fig. 1. The transaction process of a blockchain

From the entire process of the transaction, step 1, step 3 and step 5 describe generating transactions, competing to obtain the right of accounting and linking the block data to blockchain. All these steps occur in nodes themselves and do not involve communication between nodes, so if you want to shorten the time consumed in these steps, you must improve node computing and processing ability by upgrading their hardware configurations. However, step 2 and step 4 describe distributing data to all other nodes in the blockchain and involve the blockchain-based communication between nodes, so these two steps are the most time-consuming steps. In other words, the efficiency of distribution determines the time of transaction validation and confirmation. Therefore, to distribute block data to all other nodes as quickly as possible, establishing a model and constructing an algorithm for blockchain-based communication is worthy of study.

Fig. 2 shows the architecture of a blockchain, which is composed of a data presentation layer, data transport layer, validation layer, incentive layer, contract layer and application layer, from the bottom to the top. The data transport layer defines the network structure, communication mechanism, communication impact factor, communication algorithm, and performance parameters. Our research in this paper belongs to the data transport layer. A blockchain is a decentralized P2P network that has the advantages of distribution, extendibility, robustness, and load balancing. P2P network is suitable for the trading, validating and accounting of a blockchain, and it allows transaction information to be easily distributed and directly shared. When a transaction is generated by one node, quickly distributing the transaction to all other nodes requires a communication model and communication algorithm.

Paper [14] researched the hierarchical 2PC protocol algorithm, the Kruskal communication tree algorithm and the branch first communication algorithm. They belong to the single-link communication model, which does not take into account the nodes' communication capability. Papers [15,16] proposed a concurrent communication tree algorithm and an FIN multicast optimization algorithm based on the nodes' communication capability; they belong to the

multi-link communication model, and these algorithms construct communication trees with the goal of shortening communication time, but they neglect the nodes' reliability. If we do not consider the nodes' reliability, the possibility of link failure may be great. Various models have been proposed to improve reliability. Paper [17] presented an iterative and dynamic trust computation model named IDTrust, which has quick response and sensitivity to attacks. Paper [18] proposed a dynamic grouping-based trust model, DGTM, which always achieves the highest successful transaction rate under different circumstances. Papers [19, 20] established a novel dynamic trust model for P2P networks and set up a community-based trust mechanism by combining local trust and global trust strategies to enhance the security. These models solve the trust problem, but they cannot guarantee communication efficiency.

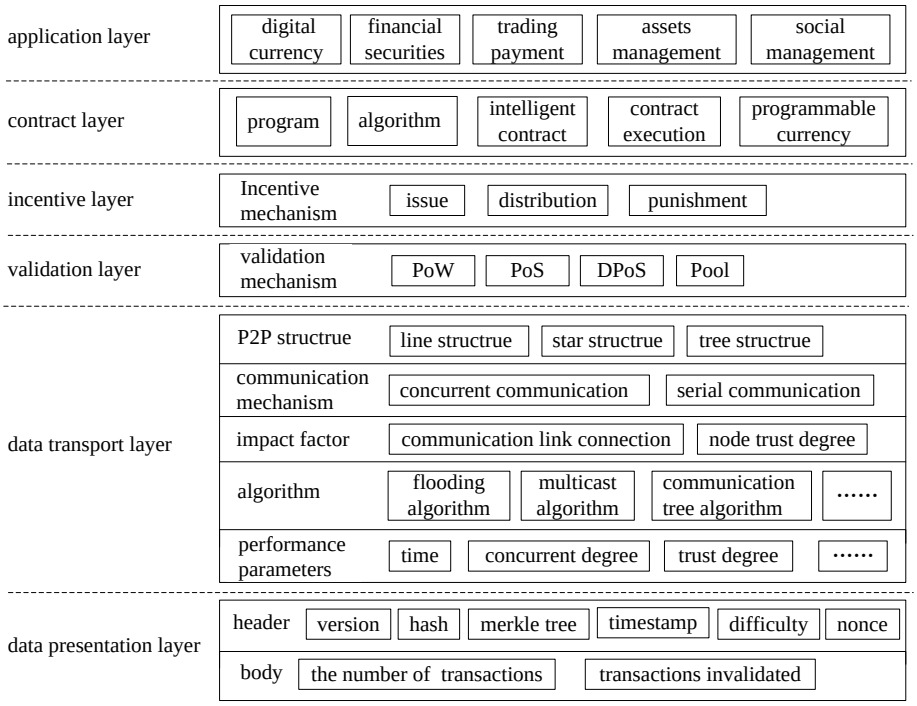

Fig. 2. The architecture of a blockchain

Paper [21] proposed a new multicast tree update strategy based on perturbation theory that reduces the data transmission interruption and improves the user experience. Paper [22] presented a novel multicast tree construction and maintenance approach based on topological distances. These multicast algorithms belong to local routing strategies. However, to finish a particular validation of a transaction in blockchain, all nodes in blockchain participate in validating the transaction, and the entire multicast tree needs to be reconstructed from a global perspective. Thus, local routing strategies are obviously not applicable for blockchain-based communication.

A large number of sensor nodes rush into a blockchain and scatter in a geometric region, with nearby nodes communicating with each other directly for transaction validation. Without the help of a large number of uniformly deployed seed nodes, the failure in WSNs can happen with possible holes [23]. There are different strategies proposed for WSNs that rely on the heat diffusion equation and traffic model to finish information querying and navigation [24].

In summation, the models or algorithms mentioned above have some limitations and do not fully satisfy the efficiency and reliability requirements for blockchain-based communication because they only consider a single communication factor (node communication capability or node trust degree) and optimize a single communication performance parameter (communication time or communication reliability); thus, there are also some differences from practical blockchain-based communication. The main objective of this paper is to integrate communication factors and improve both the efficiency and the reliability of blockchain communication. This paper first establishes the multi-link concurrent communication model and then proposes the communication algorithm based on the model; finally, the communication performance is analyzed and evaluated via simulation results.

3. Multi-link Concurrent Communication Model Based on Trust Degree

3.1 Nodes and Their Classifications

A blockchain is a distributed network system composed of nodes, which are called miners in the Bitcoin system. Nodes can be regarded as mobile terminals, PCs, servers or a miner pool made up of any number of nodes. To be one node, it is required that the node have computing power, regardless of its physical hardware.

According to the role that nodes play in a blockchain transaction, they can be classified as accounting nodes, which distribute the block data to other nodes, and validating nodes, which verify the correctness of the block data after they receive it.

Based on whether they forward data or not, nodes fall into one of three types: source nodes, forwarding nodes and leaf nodes. In a blockchain transaction, the accounting node owns the block data to be transmitted to other nodes, so the accounting node is the source node, indicated by v_0 for convenience purposes.

A node has the functions of requesting, forwarding and saving data. Nodes can be divided into full-data nodes and lightweight nodes according to the cover of block data.

3.2 Communication Link Number for Nodes

With the popularity of smartphones, a large number of mobile terminals can access the Internet anywhere and anytime via wireless networks. Mobile payments in Internet transactions have increased year by year at an alarming speed. The transmitting ability of node highly affects the network performance [25]. Both fixed and mobile terminals are regarded as nodes in a blockchain network, and they have obvious differences in forwarding capacity, so the communication link number for nodes continues to represent the communication capacity of nodes in blockchain-based communication [26].

3.3 Trust Degree for Nodes

The trust relationship is the basic relationship in transactions, and a good trust relationship is helpful to complete transactions. Most transactions in a blockchain are in the form of trust. Thus, besides the communication impact factor, such as the communication link number for the nodes, the trust relationship between nodes is also considered. This paper introduces the concept of trust degree for nodes. The research on trust degree includes definition, representation and measurement [27].

Trust degree for nodes is an evaluation index measuring the nodes' reliability with regard to providing resources and services. Trust degree for nodes consists of two aspects: (1) the credibility of data held by a node, which means that the forwarding data residing in a node

could be interpolated or disguised as a virus, which would lead to enormous harm and damage to the current blockchain network information environment; and (2) the stability of the node with regard to providing data for other nodes, which means that the node could leave randomly or fail suddenly when it forwards data to other nodes [28-30].

According to the performance of nodes in blockchain transactions, nodes can be divided into honest nodes, free-rider nodes, and malicious nodes.

(1) Honest nodes. When providing service to other nodes, nodes with a higher behavioral stability and reputation value, which can ensure the correctness of data forwarded to other nodes, are known as honest nodes. It is expected that nodes with good reputations are located in the top or upper part of the communication tree, which ensures that they can provide more reliable and more stable service for the lower nodes of the communication tree. We demand that only honest nodes be responsible for forwarding data.

(2) Free-rider nodes. A blockchain is a P2P network, which advocates the idea of free sharing and partnerships of peers. In practice, nodes are selfish to a certain extent and pursue their own self-interest, so after they achieve resources and services, they are unwilling to bear forwarding tasks. The nodes that get "something for nothing" are known as free-rider nodes. According to the selfishness and the unreliability of free-rider nodes, we try to select free-rider nodes to forward data as little as possible if sufficient honest nodes are available to finish the communication task. We demand that free-rider nodes be located in the lower part of the communication tree or located as leaf nodes.

(3) Malicious nodes. Nodes juggle and throw away data, and even worms or Trojans are disguised as data to be forwarded to others. These behaviors of nodes can severely damage the security and stability of blockchain communication. Nodes that disseminate false data and demolish resources are known as malicious nodes. To purify the blockchain transaction environment and ensure the QoS of the network, malicious nodes with deceptive data are isolated, and other nodes are prohibited from intercommunicating with the malicious nodes because they may cause the spread of false data and pollution of the network environment. Thus, in the communication tree, we demand that malicious nodes be located as leaf nodes.

In research, trust is usually represented quantitatively. Trust degrees for nodes are distributed in [0,1]. Nodes with trust degree 1 are trusted completely. In contrast, nodes with trust degree 0 are completely unbelievable. $t(v_i)$ represents the trust degree value for any node v_i. Nodes with poor trust degrees are only recommended as leaf nodes in the communication tree. Even if there are not enough honest nodes selected to forward data, nodes with poor trust degree are forbidden to bear forwarding tasks because these nodes may cause serious damage to the network. **Table 1** lists node classifications and trust degrees.

Table 1. Node classifications and trust degrees

Node classification	Trust degree interval	Role in communication	Position in communication tree
Honest nodes	$0.8 \le t(v_i) \le 1$	Forward data	Located in the top or upper part of the communication tree
Free-rider nodes	$0.6 \le t(v_i) < 0.8$	Encourage forwarding	Located in the lower part of the communication tree or located as leaf nodes
Malicious nodes	$0 \le t(v_i) < 0.6$	Ban forwarding	Located as leaf nodes

3.4 Establishing the Model

The validation and confirmation of block data in a blockchain transaction uses a line or star structure; that is, the accounting node v_0 forwards the block data to other nodes in turn to validate the correctness of the block data. This line or star communication structure depends on a single node. Once it breaks down or fails, the entire network might become paralyzed. This paper employs a tree communication structure, which is suitable for forwarding the block data. For transaction validation, based on a communication tree algorithm, all nodes in the blockchain are self-organized to forward the block data in the tree structure, which is a virtual logical network and does not change the actual physical network topology structure.

How to construct a communication tree depends on the algorithm. In an actual network, all nodes are distributed over a broad geographical area, and they are meshed together physically through wired or wireless media. Each node may provide and receive data to and from other nodes using fiber-optic cables or other types of wired or wireless connections. Whether in a long-haul fiber-optic communication system [31,32], wireless LAN or other combined system, we do not have to think about the physical media and physical connections. The proposed model only considers the logical connections between nodes for finishing an assigned transaction validation.

In a blockchain, the status of each node is equal, and each node is a service requester and service provider. Because of the concurrent communication mechanism introduced, the communication process does not rely on a single source, and nodes that receive data can undertake proper communication tasks. The basic idea of concurrent communication is to assign proper tasks to the other nodes besides the communication source node. Concurrent communication can shorten communication time and greatly improve the communication efficiency. Only considering the concurrent communication, tree communication belongs to a single-link concurrent communication model. After the introduction of the communication link number for nodes, tree communication evolves into a multi-link concurrent communication model. The evolution process is shown in **Fig. 3. Fig. 3(a)** is a single-link concurrent communication model in which any node intercommunicates with only one node at the same time; that is, all node communication link numbers are equal to 1. The numbers on the lines represent the time periods. After 1 time period, 2 nodes have received data; after 2 time periods, 4 nodes have received data; after m time periods, 2^m nodes have received data, etc. Thus, for a communication tree with N nodes, the time to finish the entire communication indicated by $f(t)$ is $\lceil \log_2 N \rceil$, the average end-to-end delay $f_{\mathrm{AED}}(t)$ is less than $\lceil \log_2 N \rceil$, the concurrent degree C_d is $Max\left\{N - 2^{\lceil \log_2 N \rceil - 1}, 2^{\lceil \log_2 N \rceil - 2}\right\}$, and the node utilization rate U_r is between 30% and 50%. When the number of nodes is exactly 2^m, the node utilization rate U_r reaches the largest value, 50%. $\lceil X \rceil$ indicates the smallest integer that is greater than or equal to X.

After the introduction of the communication link number for nodes, one node can communicate with multiple nodes at the same time. Thus, a single-link concurrent communication model, as shown in **Fig. 3(a)**, evolves into a multi-link concurrent communication model, as shown in **Fig. 3(b)**. A node in **Fig. 3(a)** evolves into a cluster that consists of several nodes, and they work together to perform a certain communication task. The number of nodes in the cluster depends on the sum of the communication link numbers of the nodes in their parent cluster. For example, the number of nodes in cluster1 is $l(v_0)$. The lower a cluster is located in the communication tree, the larger the number of nodes in the cluster. To develop the node's communication capacity with a maximum limit, any nodes v_i are fully connected with other nodes that have not received data; that is, the number of other

nodes can reach the maximum value $l(v_i)$. Nodes in the same cluster receive data at the same time. In addition, the block data is forwarded immediately after node v_i receives it, so it is a non-stop communication process.

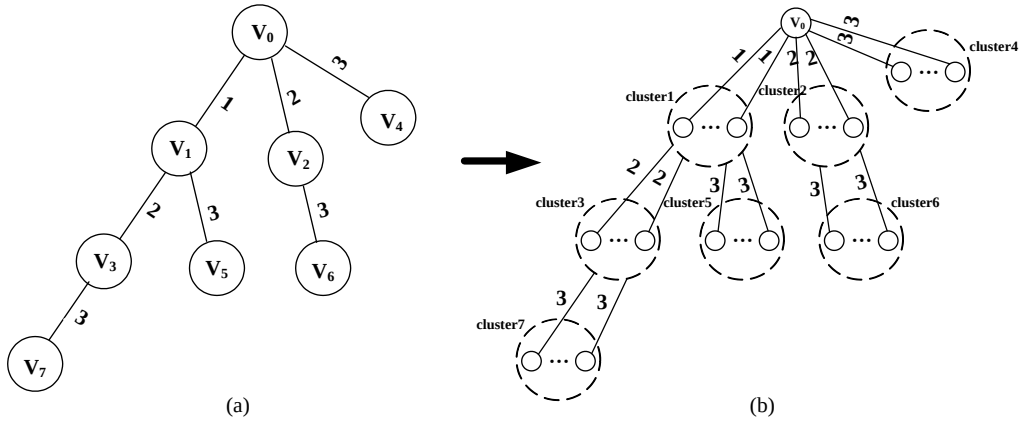

Fig. 3. A single-link concurrent communication model and a multi-link concurrent communication model

3.5 Mechanisms of the Model

Based on mentioned above, we concludes the mechanisms of the proposed model.

Firstly, the model belongs to a multi-link communication. Due to the heterogeneity of nodes, nodes are different in transmitting ability, and each node has multi-connections with other nodes for forwarding at the same time. This multi-link communication can make maximum use of nodes' potential communication capacity. From **Fig. 3**, it is obvious that the efficiency of the multi-link communication is superior to that of the single-link communication.

Secondly, the model is based on the concurrent communication mechanism, so the communication concurrency is high. Concurrent communication mechanism means to assign proper tasks to other nodes besides the communication source node. When nodes have received data, they do not hesitate to forward data to other nodes over the next communication stage. Several transmission tasks are forwarded by different nodes at the same time, so the communication efficiency is high. The details of concurrent communication rules are seen in paper[14].

Thirdly, the model introduces the concept of node trust degree and considers the reliability problem of blockchain-based communication. The proposed algorithm based on the model is to select nodes with high trust degree as much as possible to undertake transmitting.

Finally, the model adopts tree topology. Using a tree structure for blockchain-based communication has the following advantages: (1) according to the hierarchy of the communication tree, nodes that have data can participate in forwarding at the next stage, and the overall communication does not rely on a single source node; (2) if a malicious node is selected to forward data carelessly, this will only cause unreliable data in its cascaded subtrees and will not pollute other subtrees, so this structure has good pollution isolation; and (3) the failure of individual nodes will not cause the bottleneck effect and will not lead to entire-network paralysis.

4. Integrated Factor Communication Tree Algorithm

4.1 Agreements and Definitions for the Communication Tree

A block data unit of a blockchain consists of a header and a body. The header contains the version number, hash value of the parent block, Merkle root, timestamp, difficulty value, nonce, etc. The body includes the transaction counter and transaction records. The transaction validation is to assign the block data to other nodes to verify the correctness of the transaction collectively. According to the transaction validation, the agreements are as follows:

(1) The block data after validation is distributed to all nodes in the blockchain by the accounting node v_0, so all nodes receive the same block data. Therefore, it is agreed that the communication task between any two nodes is the same.

(2) Because of the transactions generated and validated frequently in the blockchain, each transaction needs to be validated by other nodes in the blockchain, so each node knows the routing information between nodes, the location information of the nodes, and the reliability and communication capacity for nodes. A blockchain network has fine perceptible characteristics for the network topology.

(3) For a blockchain with N nodes, the number of accounting nodes, which obtain the accounting right by competition of computing power, is only 1. The transactions are distributed to other nodes to validate, so the number of validating nodes is $N-1$.

We use the definitions of concurrent communication time $f(t)$, average end-to-end delay $f_{\mathrm{AED}}(t)$, node utilization rate U_r, and concurrent degree C_d from papers [16, 26, 33]. For research purposes, the definitions are added as follows.

Definition 1: For finishing of the entire communication task, the number of forwarding events by node v_i, which is a source node, is called the communication task of v_i. It is indicated by $Task(v_i)$.

For a communication tree with N nodes, to make all nodes have data, the data must be forwarded N-1 times. Then

$$N-1=\sum_{i=1}^{N} Task(v_i) \tag{1}$$

If $Task(v_i)$ is 0, the node v_i only receives data and does not forward data. Thus, the node v_i is located in the communication tree as the leaf node.

Definition 2: The reliability of data provided by v_i and the stability of forwarding by v_i are called the trust degree of node v_i. It is indicated by $t(v_i)$, $t(v_i)\in[0,1]$.

Definition 3: For a communication tree with N nodes, the trust degree for the communication tree is indicated by T_d. Then

$$T_d=\frac{\sum_{i=1}^{N} Task(v_i)t(v_i)}{N-1} \tag{2}$$

Thus, the trust degree for the communication tree is determined by integrated consideration of the number of forwarding events by v_i and its trust degree. According to definition 1 and definition 2, we know that $T_d\in[0,1]$.

Definition 4: The communication route from source node v_0 to leaf node v_i is a sequence of edges $(e_0,\cdots,e_j,\cdots,e_i)$, which is called the communication route of v_i. This is indicated by $Route(v_i)$. Then, $Route(v_i)=(e_0,\cdots,e_j,\cdots,e_i)$.

The number of forwarding events on the communication route $Route(v_i)$ is called the link stress of $Route(v_i)$. This is indicated by $Num\{Route(v_i)\}$, where $Num(X)$ is the number of the set X. Thus, the link stress of $Route(v_i)$ is the number of edges $(e_1,\cdots,e_j,\cdots,e_i)$.

Definition 5: The average link stress of the communication tree is indicated by A_{ls}:

$$A_{ls}=\frac{\sum_{i=1}^{R_n} Num(Route(v_i))}{R_n} \tag{3}$$

where R_n is the number of routes needed to finish the overall set of communication tasks. According to the definition of the concurrent degree C_d, R_n is equal to C_d.

4.2 Algorithm Description

The link first communication tree algorithm [26] (or LFT, for short) can maximize the potential communication capacity of nodes such that the concurrent communication time is minimal and optimal, but this algorithm does not consider the trust degree for nodes. If, in constructing the communication tree, the trust degrees for nodes are only considered and the nodes are added to the communication tree in strict accordance with the sizes of the nodes' trust degrees, this algorithm is called the trust first communication tree algorithm (or TFT for short). The aim of the TFT algorithm is to ensure communication reliability, security and stability, and the TFT algorithm improves the trust degree for the communication tree by selecting the node that has better a trust degree value for service. However, there is one problem with TFT: a node can have a better trust value yet still unfortunately be poor in communication capacity. In such a case, the node's communication link number is smaller, even 0, and thus the node cannot provide the services for other nodes. Thus, the TFT algorithm can enhance communication stability at the expense of the concurrent communication time; that is, the TFT algorithm may result in the efficiency of blockchain-based communication being low.

To ensure that the communication is more stable and improve the efficiency of blockchain-based communication, the integrated factor communication tree algorithm (or IFT for short) is proposed, in which the communication capability and trust degree value for each node are considered together as a whole. The IFT algorithm not only guarantees communication stability and reliability but also ensures that the communication efficiency is close to, or even equal to, the optimal solution that the LFT algorithm can obtain. Constructing a communication tree based on the IFT algorithm, we first need to determine the node priority sequence and then join the nodes according to the node priority sequence.

The fundamental principle of determining the node priority sequence is to choose the node that has a high trust degree and powerful communication capability. In many cases, it is disappointing that high trust and powerful communication capability are not uniform for the same node. If one node has a powerful capability of communication yet is poor in trust, we give up on it and prohibit it from forwarding data to other nodes because it is not worth improving the communication efficiency if the communication reliability is not ensured. The first step of determining the node priority sequence is to sort the nodes by their communication link number, from largest to smallest. If they have the same communication link number, they are sorted by the trust degree value, from largest to smallest. We now have a node priority

sequence. Next, we adjust the node priority sequence. If the node's trust degree value does not surpass a given trust threshold, it is adjusted to the end of the node priority sequence. According to **Table 1**, we know that honest nodes are required to forward and free-rider nodes are encouraged to forward, so the trust threshold is equal to or greater than 0.6.

In this paper, nodes in blockchain are abstracted logically into a tree structure to organize communication, and we do not care about the physical connection structure. According to the actual situation of communication, the blockchain network is abstracted as a graph $G=(V,E,L,T)$ with node set V, edge set E, communication link number set L and trust degree set T. $L=\{l(v_0),l(v_1),\cdots,l(v_i),\cdots,l(v_n)\}$, $T=\{t(v_0),t(v_1),\cdots,t(v_i),\cdots,t(v_n)\}$, where $l(v_i)$ represents the maximum communication link number for v_i and $t(v_i)$ represents the trust degree value for v_i. For transaction validation in blockchain, only one node generates the block data and distributes it to other nodes. Thus, for convenience purposes, node v_0 is defined as the source node to initiate a communication task. Set V_T and E_T represent node set and edge set for the communication tree. The complement sets of V_T, L_T, and T_T are $\overline{V_T}$, $\overline{L_T}$, and $\overline{T_T}$, respectively. That is, $\overline{V_T}\cup V_T=V$, $\overline{L_T}\cup L_T=L$, and $\overline{T_T}\cup T_T=T$. The trust threshold is 0.6. $f(t)$ is the concurrent communication time taken to finish a communication task. The IFT communication tree algorithm is described below.

Step 1: The source node is v_0. Sort by the nodes' communication link numbers from largest to smallest and obtain an ordered set $L=\{l(v_1),\cdots,l(v_i),\cdots,l(v_j),\cdots,l(v_n)\}$, $(1\le i\le n,1\le j\le n)$. Any two nodes v_i and v_j in L should satisfy two conditions: (1) if $l(v_i)\ne l(v_j)$, then $l(v_i)>l(v_j)$; and (2) if $l(v_i)=l(v_j)$, then $t(v_i)>t(v_j)$;
$k=1$, $S_{Tmp}=\overline{S_{Tmp}}=\emptyset$.

Step 2: Traverse each item $l(v_i)$ in set L. If $l(v_i)\ge 0.6$, $S_{Tmp}\Leftarrow S_{Tmp}\cup\{v_i\}$; otherwise, $\overline{S_{Tmp}}\Leftarrow\overline{S_{Tmp}}\cup\{v_i\}$; $k++$.

Step 3: If $k<N$, go to Step 2; otherwise, $S=S_{Tmp}\cup\overline{S_{Tmp}}=\{v_1,\cdots,v_i,\cdots,v_n\}$.

Step 4: The source node is v_0, and the node priority sequence is $S=\{v_1,\cdots,v_i,\cdots,v_n\}$ $(1\le i\le n)$
Let $V_T=\{v_0\mid v_0\in V\}$, $E_T=\emptyset$, $L_T=\{l(v_0)\mid l(v_0)\in L\}$, $T_T=\{t(v_0)\mid t(v_0)\in T\}$.
Thus, $\overline{V_T}=S$, $\overline{L_T}=\{l(v_1),\cdots,l(v_i),\cdots,l(v_n)\}$, $\overline{T_T}=\{t(v_1),\cdots,t(v_i),\cdots,t(v_n)\}$.
Let $V_{Tmp}\Leftarrow V_T$, $f(t)=0$.

Step 5: Traverse each node v_i in V_{Tmp}, $\forall v_i\in V_{Tmp}$, select m nodes $\{v_1^i,\cdots,v_k^i,\cdots,v_m^i\}$ from the ordered set $\overline{V_T}$ in turn, where $m=Min\{Num(\overline{V_T}),l(v_i)\}$, $v_k^i\in\overline{V_T}$, v_i transmits data to v_k^i, so v_i is the parent node of v_k^i.
Let $V_T\Leftarrow V_T\cup\{v_1^i,\cdots,v_k^i,\cdots,v_m^i\}$,
$E_T\Leftarrow E_T\cup\{(v_i,v_1^i),\cdots,(v_i,v_k^i),\cdots,(v_i,v_m^i)\}$,
$L_T\Leftarrow L_T\cup\{l(v_1^i),\cdots,l(v_k^i),\cdots,l(v_m^i)\}$,
$T_T\Leftarrow T_T\cup\{t(v_1^i),\cdots,t(v_k^i),\cdots,t(v_m^i)\}$,
$f(t)=f(t)+1$.

Step 6: if $Num(V_T)<N$, $V_{Tmp}\Leftarrow V_T$, go to Step 5; otherwise, the algorithm ends.

5. Simulation and Analysis

5.1 Theorems for the Communication Tree

Theorem 1: For a communication tree with N nodes, no matter which algorithm you use to construct the communication tree, the communication task of node v_i indicated by $Task(v_i)$ satisfies $0 \le Task(v_i) \le l(v_i)(f(t) - f_{v_i}(t))$.

Proof: If node v_i wants to forward data to other nodes, first it must receive data. According to definition 1 in paper [26], node v_i receives data after $f_{v_i}(t)$ time periods, so after $f(t) - f_{v_i}(t)$ time periods, node v_i forwards data to other nodes, the number of which can reach the maximum value $l(v_i)$, and the number of forwarding events by the node v_i is $Task(v_i) = l(v_i)[f(t) - f_{v_i}(t)]$. However, in the $f(t)$ th time period (the last period), if the number of nodes that have not received data is less than $l(v_i)$, the node v_i forwards data to the remaining nodes, and the communication link number for node v_i cannot be the maximum value $l(v_i)$. Therefore, $Task(v_i) \le l(v_i)[f(t) - f_{v_i}(t)]$. In addition, if $l(v_i) = 0$, that is to say, it does not have the ability to forward data, $Task(v_i) = 0$. In conclusion, $0 \le Task(v_i) \le l(v_i)(f(t) - f_{v_i}(t))$.

According to Theorem 1, $Task(v_i)$ is not only about $l(v_i)$, but it is about the elapsed time that node v_i has received data. The sooner node v_i receives data, the higher the level of the tree node v_i is located in (the closer to the source node v_0), and node v_i may forward more communication tasks. Thus, the source node v_0 can undertake most of the communication workload, and the leaf nodes in the communication tree cannot undertake any communication tasks.

Theorem 2: For a communication tree with N nodes, no matter which communication algorithm you use to construct the communication tree, the trust degree for the communication tree indicated by T_d satisfies $Min\{t(v_i)\} \le T_d \le Max\{t(v_i)\}$, where $Min\{t(v_i)\}$ and $Max\{t(v_i)\}$ are the minimum and maximum values of the nodes' trust degree, respectively.

Proof: According to definition 3, T_d satisfies

$$\frac{Min\{t(v_i)\}\sum_{i=1}^{N} Task(v_i)}{N-1} \le T_d \le \frac{Max\{t(v_i)\}\sum_{i=1}^{N} Task(v_i)}{N-1}$$

In addition, according to definition 1, $N - 1 = \sum_{i=1}^{N} Task(v_i)$. Therefore, $Min\{t(v_i)\} \le T_d \le Max\{t(v_i)\}$.

According to Theorem 2, when all nodes' trust degrees $t(v_i) \equiv m$, T_d also remains at m.

Theorem 3: The trust degree T_d for a communication tree constructed based on the TFT algorithm is optimal.

Proof: Assume that there is another optimal algorithm that constructs a communication tree called T_1. The TFT algorithm constructs a communication tree called T_2. If $T_1 = T_2$, it is not necessary to prove. Now suppose that $T_1 \ne T_2$. k ($k > 0$) is the number of nodes located in different positions between T_1 and T_2. As follows, T_2 will be gradually replaced to become T_1

so that we can prove that they have the same trust degree. Assume that the first $m-1$ steps of adding nodes to the communication tree are the same. Thus, the m th step is different, which is the first different step. That is, find the first different node v_i in T_1 and v_j in T_2, $v_i \neq v_j$. The TFT algorithm does not select node v_i because $t(v_i) \leq t(v_j)$. Now consider a new communication tree called T_3. The node set of T_3 is $V(T_3) = V(T_1) - v_i + v_j$, and we know that $t(v_i) \leq t(v_j)$, so $T_d(T_3) \geq T_d(T_1)$. After k different nodes are replaced, we obtain the communication tree T_2, so $T_d(T_2) \geq T_d(T_1)$. Therefore, the trust degree T_d of the TFT algorithm is optimal.

Theorem 4: For a communication tree with N nodes, the communication link number of any node v_i keeps at m, which means $l(v_i) \equiv m$. the concurrent degree satisfies $C_d = \max\left\{ N-(m+1)^{\lceil \log_{(m+1)} N \rceil -1}, (m+1)^{\lceil \log_{(m+1)} N \rceil -2} \right\}$.

Proof: For a communication tree with N nodes, we know $f(t) = \lceil \log_{(m+1)} N \rceil$ when $l(v_i) \equiv m$ from paper[26]. According to the concurrent communication mechanism, nodes in the communication tree increase with time to the $(m+1)$ th power. In the ($\lceil \log_{(m+1)} N \rceil$-1)th time periods, $(m+1)^{\lceil \log_{(m+1)} N \rceil -2}$ nodes transmit the data to $(m+1)^{\lceil \log_{(m+1)} N \rceil -2}$ other nodes. In the $\lceil \log_{(m+1)} N \rceil$th time periods, $N-(m+1)^{\lceil \log_{(m+1)} N \rceil -1}$ nodes transmit the data to $N-(m+1)^{\lceil \log_{(m+1)} N \rceil -1}$ other nodes. According to the definition of the concurrent degree, it can be gained by $C_d = \max\left\{ N-(m+1)^{\lceil \log_{(m+1)} N \rceil -1}, (m+1)^{\lceil \log_{(m+1)} N \rceil -2} \right\}$.

5.2 Performance Evaluation and Discussions

In the following, three algorithms are compared and analyzed for various communication performance parameters, such as the concurrent communication time $f(t)$, the average end-to-end delay $f_{\mathrm{AED}}(t)$, the trust degree for the communication tree T_d, the node utilization rate U_r, the concurrent degree C_d and the average link stress A_{ls}.

Fig. 4, **Fig. 5**, **Fig. 6**, **Fig. 7**, **Fig. 8**, and **Fig. 9** show the simulation results of performance parameters for the LFT algorithm, TFT algorithm and IFT algorithm. The X-axis signifies the node number, with values of 5, 10, 30, 50, 80, 100, 150, 200, 250, 300, 500 and 1000. The Y-axis in **Fig. 4** signifies the concurrent communication time $f(t)$. The Y-axis in **Fig. 5** signifies the average end-to-end delay $f_{\mathrm{AED}}(t)$. The Y-axis in **Fig. 6** signifies the trust degree for the communication tree T_d. The Y-axis in **Fig. 7** signifies the node utilization rate U_r. The Y-axis in **Fig. 8** signifies the concurrent degree C_d. The Y-axis in **Fig. 9** signifies the average link stress A_{ls}.

In actual communication, fewer nodes have powerful capacity, and most of the nodes are poor in communication or have no ability to forward. Therefore, nodes whose communication link number value is 3, 2, 1, and 0 are randomly selected in proportions of 5%, 10%, 45% and 40%, respectively. In addition, honest nodes, free-rider nodes, and malicious nodes are randomly selected in proportions of 40%, 30% and 30%, respectively.

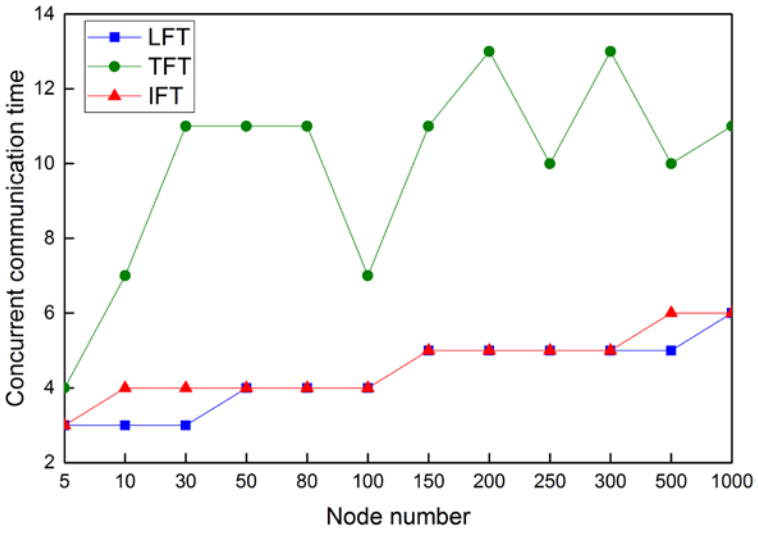

Fig. 4. Concurrent communication time $f(t)$

The concurrent communication time, indicated by $f(t)$, spent finishing the entire communication is the most important parameter to measure communication efficiency. In **Fig. 4**, the concurrent communication time based on the LTF and IFT algorithm does not increase significantly along with the increase of nodes N. It is relatively flat and has a small increase. The reason is the multi-link concurrent communication model, based on which the powerful nodes are selected to be added to the communication tree first and more nodes participate in forwarding data at later communication, so the concurrent communication degree is high and the concurrent communication time has a sudden and dramatic decline. In addition, according to theorem 4 in paper [26], the concurrent communication time of a communication tree constructed based on LFT can reach the optimal value. From **Fig. 4**, it is clearly observed that the concurrent communication time of the IFT algorithm is equal to or slightly higher than that of the LTF algorithm. However, $f(t)$ of the TFT algorithm is longer than that of the IFT algorithm and does not increase with the increase of nodes N. The reason is that the most reliable nodes are selected to add to the communication tree first, but they have no communication capacity to undertake communication, so the concurrent communication time is long and unstable with the increase of nodes N.

The average end-to-end delay, indicated by $f_{\mathrm{AED}}(t)$, is an important parameter for evaluating the communication performance, which reflects the average delay that all nodes spend to receive data. The aim of the communication algorithm is to drive the average end-to-end delay down as much as possible. From **Fig. 5**, the average end-to-end delay of the IFT algorithm approaches that of the LFT algorithm, and that of the TFT algorithm is relatively long. In addition, it can be seen from **Fig. 4** and **Fig. 5** that, for a communication tree with the same nodes N, the concurrent communication time is always higher than the average end-to-end delay, that is, $f_{AED}(t) \le f(t)$.

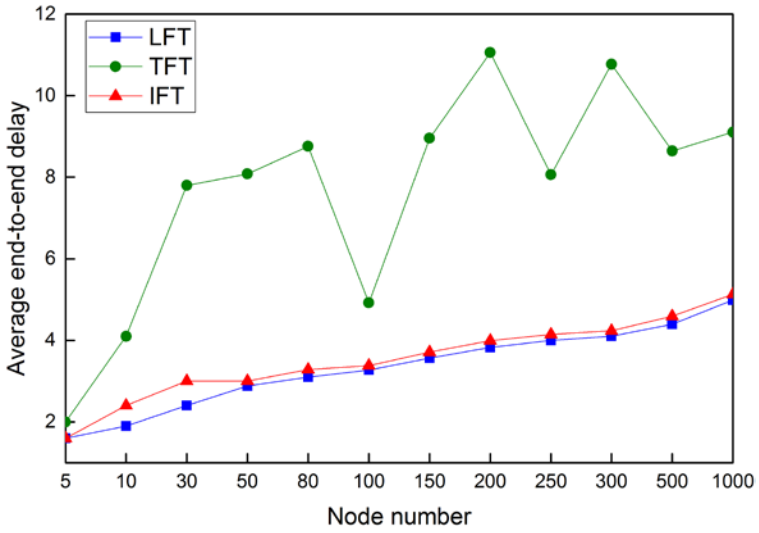

Fig. 5. Average end-to-end delay $f_{\mathrm{AED}}(t)$

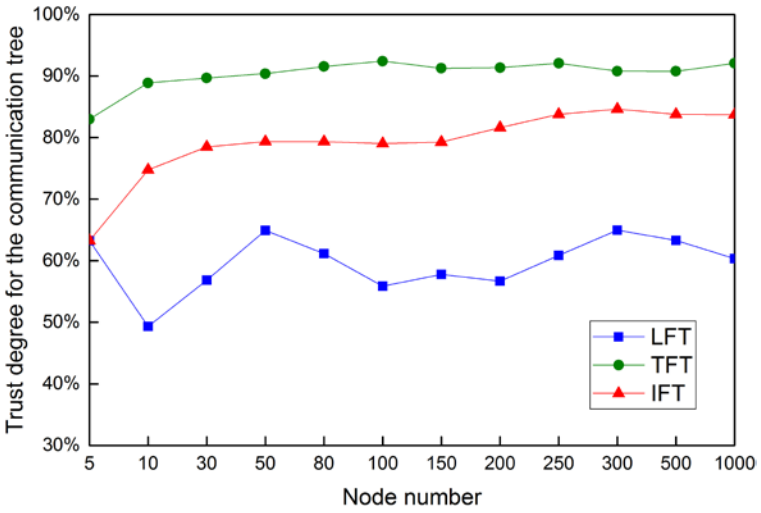

Fig. 6. Trust degree for the communication tree T_d

The trust degree for the communication tree, indicated by T_d, is an important parameter to measure communication reliability and stability. The higher the value of T_d, the higher the communication reliability and the better the communication stability. In **Fig. 6**, the trust degree for a communication tree constructed based on the TFT algorithm is the highest, remaining at approximately 90%. The reason is that, in the TFT algorithm, the high-trust nodes are selected for addition to the communication tree first, and they are thus closer to the root node in the communication tree. By definition, it is known that the trust degree for the communication tree depends on the trust degree of the forwarding nodes. Therefore, T_d of the TFT algorithm is optimal. T_d of the IFT algorithm is at approximately 80%, and the IFT algorithm can obtain better reliability and stability, so this algorithm considers both the communication link number and trust degree for nodes. The LFT algorithm only considers the communication link number for nodes and does not consider the trust degree for nodes when

constructing the communication tree. The only condition for adding nodes to the communication tree is that they have powerful communication capacity. Due to this, powerful nodes may be poor in trust degree. Thus, T_d of the LFT algorithm is only approximately 55% and the worst among the three algorithms.

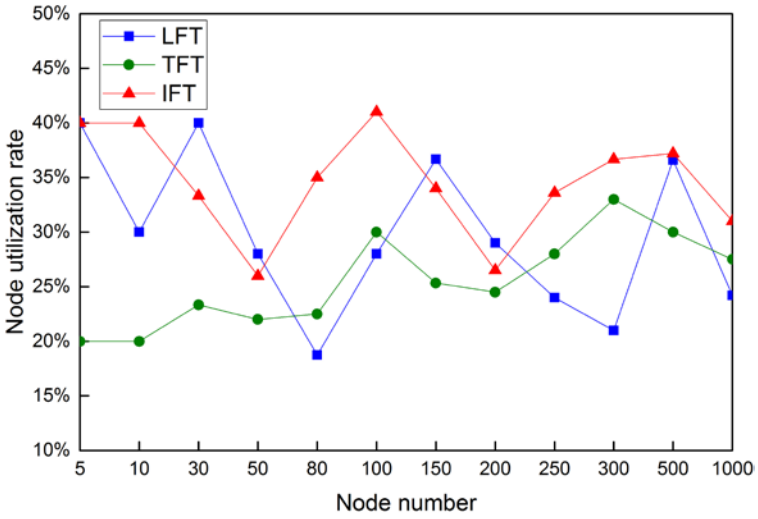

Fig. 7. Node utilization rate U_r

The node utilization rate, indicated by U_r, is defined as the ratio of the number of forwarding nodes to the number of nodes N. It is expected that the node utilization rate U_r is as small as possible; that is, fewer nodes are used to undertake the communication tasks. In **Fig. 7**, the node utilization rate U_r is 30%, which means that, in the communication tree, only approximately 30 percent of nodes undertake communication tasks, and the remaining 70 percent of nodes are located as leaf nodes in the communication tree and do not need to forward data. According to definition 1 in this paper and definition 3 in paper [26], the number of leaf nodes in the communication tree is $Num(\{v_i \mid task(v_i) = 0\}) = N(1-Ur)$.

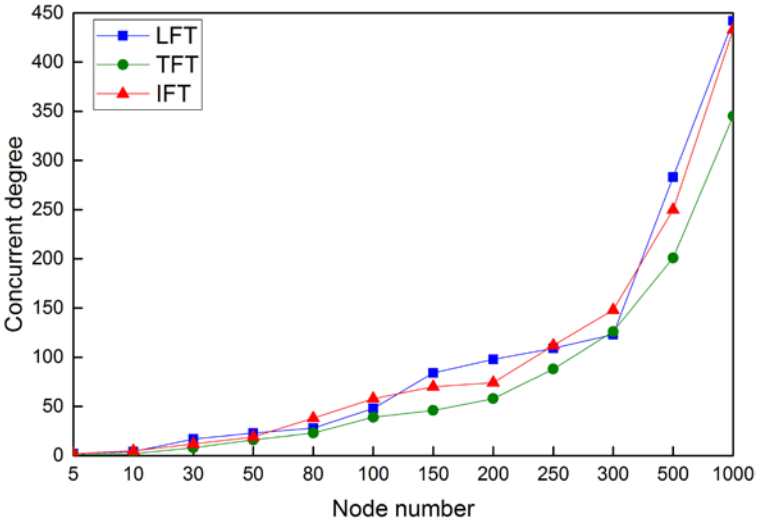

Fig. 8. Concurrent degree C_d

The concurrent degree, indicated by C_d, means the greatest number of communication tasks forwarded at the same time in a communication tree. In **Fig. 8**, the concurrent degree of the three algorithms increases sharply along with the increase of nodes N. These simulation results verify that concurrent communication is significantly better than serial communication in terms of efficiency.

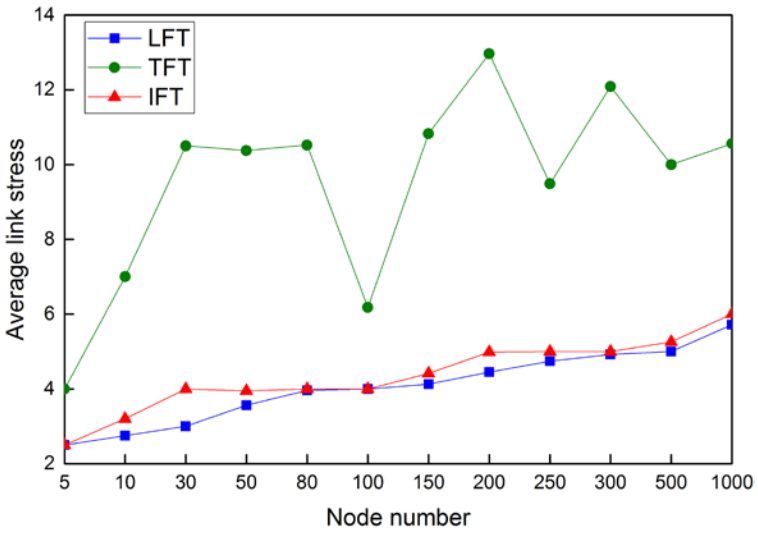

Fig. 9. Average link stress A_{ls}

The average link stress, indicated by A_{ls}, is an important parameter for measuring the average route load. It is expected that a greater number of routes leads to less average link stress. In **Fig. 9**, the average link stress of the LTF and IFT algorithms increases smoothly along with the increase of nodes N, yet that of the TFT algorithm varies in an irregular way and is greater than that of the LTF and IFT algorithms.

5.3 Algorithm Comparison for Communication Performance Parameters

Table 2 provides a comprehensive comparison of communication performance parameters for the three algorithms. From **Table 2**, it is clearly observed that the IFT algorithm obtains better communication performance than the LFT algorithm and TFT algorithm with respect to the concurrent communication time, the average end-to-end delay, the trust degree for the communication tree, the node utilization rate, the concurrent degree and the average link stress. The three algorithms have their own suitable situations. If you relatively care about communication efficiency and are striving to shorten the concurrent communication time, the LFT algorithm is chosen. If you are more concerned with communication reliability and stability, the TFT algorithm is chosen. These two types of algorithms only consider a single communication impact factor when constructing the communication tree and only pursue a single performance parameter optimization at the expense of the other performance parameters. However, the IFT algorithm balances both the communication link number and trust degree, so on the premise of guaranteed communication service reliability and communication stability, the communication efficiency of the IFT algorithm is close to or at the optimal value.

Table 2. A comprehensive comparison of communication performance parameters for the three algorithms

Algorithm	Construct principle	Communication performance parameters						Suitable situation for application
		$f(t)$	$f_{AED}(t)$	T_d	U_r	C_d	A_{ls}	
LFT	Only consider communication capacity for nodes	Best	Best	Poor	Good	Best	Best	Pursue efficiency
TFT	Only consider trust degree for nodes	Poor	Poor	Best	Good	Best	Poor	Pursue reliability
IFT	Balance both	Best	Best	Good	Good	Best	Best	Pursue efficiency and reliability

5.4 Relationship between the Ratio of Honest Nodes and the Trust Degree for the Communication Tree

In a blockchain, it is expected that nodes undertake communication tasks voluntarily to embody the concept of friendship and cooperation, i.e., "one for all, all for one," in P2P, and they are honest not to tamper and falsify the forwarding data. However, nodes are actually selfish to some degree and want to request data but do not want to undertake communication tasks. Here is a problem – how many honest nodes in a blockchain network can guarantee forwarding service reliability and communication stability? Below, we analyze the relationship between the ratio of honest nodes and the trust degree for the communication tree in blockchain-based communication.

In a simulation based on the proposed IFT algorithm, the communication link numbers for all nodes are set as 1 to avoid the influence of communication capacity for nodes. Honest nodes are randomly selected in certain proportions of 10%, 20%, 40%, and 60% of N. The remaining nodes (free-rider nodes and malicious nodes) are divided equally and account for 45%, 40%, 30%, and 20% of N, respectively. The relationship between the ratio of honest nodes and the trust degree for the communication tree is shown in **Fig. 10**, in which the X-axis signifies the node number, with values of 5, 10, 30, 50, 80, 100, 150, 200, 250, 300, 500 and 1000, and the Y-axis signifies the trust degree for the communication tree T_d. For the same ratio of honest nodes, the trust degree for communication T_d changes smoothly along with the increase of nodes N. When the ratio of honest nodes is 10%, T_d is between 75% and 80%. When the ratio of honest nodes is 20% and 40%, T_d is at approximately 85% and 90%, respectively. When the ratio of honest nodes rises to 60%, T_d can reach approximately 95%. In **Fig. 10**, T_d also increases along with the increase of the ratio of honest nodes. Even in the case of a low proportion of honest nodes (such as 10%), the entire communication tree can obtain good stability. The reason is that a concurrent communication mechanism is introduced. If we just make sure that nodes located in the top or the upper part of the communication tree have high trust degrees, the trust degree of the entire communication tree is greatly improved.

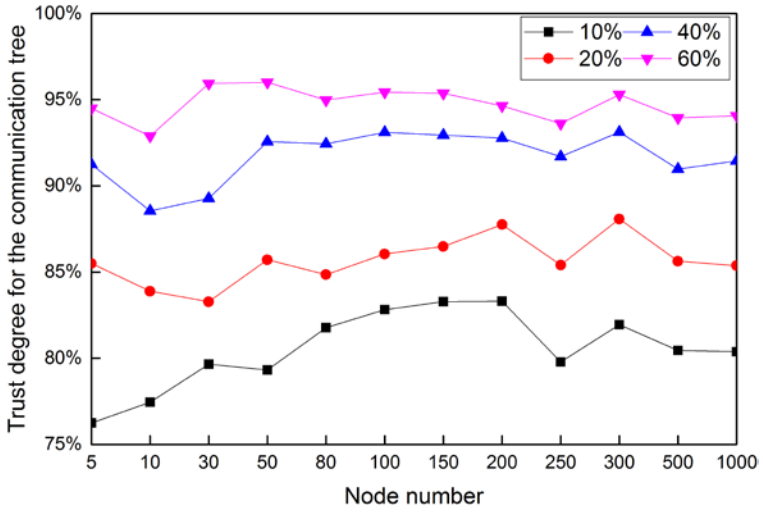

Fig. 10. The relationship between the ratio of honest nodes and the trust degree for the communication tree

6. Conclusion

In a blockchain, quickly distributing the block data is a challenge to the efficiency of blockchain-based communication, and establishing the trust relationship between nodes to finish blockchain transactions is also a challenge to the stability and reliability of blockchain communication. The purpose of this research is to improve the efficiency and reliability of blockchain-based communication. This paper first establishes a multi-link concurrent communication model based on the trust degree and then proposes an integrated factor communication tree algorithm named IFT. In the algorithm, according to the behavioral characteristics of nodes in blockchain-based communication, nodes are classified into three types: honest nodes, free-rider nodes and malicious nodes. This algorithm integrates and balances the communication link number and trust degree, and it first selects nodes with powerful capacity and high trust to add to the communication tree so that they are located in the upper layer of the communication tree and thus can undertake more communication tasks in the ensuing communication. In addition, because of the introduction of a concurrent communication mechanism, the concurrency and communication efficiency are high. The simulation results indicate that, as long as the few nodes with powerful capacity and high trust can be well equipped and maintained in blockchain-based communication, the efficiency and reliability of transaction validation will be improved greatly. The proposed algorithm can reasonably deploy a routing scheme for blockchain-based communication. In the future, we plan to extend the proposed algorithm to solve the problem of blockchain communication by considering more communication impact factors, such as the communication cost between nodes and the priority of node service requests [33].

References

[1] Godsi P, “Bitcoin: bubble or blockchain,” in *Proc. of 9th KES International Conference on Agent and Multi-Agent Systems: Technologies and Applications*, pp. 191-203, May 29, 2015. Article (CrossRef Link).

[2] Kraft D, "Difficulty control for blockchain-based consensus systems," *Peer-to-Peer Networking and Applications*, vol. 9, no. 2, pp. 397-413, March, 2016. Article (CrossRef Link).
[3] Swan M, "Blockchain thinking: the brain as a decentralized autonomous corporation," *IEEE Technology and Society Magazine*, vol. 34, no. 4, pp. 41-52, December, 2015. Article (CrossRef Link).
[4] Eldred M, "Blockchain thinking and euphoric Hubris," *IEEE Technology and Society Magazine*, vol. 35, no. 1, pp. 39, March, 2016. Article (CrossRef Link).
[5] Zyskind G, Nathan O and Pentland A S, "Decentralizing privacy: using blockchain to protect personal data," in *Proc. of 2015 IEEE Security and Privacy Workshops*, pp. 180-184, May 21-22, 2015. Article (CrossRef Link).
[6] Wilson D and Ateniese G, "From pretty good to great: enhancing PGP using Bitcoin and the blockchain," in *Proc. of 9th International Conference on Network and System Security*, pp. 368-375, November 3-5, 2015. Article (CrossRef Link).
[7] Kypriotaki K N, Zamani E D and Giaglis G M, "From Bitcoin to decentralized autonomous corporations: extending the application scope of decentralized peer-to-peer networksand blockchains," in *Proc. of 17th International Conference on Enterprise Information Systems*, pp. 284-290, April 27-30, 2015. Article (CrossRef Link).
[8] George H, "Might the Blockchain Outlive Bitcoin?" *IT Professional*, vol. 18, no. 2, pp. 12-16, March-April, 2016. Article (CrossRef Link).
[9] Nakamoto S, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008 Article (CrossRef Link).
[10] Pai V, Kumar K, Tamilmani K, Sambamurthy V and Mohr A E, "Chainsaw: eliminating trees from overlay multicast," in *Proc. of* 4th *International Workshop on Peer-to-Peer Systems*, pp. 127-140, February 24-25, 2005. Article (CrossRef Link).
[11] Biskupski B, Schiely M, Felber P and Meier R, "Tree-based analysis of mesh overlays for peer-to-peer streaming," in *Proc. of 8th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems*, pp. 126-139, June 4-6, 2008. Article (CrossRef Link).
[12] Mokhtarian K and Jacobsen H A, "Minimum-Delay multicast algorithms for mesh overlays," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 973-986, June, 2015. Article (CrossRef Link).
[13] Kim K, Mehrotra S and Venkatasubramanian N, "Efficient and reliable application layer multicast for flash dissemination," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2571-2582, October, 2014. Article (CrossRef Link).
[14] Liu T S, Li J, Cao Q N. "Study on a network communication optimization algorithm of P2P mode," in *Proc. of Artificial Intelligence and Computational Intelligence*, pp. 212-217, November, 7-8, 2009. Article (CrossRef Link).
[15] Liu T S, Yang K Y, Li J, "Study on a concurrent communication tree algorithm of P2P multi-link mode," in *Proc. of International Conference on Multimedia Technology*, pp. 2034-2038, October 29-31, 2010. Article (CrossRef Link).
[16] Liu T S, Zhang L M, Cheng G J, Li J and Yang K Y, "FIN multicast optimization algorithm in P2P communication," *Application Research of Computers*, vol. 30, no. 5, pp. 1464-1466&1491, May, 2013. Article (CrossRef Link).
[17] Tan Z H, Wang X W and Wang X Y, "A novel iterative and dynamic trust computing model for large scaled P2P networks," *Mobile Information Systems*, no. 4, pp. 1-12, April, 2016. Article (CrossRef Link).
[18] Jia M J, Wang H Q, Ye B and Wang Y, "A dynamic grouping-based trust model for mobile P2P networks", in *Proc. of 2016 IEEE International Conference on Services Computing*, pp. 848-851, June 27, 2016. Article (CrossRef Link).
[19] Liao J and Li Z, "A novel dynamic trust model for P2P network," *Open Automation and Control Systems Journal*, vol. 7, no. 1, pp. 893-901, August, 2015. Article (CrossRef Link).
[20] Wang Q J, Wang J R, Yu J and Yu M, "Trust-aware query routing in P2P social networks," *International Journal of Communication Systems*, vol. 25, no. 10, pp. 1260-1280, October, 2012. Article (CrossRef Link).

[21] Shen Y, Feng J, Ma W J, Jiang L and Yin M., "Overlay multicast update strategy based on perturbation theory," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 1, pp. 171-192, January, 2017. Article (CrossRef Link).
[22] Alekseev S and Schäfer J, "Evaluation of a topological distance algorithm for construction of a P2P multicast hybrid overlay tree," *International Journal of Computer Networks and Communications*, vol. 8, no. 1, pp. 1-12, January, 2016. Article (CrossRef Link).
[23] Wei W, Yang X L, Shen P Y and Zhou B., "Holes detection in anisotropic sensornets: topological methods," *International Journal of Distributed Sensor Networks*, vol. 8, no. 10, October 23, 2012. Article (CrossRef Link).
[24] Wei W and Qi Y., "Information potential fields navigation in wireless Ad-Hoc sensor networks," *Sensors*, vol. 11, no. 5, pp. 4794-4807, May, 2011. Article (CrossRef Link).
[25] Wei W, Xu Q, Wang L, Hei X H, Shen P, Shi W and Shan L., "GI/Geom/1 queue based on communication model for mesh networks," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 3013-3029, November, 2014. Article (CrossRef Link).
[26] Li J and Liu T S, "Study on a P2P communication tree algorithm based on multi-link," *Northwest University Journal: Natural Science*, vol. 40, no. 6, pp. 970-974, November, 2010. Article (CrossRef Link).
[27] Selvaraj C and Anand S, "A survey on security issues of reputation management systems for peer-to-peer networks," *Computer science review*, vol. 6, no. 4, pp. 145-160, July, 2012. Article (CrossRef Link).
[28] Duarte J, Siegel S and Young L, "Trust and credit: the role of appearance in peer-to-peer lending," *Review of Financial Studies*, vol. 25, no. 8, pp. 2455-2484, August, 2012. Article (CrossRef Link).
[29] Huang G M, Hu M, Zhou Y, Liu P S, and Zhang Y C, "A distributed trust model based on reputation management of peers for P2P VoD services," *KSII Transactions on Internet and Information Systems*, vol. 6, no. 9, pp. 2285-2301, September, 2012. Article (CrossRef Link).
[30] Sanchez D, Martinez S and Domingo-Ferrer J, "Co-utile P2P ridesharing via decentralization and reputation management," *Transportation Research Part C: Emerging Technologies*, vol. 73, pp. 147-166, December, 2016. Article (CrossRef Link).
[31] Song H B and Maite B P. "Model-centric nonlinear equalizer for coherent long-haul fiber-optic communication systems," in *Proc. of IEEE Global Telecommunications Conference*, pp. 2394-2399, December 9-13, 2013. Article (CrossRef Link).
[32] Song H B, Maite B P, Xie T J and Wilson Stephen G, "Combined constrained code and LDPC code for long-haul fiber-optic communication systems," in *Proc. of IEEE Global Telecommunications Conference*, pp. 2984-2989, December 3-7, 2012. Article (CrossRef Link).
[33] Li J, Liang G Q, Liu T S and Li X J, "P2P multicast algorithm considering node service priority," *Application Research of Computers*, vol. 34, no. 4, pp. 1176-1179, April, 2017. Article (CrossRef Link).

Jiao Li is a lecturer in the School of Computer Science, Xi'an Shiyou University, China. She is currently pursuing a doctoral degree in the School of Management at Northwestern Polytechnical University, China. Her research interests include blockchain communication, the theory and application of blockchain.

Gongqian Liang received his M.E. and Ph.D. degrees from Northwestern Polytechnical University, China, in 1987 and 2000 respectively. He is a professor in the School of Management at Northwestern Polytechnical University, China. His research interests include Bitcoins system and quality management.

Tianshi Liu received his Ph.D. degree from Northwestern Polytechnical University, China, in 2005, and his M.S. degree from Xi'an Jiaotong University, China, in 1985. He is a professor in the School of Computer Science at Xi'an Shiyou University, China. In 2008, he went to the University of Birmingham as a visiting scholar and studied on P2P communication. His research interests include P2P distributed database and P2P network.