PAPER

MSLT: A Scalable Solution for Blockchain Network Transport Layer Based on Multi-Scale Node Management

Longle CHENG[†,††a)], **Xiaofeng LI**[†,††b)], **Haibo TAN**[†c)], **He ZHAO**[†d)], ***and*** **Bin YU**[††e)], ***Nonmembers***

SUMMARY Blockchain systems rely on peer-to-peer (P2P) overlay networks to propagate transactions and blocks. The node management of P2P networks affects the overall performance and reliability of the system. The traditional structure is based on random connectivity, which is known to be an inefficient operation. Therefore, we propose MSLT, a multiscale blockchain P2P network node management method to improve transaction performance. This approach involves configuring the network to operate at multiple scales, where blockchain nodes are grouped into different ranges at each scale. To minimize redundancy and manage traffic efficiently, neighboring nodes are selected from each range based on a predetermined set of rules. Additionally, a node updating method is implemented to improve the reliability of the network. Compared with existing transmission models in efficiency, utilization, and maximum transaction throughput, the MSLT node management model improves the data transmission performance.

key words: *blockchain, peer-to-peer network, scalability, node management*

1. Introduction

Blockchain is an emerging distributed ledger technology, with unique characteristics of decentralization, tamper-proof design, and traceability. Research on blockchain systems in recent years often aimed at on-chain applications, node storage, data transfer, and consensus efficiency [1]–[3]. The demand for high frequency and high concurrent transactions in real-time payment cannot be met by the current blockchain technology. As a result, how to increase the scalability of decentralized blockchain networks has become a vital question [4], [5].

Peer-to-peer (P2P) networks have intrinsic advantages for blockchain systems over traditional Client/Server structures, such as discreteness, scalability, and fault tolerance [6]. The defining feature of blockchain technology is its dependence on a global P2P network to authenticate and validate all transactions. Therefore, the nodes of the blockchain P2P network require a faster and more efficient transmission mechanism to assure the transmission efficiency of the blockchain network and improve the scalability of the blockchain network. However, there are still some issues in the blockchain P2P network that need to be resolved [7]–[9].

- Low network transmission efficiency. Transactions and blocks should be forwarded multiple times to achieve an agreement, this would increase data transmission time, cause data synchronization delay, and reduce network transmission efficiency.
- Low network utilization. The traditional approach of a P2P network allows for a broadcast with redundancy, which ensures reliable synchronization. However, each node is bound to receive the same data from different nodes many times, which consumes network resources and reduces network utilization.
- Low security and reliability. The blockchain is open to entry by any node. As a result, over the blockchain networks, an attack is possible. For each node to receive the proper data during network transmission, it is essential to guarantee the secure and trustworthy transfer of data in the P2P networks.

In this paper, we propose MSLT, a hierarchical transmission model of the blockchain network based on multi-scale node management, in light of the above analysis of the issues and requirements that need to be resolved immediately in the P2P network for blockchain systems. The main contributions of this paper are as follows:

- A model for multi-scale node management is proposed. The network is set to multiple scales according to different division strengths. One scale corresponds to multiple equalization ranges, and the transmission node chooses one node from each range at each scale to form a list of neighbor nodes.
- A neighbor node update mechanism dependent on transmission speed is adopted. The network transmission speed between nodes is used to update the nearby nodes during each range, i.e., the faster node is utilized to replace the slower node as the neighbor node, ensuring that the data transmission speed between nodes is always the highest.
- A hierarchical transmission mechanism for the blockchain network is proposed. The multi-scale node management mode defines multiple transmission levels, transactions and blocks are transmitted to the neighbors at appropriate scales according to the transmission levels. In addition, sibling nodes are used as a supplement to ensure the dependability of network

Manuscript received April 12, 2023.
Manuscript revised July 25, 2023.
Manuscript publicized September 12, 2023.

[†]The authors are with the Hefei Institute of Physical Science, Chinese Academy of Sciences, Hefei 230031, China.
[††]The authors are with the University of Science and Technology of China, Hefei 230026, China.
a) E-mail: llcheng@hfcas.ac.cn
b) E-mail: xfli@hfcas.ac.cn
c) E-mail: hbtan@hfcas.ac.cn
d) E-mail: zhaoh@hfcas.ac.cn (Corresponding author)
e) E-mail: yub@hfcas.ac.cn (Corresponding author)

DOI: 10.1587/transcom.2023EBP3059

transmission.

The structure of this paper is organized as follows. Section 2 introduces and analyzes related research on a blockchain network. The scheme design of the multi-scale node management model is introduced in Sect. 3. Section 4 constructs a data-transmitting process for blockchain systems adopting the multi-scale node management model. In Sect. 5, the effective transmission rate, maximum throughput, network utilization, and security of data transmission are analyzed and discussed. Section 6 summarizes this paper and outlines future work.

2. Related Work

research aimed at improving the scalability of blockchain can be summarized as off-chain solutions, on-chain solutions, and network transport layer solutions. Wu et al. [6] analyze the P2P protocols of Bitcoin, and Ethereum [14], and discuss the changes in the evolution process of the P2P protocol of blockchain. Delgado et al. [11] characterize P2P cryptocurrency networks by analyzing Bitcoin and conclude that a P2P network presents a new paradigm and a cryptocurrency has to provide reliability and security.

In the Bitcoin blockchain [10], the INV mechanism only broadcasts block hash [12], [13]. The node first sends an INV message to its neighbor, and then only sends complete data to the neighbor that responds to the GETDATA message. This improves the transmission efficiency of complete data but increases the number of network transmissions.

In Ethereum [14], [15], when a validator constructs a new block, the validator sends two different types of messages to its neighbors: NewBlockMsg (containing the entire block data) and NewBlockHashesMsg (containing only the block hash) [16]. Similar to the Bitcoin blockchain, Block hash-only message transfer types in Ethereum also increase the number of network transfers.

Clifford et al. [17] proposed the ultra-thin block transmission technology, the amount of data in the transmission block is only 1/24 of the original, and the deployment of the ultra-thin block does not require forks. Txilm block proposed by 18. Donghui D et al. [18], is a lossy compression block with a salt short hash. According to the short hash, the receiving node obtains transaction data from the local transaction pool and completes the block reorganization. In Vault [19], [20], a cryptocurrency developed by MIT, nodes only need to download a small amount of transaction data to join the blockchain network, reducing bandwidth by 99% compared to Bitcoin and 90% compared to Ethereum.

Other studies improve the P2P network model based on the performance requirements of the blockchain system. Harmony [21] introduces the QUIC [22] protocol (Quick UDP Internet Connections) to realize fast and reliable data transmission. In the EOS blockchain [23], a new block is built and broadcast by one of the 21 block producers, but with high-quality hardware and network. Corallo et al. launched the Fast Internet Bitcoin Relay Engine (FIBRE), aiming to build a more powerful version of the Bitcoin relay network by improving data transmission speed and reducing the number of isolated blocks, or transaction blocks rejected by the network [24]. bloXroute [25] is a Blockchain Distribution Network (BDN), a global content distribution network of high-performance servers for blockchain scalability.

With the need for a P2P network of a blockchain system, experts proposed the blockchain network transmission expansion scheme. The researcher designed new P2P network models based on specific blockchain scenarios. Yang et al. [26] propose PPISM for interactive streaming media. Frahat et al. [27] propose a fully distributed trust management model for IoT P2P networks that provide a large-scale trust model and address the limitations of Blockchain.

In the previous research work on network transmission expansion, our team proposed a Scalable Blockchain P2P network transmission model [28] and a MANDALA Mesh-and-Spoke network [29]. This model groups nodes into different layers and regulates communication rules among groups, which improves the network transmission efficiency. In this paper, the additional transmission layer is adopted to replace the transmission path, which greatly reduces the amount of additional data and further improves the network transmission efficiency and utilization rate.

3. Node Management Model

3.1 Symbol Definition

The symbols involved in this paper mainly focus on MSLT network, the symbols and meanings about the MSLT network are shown in Table 1.

3.2 Kademlia for ETH

In the context of Ethereum (ETH), the Kademlia algorithm serves as a fundamental building block for its P2P network. This algorithm utilizes a distributed hash table (DHT) approach to enable efficient node management. Each node within the Ethereum network is assigned a globally unique identifier known as the “Node ID.” These Node IDs are randomly selected from a 256-bit space, ensuring a broad and diverse representation of nodes.

The Kademlia algorithm in Ethereum provides a robust foundation for the Multi-Scale Node Management Model

Table 1 Symbols of MSLT network.

Symbol	Description
m	Number of bits in the node ID
n	Number of ranges at minimum scale (Scale 1)
k	Size of the scale, ranging from 1 to m/$\log_2$(n)-1
M	Total number of nodes
N	Maximum number of nodes in each range
r_i	Number of node at range i
d_i	Degree of the node i
h	Height of the transmission tree

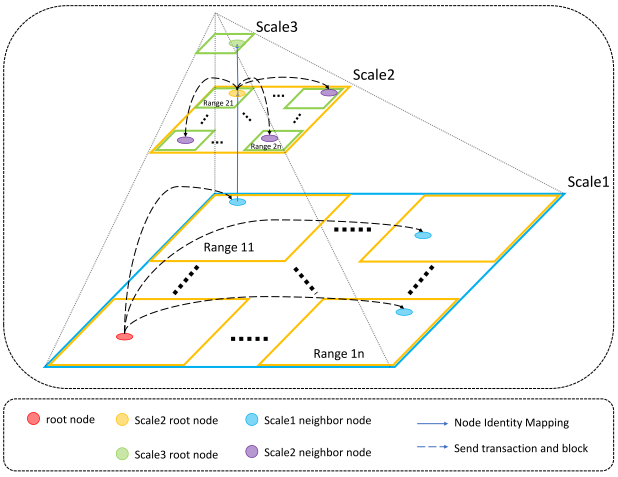


Fig. 1 Structure of multi-scale node management model.

(MSLT) introduced in this paper. This model leverages the Node IDs to facilitate the organization of nodes into multiple scales of different levels. By effectively sorting the nodes based on their Node ID sizes, the network is abstractly divided into various ranges, each corresponding to a specific scale. Such hierarchical structuring enhances the efficiency and security of the P2P network, as depicted in Fig. 1, providing a scalable and adaptable framework for node management.

3.3 MSLT Model Structure

The Multi-Scale Node Management Model (MSLT) proposed in this research presents a hierarchical approach to optimize the P2P network for blockchain systems. Figure 1 illustrates the overall structure of the MSLT model, providing a visual representation of its multi-scale organization. The root node broadcasts transactions and blocks to its neighbor nodes within the same scale. After receiving the messages, these neighbor nodes map their identities to the higher-scale and then broadcast the messages within the same scale. The same process occurs within all ranges.

3.3.1 Node Hierarchies and Range Division

In the Multi-Scale Node Management Model (MSLT), the classification of nodes and the division of the network into multiple scales play a crucial role in achieving an optimized P2P network for blockchain systems.

The MSLT model leverages the Node ID assigned to each node to create distinct hierarchies within the network. Based on the size of the Node ID, nodes are sorted and abstractly organized into multiple scales of different levels. Each scale represents a distinct layer of node categorization, enabling a hierarchical structure.

At each scale, nodes are evenly distributed into multiple ranges. The division of nodes into ranges ensures a balanced distribution across the network, promoting efficient data retrieval and communication. The number of ranges at the minimum scale (Scale 1) is denoted by the variable n. As the scale increases, the number of ranges grows exponentially, providing a scalable framework for node management.

For example, Scale 1 contains n ranges, and at Scale 2, each range from Scale 1 is further subdivided into n^2 additional ranges. This pattern continues as the scale increases, with the network eventually being divided into n^k ranges when the maximum scale value k is reached.

The establishment of multiple scales and the corresponding range division allows nodes to exhibit diverse identities at different scales. At a specific scale, each node belongs to a unique range, but its categorization may differ when viewed from another scale. This flexibility in node identity enables the MSLT model to adapt dynamically to changes in the network's requirements and conditions.

3.3.2 Node Relationships

In the Multi-Scale Node Management Model (MSLT), the organization of nodes into hierarchical scales introduces essential concepts that define node relationships within the network. These concepts play a crucial role in optimizing communication and data exchange among nodes:

1. Parent Range.

The "parent range" establishes a vital link between a node's current range and the range it belongs to at the previous scale. This relationship allows nodes to navigate efficiently within the hierarchical structure. Each node's parent range serves as a higher-level category, providing context and facilitating seamless data retrieval and communication.

2. Child Range

Conversely, the "child range" connects a node's current range to the range it belongs to at the next scale. This connection allows nodes to access and interact with subcategories within the hierarchy. By understanding their child ranges, nodes can explore the lower-level scales of the network and engage with more granular data and information.

3. Sibling Range

The concept of "sibling range" relates to all the sub-ranges within the previous scale to which a node belongs. Nodes sharing the same parent range are considered siblings in the network hierarchy. This relationship enables nodes to efficiently communicate and share information with other nodes that are part of similar sub-categories.

In Fig. 1, when the scale is 2, the parent range of node 0 is Range11, the child range is Range31, and the sibling range is Range21 to Range2n.

For the convenience of model expression, nodeID is defined as m-bits, and m is determined for any running blockchain system. The n is defined as the number of ranges when the scale is minimum (i.e., 1). Let k represent the size of the scale, with the minimum being 1 and the maximum being the smallest integer greater than or equal to $m/\log_2(n) - 1$. The j is defined as the serial number of the range at each scale, with the minimum being 1 and the

maximum is n^k.

According to the model, the maximum number of nodes in each range is:

$$N_{NodeInRange} = 2^m \Big/ n^k \tag{1}$$

Within each range, the minimum and maximum values of node ID are shown in Eqs. (2) and (3):

$$R_{\min} = 2^m (j-1) \Big/ n^k \tag{2}$$

$$R_{\max} = 2^m j \Big/ n^k - 1 \tag{3}$$

3.3.3 The Upper Bound of the Scale

The maximum scale k_{Max} is an important parameter in this MSLT model, choosing a proper k_{Max} can improve transmission performance without adding too much network costs. We employ the theory of complex networks [30] to solve this problem. The transmission path forms a tree-like topology, the message source node can be treated as the root of the tree, the degree of the node is d, the total number of nodes of the tree is M, and the height of the tree is h. The relationship between d, M, and h is:

$$\left(\left(d^h - 1\right)/(d-1)\right) \le M \le \left(\left(d^{h+1} - 1\right)/(d-1)\right) \tag{4}$$

$$d^h < M(d-1) + 1 \le d^{h+1} \tag{5}$$

$$h < \log_d(M(d-1)+1) \le h+1 \tag{6}$$

the minimum height of the tree can be obtained:

$$\begin{aligned} h_{\min} &= \lfloor \log_d(M(d-1)+1) \rfloor \\ &= \lfloor \ln(M(d-1)+1)/\ln d \rfloor \end{aligned} \tag{7}$$

The block propagation network generally has $d \gg 1$ and $M(d-1) \gg 1$. In this way, the above equation reduces to:

$$h_{\min}(d) = \lfloor \ln M/\ln d + 1 \rfloor \tag{8}$$

In this MSLT model, nodes are divided into n ranges at *Scale 1*, so the range number at *Scale 1* is $N_1 = n$. At *Scale 2*, each range of *Scale 1* is subdivided into n ranges, thus the range number N_2 at *Scale* 2 is n^2. In general, the range number at *Scale k* can be written as $N_k = n^k$. If there have r nodes in each range, the total number of nodes is $M = rn^k$, this means the upper bound of k is:

$$k_{\max} = \lceil (\ln M - \ln r)/\ln n \rceil \tag{9}$$

The transmission node in the MSLT model selects one node as its neighbor in all sibling ranges at different scales, each node has n neighbors at each scale. So, the degree d of the MSLT model is n. If this hierarchical structure covers all the nodes in the network, transmission level $N_{level} = k_{\max} + 1$ and N_{level} should be equal to $h_{\min}$, Thus:

$$k_{\max} + 1 = h_{\min}(n) \tag{10}$$

$$\lceil (\ln M - \ln r)/\ln n \rceil + 1 = \lfloor \ln M/\ln n + 1 \rfloor \tag{11}$$

$$\lceil (M - \ln r)/\ln n \rceil + 1 = \lfloor \ln M/\ln n \rfloor \tag{12}$$

Table 2 Node distribution at different scales.

Scale value	Maximum node count per range	Range count	Minimum node ID	Maximum node ID
1	2^{250}	1	0	2^{250}-1
1	2^{250}	2	2^{250}	2^{251}-1
1	2^{250}	64	2^{250}*(64-1)	2^{256}-1
2	2^{244}	1	0	2^{244}-1
2	2^{244}	2	2^{244}	2^{245}-1
2	2^{244}	64^2	2^{244}*(64^2-1)	2^{256}-1
42	16	1	0	15
42	16	2	2^{256}(2-1)/64^2=16	2*2^{256}/64^{42}-1=31
42	16	64^{42}	2^{256}(64^{42}-1)/64^{42}=2^{256}-16	2^{256}-1

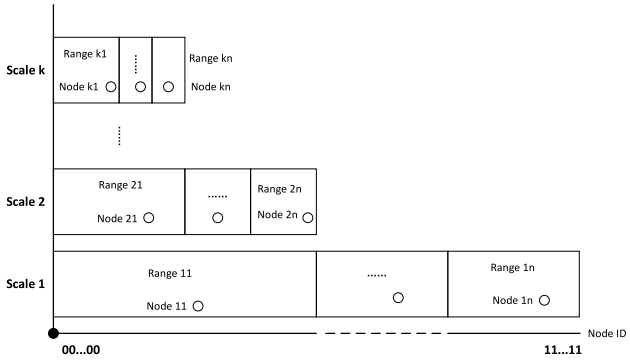


Fig. 2 Neighbor node structures at different scales.

So, we have:

$$r = n \tag{13}$$

This means, there is no need to set a higher scale when the maximum number of nodes contained in the range is less than or equal to n, as shown in Table 2, the maximum scale is 42. For instance, Table 2 displays the distribution of nodes at different scales, where m is 256 and n is 64.

3.4 Perspective Conversion

1. The perspective of a single node.
Each node in a blockchain P2P network must have a list of neighbor nodes to communicate with other nodes. Given that there are so many nodes in a P2P network, it is impractical to store all of them, thus a single node will select a few of its neighbors for storage. In this paradigm, each node chooses a neighbor from its sibling ranges at each scale. Figure 2 shows neighbor node structures at different scale.

Figure 2 shows an example of a node with an ID value of 0. At *Scale 1*, a neighbor node is chosen from each range between *Range 11* and *Range 1n*. Similarly, when the scale is 2, chosen from each range between *Range 21* and *Range 2n*. So, when the scale is k, a node is selected as the neighbor node from each range between *Range k1* and *Range kn*. The node has n neighbor nodes at each scale, for a total of kn neighbor nodes.
2. The perspective of all nodes
The range of a high-level scale is divided based on distinct ranges of its forward scale, and the sibling ranges of the for-

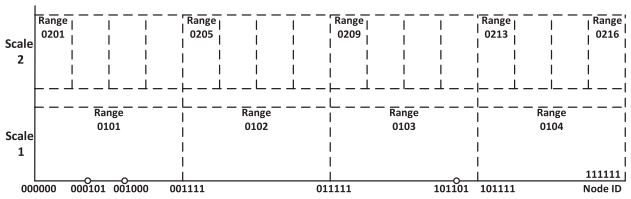


Fig. 3 Global perspective of nodes at different scales.

ward scale are independent of each other. As a result, nodes in different ranges of a certain Scale k must have different neighbors under the backward scale (*Scale* $n > k$), but their neighbors may be the same under the forward scale (*Scale* $n < k$). Further, if these nodes are in the same range under *Scale k*, with the increasing scale division strengths, these nodes tend to disperse to different ranges under the corresponding scale.

Their neighbors are bound to have differences under increasing scale, and these differences will gradually increase until they are completely different. Therefore, no matter how the nodes are distributed, their neighbors tend to be discrete with the increase of scale. The global perspective of nodes at different scales is shown in Fig. 3.

As shown in Fig. 3, at *scale 1*, *Node 000101*, *001000*, and *101101* can all choose any node as their neighbor node in each range of *Range0101-Range0104*, so their neighbor nodes may be duplicated; At *scale 2*, *node 000101* and *001000* can choose their neighbors in *Range0201-Range0204*, while *node 101101* can only select in *Range0209-Range0212*, resulting in completely different neighbor nodes for *Nodes 000101* and *101101*. Low repetition rates of neighbor nodes from various nodes promote the P2P network's quick spread. At the same time, the network's expansibility is enhanced concurrently.

3.5 Neighbor Node Updating

If a new node A connects to node B, node A will get the ID value of node B, then calculate the range where node B belongs at each scale, eventually, node A puts node B into the corresponding range of its neighbor node list.

After establishing a connection with another node C in the network, node A determines the interval to which node C belongs based on its ID value at different scales. If there already exists a neighboring node B in that interval, node C is added as a backup node. Within a certain heartbeat period, node A compares the network response speed of node C with that of node B. If node C's response speed is greater than that of node B, node C replaces node B as node A's neighbor in that interval; otherwise, node B remains as node A's neighbor. This approach of selecting the faster-responding node as the neighbor ensures that messages can be quickly transmitted to other nodes in the P2P network, thereby improving network transmission efficiency.

4. Hierarchical Transmission Design

4.1 Transmission Architecture

The multi-scale node management model is used to enable the efficient management of nodes in P2P networks. The network transmission based on this model can reduce the number of message transit times between the initiator and the target, and improve the transmission efficiency of the network. The transmission architecture based on the multi-scale node management model is shown in Fig. 4, when a node commences the transmission of data (such as transactional or block data), both the scale value and the data transmission-level value (data forwarding times) is set to 1. The node selects all of its scale 1-neighbor nodes and then adds a transmission-level data item (1 at this time) to send data to those nodes. Based on the additional transmission-level value, the receiver node increases the transmission-level value by 1 and updates the scale value to 2. Then, all scale 2-neighbor nodes of the receiver node are chosen to deliver data, and the transmission-level data item is added (2 at this time). This process continues until the scale value in the received data reaches the maximum, at which point the data broadcast is finished.

4.2 Transmission Process

Sending. The transmission node packages the data into a standard data format according to the blockchain-related protocol and attaches the transport-level data item when the node delivers the transactions or blocks. At this time, the data is sent for the first time, so the transmission-level value is 1. According to the transmission-level, the scale value is also determined to be 1, and all the neighbor nodes of the data transmitting node at the scale value are selected, and the data attached with the transmission-level is sent to these neighbor nodes of the nod. The detailed steps are as follows:

- Additional transmission-level data items. The transmission-level value is now set to 1, and the transmission-level data item is added to the transport data and signed using the data sender's private key.
- Determine the target neighbor node for data transmission. Since all nodes store lists of neighbor nodes with different scale values, the scale value can be determined as 1 according to the transmission-level value 1, and then all neighbor nodes with scale value 1 can be selected as the target neighbor node for data transmission.
- Send data. The sender node broadcasts packaged data, transport-level data, and transport-level signature data to target neighbor nodes.

As shown in Fig. 4, the neighbor nodes of *Node 0*, at *scale 1*, are *Node 0101*, *Node 0102*, *Node 0103*, and *Node 0104*. Therefore, when *Node 0* sends transaction or block data, it is first transmitted to these neighbor nodes to complete data

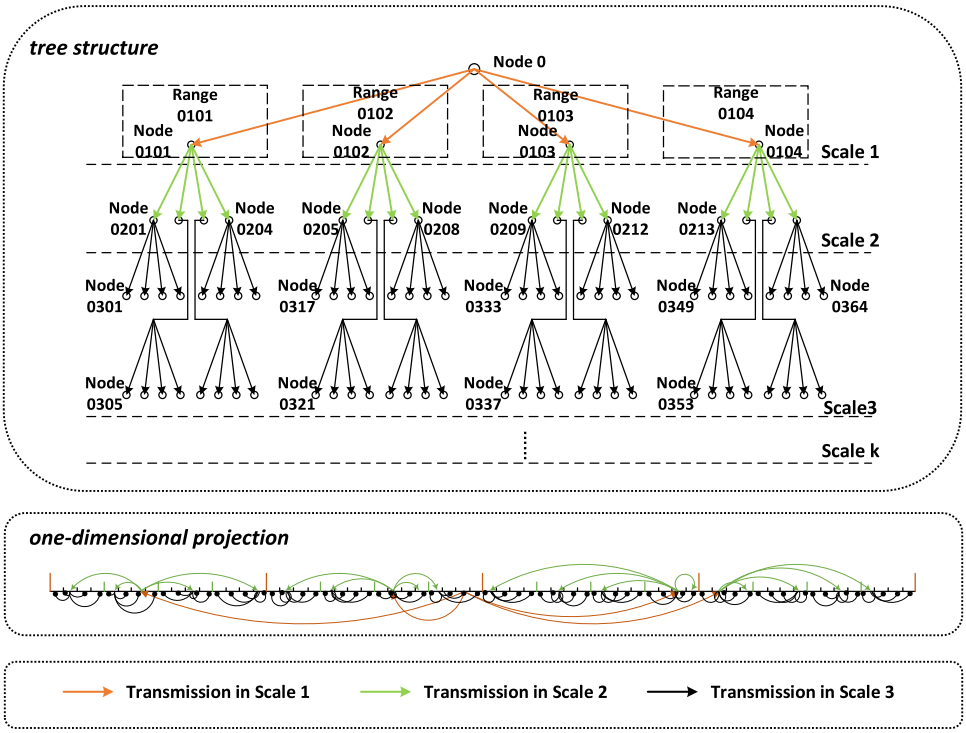


Fig. 4 Transport architecture based on a multi-scale node management model.

transmission at transmission-level 1.

Receiving and Forwarding. The node checks and forwards the data after receiving it. Here are the specific steps:

- Verification. The node verifies the validity of data such as transactions and blocks, at the same time, the minimum transmission-level should be greater than 1, and the maximum not exceed the maximum scale value; Then the node also validates the signature of the transmission-level, judge the validity of the signature.
- Forwarding. After the data has been validated, the node will determine data forwarding; First, add 1 to the transmission-level, and the result is used as the value of the data forwarding scale selected by the neighbor node; Then, determine if the scale value exceeds the maximum scale value of this blockchain system; At last, the node determines whether a neighbor node exists at this scale. The node will forward the data if the above requirements are satisfied.

The data receiving and forwarding algorithm is shown in Algorithm 1. After the data is forwarded, the next batch of nodes receiving the data repeat the process until the majority of nodes in the P2P network of the blockchain system have received the data.

Algorithm1: Data receiving and forwarding process

Input: dataReceived
Output: result //Result of receiving and forwarding data
// Parse the received data
1: dataTransmit, levelTransmit, levelSignaure ← Parse(dataReceived)
// Data verification
2: **if** IsNotVolidData(dataTransmit)
3: **return** false
4: **end if**
// Get the maximum scale value of the blockchain system
5: levelMax ← GetMaxScale()
// Verify transport level values and signatures
6: **if** levelTransmit<1 || levelTransmit> levelMax – 1 || IsNotVolidSignature(levelSignaure)
7: **return** false
8: **end if**
// Get the new Scale value of data forwarding
9: scale ← levelTransmit + 1
// Get a list of neighbor nodes for data forwarding
10: listNeighborNode ⇓ Get tNeighborNode(scale)
11: **if** listNeighborNode.Count == 0
12: return false
13: **end if**
// Attach transport level data and Signuare
14: dataSend ⇓ Pack (dataTransmit, levelTransmit + 1)
// Forward data to neighbor nodes
15: **for** neighborNode in listNeighborNode **do**
16: SendData(neighborNode. dataSend)
17: **end for**
18: **return** true

4.3 Transmission Reliability Design

At different scales, a node's neighboring nodes represent all nodes within their respective intervals. Failure of a neighboring node to receive data could cause all nodes within that interval to miss out on the same data, resulting in potential data loss and system instability. There are two methods to ensure the reliability of network transmission.

Neighbor nodes Backup. The data-sending node transmits data to one neighbor node in each range, so there may be

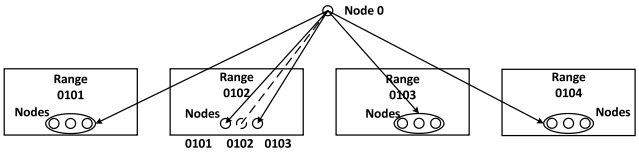


Fig. 5 Neighbor node backup scheme.

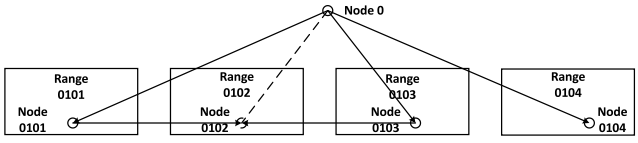


Fig. 6 Horizontal supplementary transmission scheme.

a single point of failure during transmission. The neighbor node backup scheme uses multiple neighbor nodes (typically set to 3) instead of one neighbor node, the nodes transmit data to three neighbor nodes in each scale to ensure reliable data transfer. The neighbor node backup scheme is shown in Fig. 5.

Figure 5 shows three neighbor nodes, 0101,0102,0103, and the data transmitting Node 0 sends data to all three at the same time. In this example, even if one Node (e.g., Node0102) does not receive data, the other two nodes may assure that there are nodes in Range0102 receiving data.

The strategy maintains data transmission reliability by increasing the redundancy of neighbor nodes, it also increases the number of neighbor nodes and the repetition of data transmission, increasing the network's transmission load pressure.

Network load balance. Distributed hash table networks have an inherent load-balancing problem. It is because consistent hashing produces a bound of $O(\log n)$. Bottlenecks can arise due to query overflow (too many queries received by the node at once) or data overflow (too much data needs to be forwarded by the node). In this work, we use the elastic routing table (ERT) algorithm [31] to deal with the network load balancing problem. We assume ci as the capacity that the node is willing to devote or able to process queries of *node*, *i* in practice, *ci* should be determined as a function of a node's access bandwidth, disk speed, etc. And *li* is the number of messages that the node receives and forwards to its neighbors over time T. Set *Si* as the load threshold. We refer to a node with $li/ci > Si$ as a heavy node, otherwise a light or un-overloaded node. Reduces the number of incoming connections to a node when the node is overloaded and lets the node's backup node handle forwarding a new message.

Transverse supplementary transmission. The node simultaneously transmits data to all neighbor nodes on the same scale when broadcasting, if one of the neighbor nodes does not get data, two adjacent neighbor nodes that have received data can send data again to this node. The transverse supplementary transmission scheme is shown in Fig. 6.

In Fig. 6, transmitting Node 0 simultaneously broadcasts data to all neighbor nodes 0101, 0102, 0103, 0104 at the same scale, where Node 0102 does not receive data. In this case, neighbor node 0101, 0103 can send data to Node 0102 again to ensure that Node 0102 can receives data.

In this scheme, data is transported again from the neighbor node that has received data to the neighbor node that has not received data, removing the problem of transmission failure between the two nodes due to network anomalies. However, transmission nodes are required to receive data reception response messages from each neighbor node. If the response information is not received, the two neighbor nodes that sent data must be notified again.

Combining the two schemes, it is clear that the neighbor node backup scheme has a high data transmission repetition rate and a low probability of data transmission failure. Consequently, the transverse supplementary transmission scheme is preferred.

Table 3 Parameter configuration of the prototype system.

Parameter	Value	Description
m	256	Length of the node ID
n	64	Number of target nodes per forwarding
$S_{bandwidth}$	10-50Mbps	Average network bandwidth between nodes
R_{fai}	≤50%	The probability that a node fails to transmit data
R_{mali}	≤1/3	Percentage of malicious nodes

5. Implementation and Analysis

5.1 Prototype System Design

We use GO [32] programming language to realize the prototype system of multi-scale node management and network transmission model. The parameters configured in the prototype system are shown in Table 3.

The prototype system is used to evaluate the transmission efficiency, network maximum throughput, network utilization, and security of the P2P network of blockchain system using a multi-scale node management model, and the network transmission of this model is compared with several other common P2P network transmissions of blockchain systems.

Different indicators in the blockchain system are evaluated, such as transmission level, transmission delay, transmission times, transmission repetition rate, maximum throughput, etc., and security is analyzed.

5.2 Communication Complexity

Communication complexity aims at studying the number of communication bits that the participants of a communication system need to exchange to perform certain tasks [33]. It directly affects data transmission efficiency; we try to discuss the transmission level and the lookup complexity of MSLT to explain the communication complexity of the model.

Transmission level. The transmission time is correlated with the transmission level when both the network bandwidth and the volume of data delivered remain constant. The

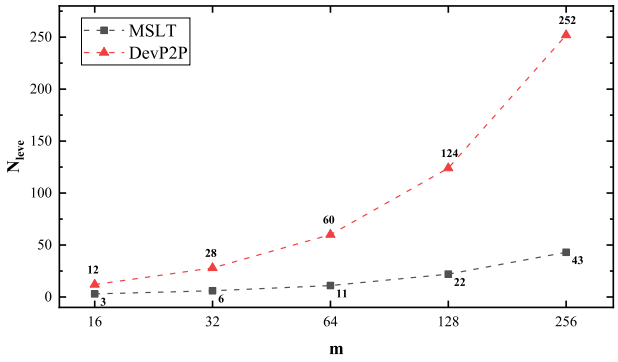


Fig. 7 Comparison of the transmitting level.

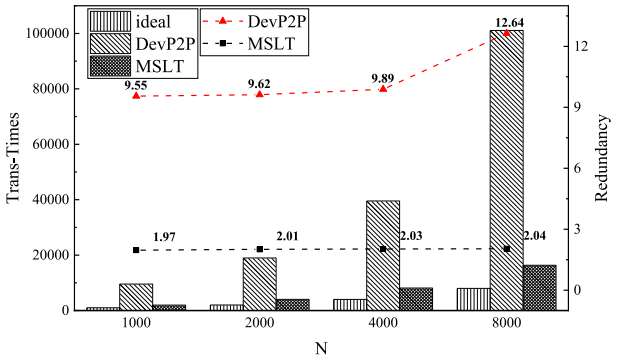


Fig. 8 Comparison of the transmitting redundancy.

maximum transmission level of the multi-scale node management model is:

$$N_{level} = m/\log_2 n \tag{14}$$

where m is the bit length of the node ID in the blockchain system. For any running blockchain system, m is determined. The n is defined as the number of ranges when the scale level is minimum (i.e., 1). Additionally, since there is always one chosen neighbor node in each range, n determines how many nodes will be broadcast at once.

Assume that in the MSLT model, the number of neighbor nodes that each node forwards data broadcast is the same as DevP2P in Ethereum, which is also 64. When using the MSLT model, the maximum transmission level value is calculated as 43 according to Eq. (14), much lower than DevP2P's transmission level. The comparison of the transmitting level between MSLT and DevP2P is shown in Fig. 7:

Lookup Complexity. DevP2P is based on the Kademila algorithm, the overlay routing is responsible for finding nodes according to their identifier (key) in the overlay. The routing path is a tree-like structure based on recursive lookup, so the lookup complexity of DevP2P is O(log(N)), which N is the number of network nodes. MSLT node management is also a kind of tree-like structure, but the setting of multiple scales makes the transmission level of this model lower. In this model, the maximum number of lookups should be equal to the number of transmission levels, according to Eq. (4), the lookup complexity of MSLT is O(log(N-r)), which N is the number of network nodes and r is the division strength.

5.3 Network Resource Consumption

Transmission redundancy. An effective solution to enhance reliability is introducing redundancy into message delivery. However, redundancy can also increase the network burden and resource consumption. Therefore, in blockchain networks, reducing transmission redundancy based on ensuring network-wide consensus is conducive to improving network performance and reducing network resource consumption.

In this MSLT model, data is transmitted hierarchically from the sending node to all nodes in the P2P network. When the transmission fails, the two nearest neighbors supplement the transmission of data, so the required times of

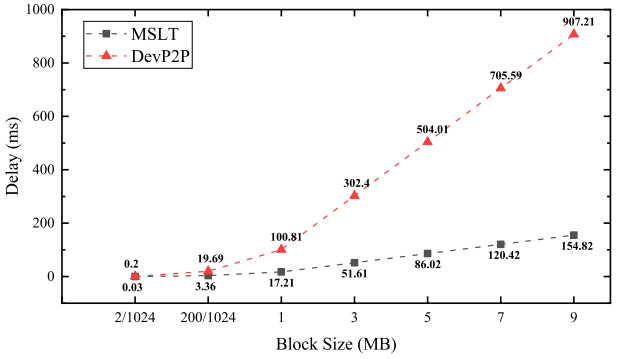
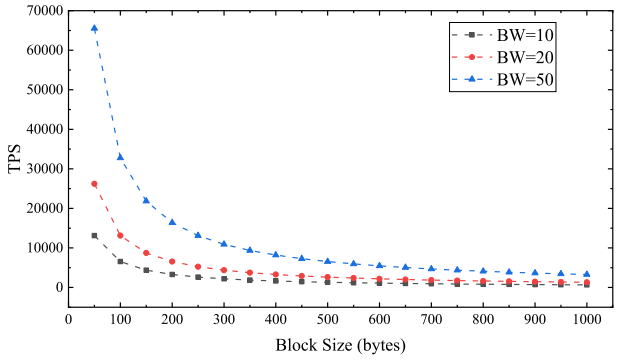


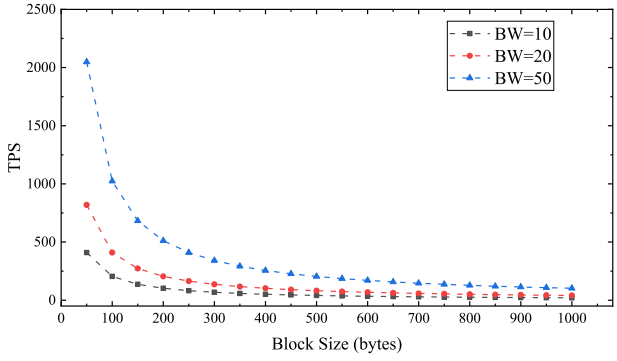
Fig. 9 Comparison of the transmission delay.

transfers are: $T_{transTimes} = N * (1 + 2 * R)_{fai}$. N is the total number of nodes, R_{fai} is forwarding data transmission failure probability, assuming that the maximum is less than 50%. Taking the structured DevP2P network of Ethereum as an example, set each node has three neighbors. The transmission times analysis of MSLT and DevP2P is shown in Fig. 7. The comparison of transmission times and redundancy between MSLT and DevP2P is shown in Fig. 8, N is the number of nodes, and the transmission redundancy of the MSLT model is about 2.0, while that of the DevP2P model is about 10.4, decreasing by about 80.8%.

Transmission delay. Taking the data transmission of Ethereum as an example, the average network transmission rate ($S_{bandWidth}$) is 20 Mbps, m is 256, and $N_{nodeInBucket}$ is 16. According to the transmission-level analysis, Ethereum's transmission-level is 252. If the transmission-level in the MSLT model is 43, Ethereum block height 14581702 has a block size of 189,646 Bytes [34]. The transmission delay of the Ethereum P2P network is 18.2 seconds, while the transmission delay of the MSLT model is 3.1 seconds. When the block size is 1 MB, the transmission delay of the Ethereum P2P network is 100.8 seconds, and the transmission delay of the MSLT model is 17.2 seconds. Transmission delay analysis is shown in Fig. 9, with the increase in the amount of transmitted data, the transmission efficiency advantage of the MSLT model becomes more obvious. The transmission delay of the MSLT network is relatively small, at least 82% lower than that of DevP2P.



(a) TPS in MSLT



(a) TPS in DevP2P

Fig. 10 Comparison of the transaction per second.

5.4 Maximum Throughput

The throughput is one of the main indicators of a blockchain system. Many factors affect its size, including consensus mechanism, data structure, encryption algorithm, transaction verification, P2P network status, etc. External factors include CPU performance, memory size, disk capacity, and network bandwidth.

While assuming that bandwidth is 10 Mbps, 20 Mbps, and 50 Mbps, the relationship between block size and transaction per second (TPS) is shown in Fig. 10, by comparison, the maximum throughput of the MSLT network is about 32 times that of the DevP2P network.

5.5 Security

The possible attack and their solutions mainly include the following aspects:

Node constructs false transmission level and signature data. The node may not comply with the data structure of transmission-level and constructs fake transmission-level and signature data when broadcasting. According to the description in Sect. 4.2, each node will verify the transmission-level and signature data according to the data receiving and forwarding algorithm. If the verification is unsuccessful, the node will ban the data-sending node and publish this information to other nodes. When the data-receiving node fails to verify data, it will inform the two sibling nodes and ensure that the local node can receive correct data through the data transmission of the two sibling nodes.

When the target node conducts data transmission, the two neighbor sibling nodes construct false transmission level and signature data again, then the malicious node attacks successfully. The probability R_{mali} that the first sibling node is malicious is shown in Eq. (15):

$$R_{mali} = N_{mali}/N_{bro} \tag{15}$$

where N_{mali} represents the number of malicious nodes in the neighbor nodes. In this way, the probability of R_{mali} the first N_{mali} sibling nodes are all malicious nodes is shown in Eq. (16):

$$R_{mali} = N_{mali}/N_{bro} \times (N_{mali} - 1) / (N_{bro} - 1) \times \ldots \\ \ldots \times 1/ (N_{bro} - N_{mali} - 1) \tag{16}$$

where, the number of sibling nodes N_{bro} is 63, and the number of malicious nodes in sibling nodes does not exceed 1/3. The probability that the first three closest sibling nodes are all malicious nodes is:

$$R_{mali} = 21/63 \times 20/62 = 10.75\% \tag{17}$$

The probability of the data-sending node constructing false transmission-level and signature data is less than 33.3%. Therefore, the probability of the malicious node successfully attacking is:

$$R_{atcSuccess} = R_{mali} \times 33.3\% = 3.58\% \tag{18}$$

In conclusion, through the horizontal supplementary transmission design of sibling nodes, the probability of a successful attack by constructing false transmission level and signature data is low. By increasing the number of sibling nodes with the horizontal supplementary transmission, the probability of a successful attack by malicious nodes can be further reduced.

Routing Table Attack [35]. In a P2P network, the nodes keep all of their neighbor node ID in the routing table. These nodes regularly update the information of the routing table. The attacker can exploit this process to force the peers to insert/ update the compromised node in the routing table of neighbor peers. This attack can cause severe damage to the network as it diverts the route of the message.

The routing table attack works by gathering the network topology and peer relationships and passing them to a route prediction procedure [35]. It will then predict routing paths between a pair of nodes. According to the description in Sect. 3.3, In the MSLT model, the range of a high-level scale is divided based on different ranges of its forward scale, and all the sibling ranges of the forward scale are independent. So, it forms multiple disjoint paths to forwarding data or lookup keys, and fewer intersecting nodes make it more difficult for an attacker to predict the routing path. Moreover, MSLT verifies a node's legitimacy to join the network by requiring it to complete resource-consuming

Table 4 Comparison of MSLT with other schemes.

Consensus	Decentralization	Latency	Redundancy	Scalability	Network Utilization	Fault Tolerance	Network Security
Bitcoin INV	High	High	Low	Low	Medium	Low	Medium
Ethereum NewBlockHashesMsg	High	Medium	Low	Medium	High	Medium	High
EOS blockchain	Medium	Low	Medium	High	High	High	High
MSLT	High	Low	Low	High	High	High	High

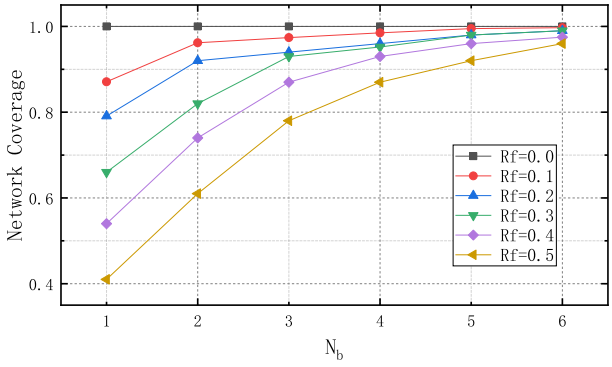


Fig. 11 Network coverage under different levels of malfeasance.

tasks (e.g., PoW [10]). This approach raises the cost of the attacker's misdeeds.

DDoS attack [36]**.** Broadcast protocols aim to distribute information to all nodes in the network, due to the flooding mechanism, the large amounts of data sent by malicious nodes may cause network breakdown. To avoid DDoS attacks, the MSLT model only sends data to nodes that have not received data, and the amount of data transmitted in the network is reduced. Meanwhile, the transmission path can be traced back to the first transmitting node, so the size and frequency of data transmitted by nodes can be limited.

Fault tolerance and recovery. Due to the complexity of the real network environment, nodes in the model can become faulty nodes, resulting in the loss of some correct block data. As is well known, to ensure the fault tolerance of a blockchain system, it can be addressed from two aspects: the design of consensus algorithms and the design of network transmission mechanisms. This research focuses on the latter, as described in Sect. 4.3, the MSLT model introduces the *Neighbor Nodes Backup* method, which is a redundant transmission mechanism that ensures the fault tolerance of the system.

To illustrate the operation of this model in a network with faulty nodes, simulation experiments were conducted. We randomly labeled a proportion R_f of faulty nodes in the network, which do not forward the data after receiving it. N_b represents the number of backup nod.

Figure 11 shows the network coverage in dependence of R_f and N_b: When $R_f = 0$, the network obviously has a 100% coverage rate. As the number of faulty nodes increases, the network coverage rate decreases significantly. However, the increase in N_b improves the poor network coverage situation. When $N_b \geq 3$ and $R_f \leq 0.3$, the coverage rate can reach over 90%. Therefore, we set the number of backup nodes for node transmission on the same scale to be 3.

5.6 Schemes Comparison

Table 4 lists the comparison of this model with three popular blockchain platforms: Bitcoin, Ethereum, and EOS, with a focus on various key metrics that determine their performance and suitability for different use cases. The comparison includes factors such as decentralization, latency, redundancy, scalability, network utilization, fault tolerance, and network security.

MSLT (Multi-Supervised Learning Trust) demonstrates several advantages over Bitcoin INV, Ethereum NewBlockHashesMsg, and EOS Blockchain. It exhibits strong decentralization with a distributed network of nodes, ensuring a robust structure. MSLT achieves lower latency through optimized transaction confirmations, and it introduces innovative neighbor node backup transmission and increased redundancy for higher data availability and fault tolerance. In terms of scalability and network utilization, MSLT performs well, efficiently processing a higher number of transactions and maximizing resource usage. Moreover, MSLT excels in fault tolerance, ensuring network stability during node failures, and maintains a high level of network security with its robust consensus model and security measures. These combined strengths make MSLT a competitive and reliable choice for various blockchain applications.

6. Conclusions

The node management of P2P networks affects the overall performance and reliability of the system. In this paper, a multi-scale node management model is proposed. MSLT set the network to multiple scales according to different division strengths. One scale corresponds to multiple equalization ranges. The transmission nodes select one node as its neighbor in each range and adopt the neighbor update method based on transmission speed. Through the network transmission between nodes update the neighbor nodes in each range, so that the transmission speed between nodes is always high. At the same time, based on the multi-scale node management model, a hierarchical transmission mode of a blockchain network is proposed, which is forwarded to neighbor nodes with corresponding scale values through different transmission levels.

According to research and calculations, the proposed MSLT model reduces the number of data transfers and transmission repetition rates and improves network utilization.

The future work of this paper is to apply the MSLT model to specific blockchain systems (such as Ethereum, etc.) to further analyze and improve transmission efficiency.

Acknowledgments

This research was funded by the National Key R&D Program of China under Grant nos. 2021YFB2700800; in part by the National Natural Science Foundation of China under Grant nos. 61602435.

References

[1] Y. Yuan and F. Wang, "Blockchain: The state of the art and future trends," Acta Automatica Sinica, vol.42, no.4, pp.481–494 2016. https://doi.org/10.16383/j.aas.2016.c160158

[2] H.B. Tan, T. Zhou, H. Zhao, Z. Zhao, W.D. Wang, Z.X. Zhang, N.Z. Sheng, and X.F. Li, "Archival data protection and sharing method based on blockchain," J. Softw. (China), vol.30, no.9, pp.2620–2635 2019. https://doi.org/10.13328/j.cnki.jos.005770

[3] S. Zeng, R. Huo, T. Huang, J. Liu, S. Wang, and W. Feng, "Survey of blockchain: Principle, progress and application," Journal on Communications, vol.41, no.1, pp.134–151, 2020. https://doi.org/10.11959/j.issn.1000-436x.2020027

[4] C. Pan, Z. Liu, Z. Liu, and Y. Long, "Research on scalability of blockchain technology: Problems and methods," Journal of Computer Research and Development, vol.55, no.10, pp.2099–2110, 2018. https://doi.org/10.7544/issn1000-1239.2018.20180440

[5] Z. Sun, X. Zhang, F. Xiang, and L. Chen, "Survey of storage scalability on blockchain," J. Softw. (China), vol.32, no.1, pp.1–20, 2021. https://doi.org/10.13328/j.cnki.jos.006111

[6] Y. Wu and J. Li, "Evolution process of blockchain P2P network protocol," Application Research of Computers, vol.36, no.10, pp.2881–2886, 2019. https://doi.org/10.19734/j.issn.1001-3695.2018.07.0365

[7] H. Barjini, M. Othman, H. Ibrahim, and N.I. Udzir, "Shortcoming, problems and analytical comparison for flooding-based search techniques in unstructured P2P networks," Peer-to-Peer Netw. Appl., vol.5, no.1, pp.1–13, 2012. https://doi.org/10.1007/s12083-011-0101-y

[8] R. Gaeta and M. Sereno, "Generalized probabilistic flooding in unstructured peer-to-peer networks," IEEE Trans. Parallel Distrib. Syst., vol.22, no.12, pp.2055–2062, 2011. https://doi.org/10.1109/TPDS.2011.82

[9] A.D. Liu, X.H. Du, N, Wang, and S.Z. Li, "Research progress of blockchain technology and its application in information security," J. Softw. (China), vol.29, no.7, pp.2092–2115, 2018. https://doi.org/10.13328/j.cnki.jos.005589

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Technical Report, Manubot, 2019.

[11] S. Delgado-Segura, C. Perez-Sola, J. Herrera-Joancomarti, G. Navarro-Arribas, and J. Borrell, "Cryptocurrency networks: A new P2P paradigm," Mobile Information Systems, vol.2018, pp.1–16, 2018. https://doi.org/10.1155/2018/2159082

[12] A. Antonopoulos, Mastering Bitcoin: Unlocking Digital Cryptocurrencies, O'Reilly Media, pp.32–40, 2014.

[13] P2P Network, 2018. [Online]. Available: https://developer.bitcoin.org/reference/p2p_networking.html

[14] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Project Yellow Paper, vol.151, no.2014, pp.1–32, 2014. https://ethereum.github.io/yellowpaper/paper.pdf

[15] Y. Yan, K. Zheng, and Z. Guo, Detailed Explanation and Actual Combat of Ethereum Technology, China Machine Press, pp.1–14, 2018.

[16] S. Katkuri, "A survey of data transfer and storage techniques in prevalent cryptocurrencies and suggested improvements," arXiv preprint arXiv:1808.03380, 2018. https://doi.org/10.48550/arXiv.1808.03380

[17] A. Clifford, Towards Massive On-Chain Scaling: Block Propagation Results with Xthin, Medium, Np, 4, 2016.

[18] D. Ding, X. Jiang, J. Wang, H. Wang, X. Zhang, and Y. Sun, "Tx-ilm: Lossy block compression with salted short hashing," arXiv. arXiv:1906.06500, 2019.

[19] D. Leung, A. Suhl, Y. Gilad, and N. Zeldovich, "Vault: Fast bootstrapping for the algorand cryptocurrency," 26th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA2019, pp.24–27, Feb. 2019.

[20] R. Matheson, "A faster, more efficient cryptocurrency," 2019. [Online]. Available: http://news.mit.edu/2019/vault-faster-more-efficient-cryptocurrency-0124

[21] Harmony Technical Whitepaper, 2020. [Online]. Available: https://harmony.one/whitepaper.pdf

[22] QUIC, a multiplexed transport over UDP. [Online]. Available: https://www.chromium.org/quic/

[23] EOS Whitepaper. [Online]. Available: https://eos.io/documents/EOS_An_Introduction.pdf

[24] Bitcoin's 'Nervous System' Gets an Upgrade with FIBRE Network. [Online]. Available: https://www.coindesk.com/markets/2016/07/21/bitcoins-nervous-system-gets-an-upgrade-with-fibre-network/

[25] U. Klarman, S. Basu, A. Kuzmanovic, and E. Sirer, "bloXroute: A scalable trustless blockchain distribution network whitepaper," IEEE Internet Things J., 2018.

[26] H. Yang, M.D. Liu, B.C. Li, and Z.Q. Dong, "A P2P network framework for interactive streaming media," 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, pp.288–292, Aug. 2019.

[27] R.T. Frahat, M.M. Monowar, and S.M. Buhari, "Secure and scalable trust management model for IoT P2P network," 2nd International Conference on Computer Applications and Information Security (ICCAIS), Riyadh, Saudi Arabia, pp.1–6, May 2019.

[28] B. Yu, X.F. Li, H. Zhao, and T. Zhou, "A scalable blockchain network model with transmission paths and neighbor node subareas," Computing, vol.104, no.10, pp.2253–2277, 2022. https://doi.org/10.1007/s00607-021-00913-1

[29] J.Z. Li, X.F. Li, H. Zhao, B. Yu, T. Zhou, H.T. Cheng, and N.Z. Sheng, "MANDALA: A scalable blockchain model with mesh-and-spoke network and H-PBFT consensus algorithm," Peer-to-Peer Netw. Appl., vol.16, pp.226–244, 2023. https://doi.org/10.1007/s12083-022-01373-w

[30] M.E.J. Newman, "Detecting community structure in networks," Eur. Phys. J. B, vol.38, no.2, pp.321–330, 2004.

[31] H.Y. Shen and C.Z. Xu, "Elastic routing table with provable performance for congestion control in DHT networks," IEEE Trans. Parallel Distrib. Syst., vol.21, no.2, pp.242–256 2010. https://doi.org/10.1109/TPDS.2009.51

[32] Golang. [Online]. Available: https://golang.org/

[33] M. Li, R.-R. Liu, L. Lü, M.-B. Hu, S. Xu, Y.-C. Zhang "Percolation on complex networks: Theory and application," Physics Reports, vol.907, no.1, pp.1–68, 2021. https://doi.org/10.1016/j.physrep.2020.12.003

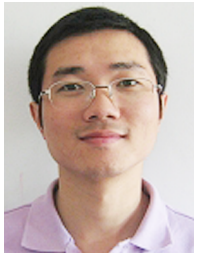
[34] Ethereum (ETH) Blockchain Explorer. [Online]. Available: https://etherscan.io/

[35] T. Baumeister, Y.F. Dong, G.Y. Tian, and Z.H. Duan, "Using randomized routing to counter routing table insertion attack on freenet," IEEE Global Communications Conference (GLOBECOM), Atlanta, GA, USA, pp.754–759, Dec. 2013. https://doi.org/10.1109/GLOCOM.2013.6831163

[36] DDoS. [Online]. Available: https://en.wikipedia.org/wiki/Denial-of-service_attack

Longle Cheng received M.S. degrees in Electrical Engineering from Anhui University in 2015, he is currently pursuing the Ph.D. degree with University of Science and Technology of China Hefei, China. His research interest covers blockchain scalability, P2P networks and consensus algorithm.

Xiaofeng Li is a research professor of Hefei Institutes of Physical Science, Chinese Academy of Sciences (CASHIPS), and a doctoral supervisor at the University of Science and Technology of China. He is the director of Information Center at Hefei Institutes of Physical Science, Chinese Academy of Sciences. His current research interests focus on blockchain technology, computer applied technology and measurement and control technology.



Haibo Tan is a research professor of Hefei Institutes of Physical Science, Chinese Academy of Sciences (CASHIPS). His current research interests focus on computer application and network data transmission performance optimization.

He Zhao received the Ph.D. degree from the University of Science and Technology of China in 2016, and B.S. and M.S. degrees from Nanjing University of Posts and Telecommunications in 2007 and 2010 respectively. He is currently a senior engineer at Hefei Institutes of Physical Science, Chinese Academy of Sciences (CASHIPS). His research interests include computer networking, blockchain technology and software architecture.

Bin Yu received the Ph.D. degree from the University of Science and Technology of China in 2021, and the M.S. degree from Hefei University of Technology, China, in 2007. His research interests cover blockchain technology and blockchain scalability.