Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.047166

ARTICLE

FPSblo: A Blockchain Network Transmission Model Utilizing Farthest Point Sampling

Longle Cheng[1,2], Xiru Li[1], Shiyu Fang[2], Wansu Pan[1], He Zhao[1,*], Haibo Tan[1] and Xiaofeng Li[1,2]

[1]Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei, 230031, China

[2]University of Science and Technology of China, Hefei, 230026, China

*Corresponding Author: He Zhao. Email: zhaoh@hfcas.ac.cn

Received: 27 October 2023 Accepted: 11 December 2023 Published: 27 February 2024

ABSTRACT

Peer-to-peer (P2P) overlay networks provide message transmission capabilities for blockchain systems. Improving data transmission efficiency in P2P networks can greatly enhance the performance of blockchain systems. However, traditional blockchain P2P networks face a common challenge where there is often a mismatch between the upper-layer traffic requirements and the underlying physical network topology. This mismatch results in redundant data transmission and inefficient routing, severely constraining the scalability of blockchain systems. To address these pressing issues, we propose FPSblo, an efficient transmission method for blockchain networks. Our inspiration for FPSblo stems from the Farthest Point Sampling (FPS) algorithm, a well-established technique widely utilized in point cloud image processing. In this work, we analogize blockchain nodes to points in a point cloud image and select a representative set of nodes to prioritize message forwarding so that messages reach the network edge quickly and are evenly distributed. Moreover, we compare our model with the Kadcast transmission model, which is a classic improvement model for blockchain P2P transmission networks, the experimental findings show that the FPSblo model reduces 34.8% of transmission redundancy and reduces the overload rate by 37.6%. By conducting experimental analysis, the FPS-BT model enhances the transmission capabilities of the P2P network in blockchain.

KEYWORDS

Blockchain; P2P networks; scalability; farthest point sampling

1 Introduction

Blockchain, employing cryptography and a decentralized network structure, is a distributed ledger technology designed to safeguard the security and integrity of data [1–3]. The node can equally perform data transmission and transfer transactions, achieving decentralized transaction management. This technology has undergone several stages of development, including the widely recognized blockchain 1.0-Bitcoin [4], blockchain 2.0-Ethereum [5], and emerging permissioned blockchains such as Hyperledger Fabric [6]. However, these solutions have not effectively addressed the real-time transaction requirements of users, restricting the blockchain's scalable implementation. Consequently, the scalability of this systems has emerged as a prominent subject in contemporary research [7,8].

Presently, efforts to enhance blockchain scalability can be categorized into layer 2 settlements (off-chain), layer 1 settlements (on-chain), and layer 0 settlements (P2P transport) within the realm of research [9]. Layer 2 settlements, like Payment Channel (Lightning Network [10]), Sidechain (Plasma [11]), off-chain computation (Arbitrum [12]), and cross-chain (Cosmos [13]), indirectly improve scalability. Layer 1 settlements prioritize tackling the architectural issues inherent in the blockchain itself, aiming to enhance system scalability. For instanceBlock Compression (Txilm [14]), Consensus Strategy Improvement (Bitcoin-NG [15]), and Sharding (Elastico [16]). In contrast to the above two solutions, Layer 0 aims to improve network performance by optimizing the transport protocol and topology of the underlying blockchain network, thereby improving scalability through measures such as reducing transmission latency, increasing throughput, improving load balancing, and reducing redundant transmissions. The advantage of this solution is that it does not require modifying the blockchain protocol or upgrading existing network nodes.

This paper concentrates on exploring Layer 0 solutions aimed at enhancing the transport performance of the foundational blockchain network. As we know, the distinguishing feature of blockchain is its reliance on peer-to-peer networks to verify and validate all trades [17]. The stability of a blockchain network is directly affected by the performance of its peer-to-peer network transmission. Nonetheless, certain issues persist within the P2P network of the blockchain system that requires resolution [18–20]:

1) Mismatch between the upper layer traffic and the underlying physical topology. This mismatch may arise when the traffic generated by the application layer deviates from the path defined by the physical connections among network nodes, resulting in inefficient routing and potentially long latency;
2) Imbalanced network traffic distribution. There is often a concentration of traffic in the blockchain network with traditional routing methods, leading to hotspots and potential congestion issues. This can create bottlenecks and reduce the efficiency of the network, especially in large-scale networks with a high volume of traffic;
3) Significant redundancy in data transmission. In a P2P blockchain network, the same data is often transmitted multiple times to achieve a large coverage, these redundancies can lead to inefficiencies and increased network congestion.

In this article, we proposed FPSblo, a blockchain network transmission model utilizing the farthest point sampling (FPS) method [21], given the aforementioned analysis of the pressing issues and requirements within the blockchain P2P network that demands immediate attention. The FPSblo model is location-based, the local nodes used the farthest sampling (FPS) algorithm to construct FPSblo nodes and form a preferred forwarding table. Blocks mined or transactions initiated by the local node were transmitted to the FPSblo node first, then the FPSblo node forwarded the blocks or transactions to its neighbor nodes. Our primary contributions can be succinctly outlined as follows:

1) We have developed a streamlined transmission approach for blockchain networks, inspired by the FPS algorithm commonly utilized in point cloud image processing.
2) We have developed a node management model to enhance the upper layer's awareness of the underlying physical topology.
3) This model has effectively reduced redundant transmissions within the blockchain network while improving load balancing without compromising network coverage.

The paper is structured as follows: In Section 2, we present and analyze previous work related to blockchain networks. Section 3 outlines the system design of the FPSblo transmission methodology. Section 4 provides an analysis and discussion of the transmission performance and security aspects of the data-transmitting model. Finally, Section 5 offers a summary of this paper and mapped the potential directions for future study.

2 Related Work

2.1 Farthest Point Sampling (FPS)

FPS stands out as a widely adopted algorithm for the selection of a subset of points from a larger set, typically used in computer vision, machine learning, and geometric modeling. As shown in Fig. 1, the input point cloud image of the red lamp has 2048 points. The blue lamp is sampled from the red lamp, and they have 512,256 and 128 points, respectively. The output image with the least number of points is only 6.25% of the input image, but still retains the base frame of the table lamp very well. So, the goal of FPS is to choose a set of points that maximizes the minimum distance between any two points in the subset, thereby ensuring that the selected points are well-distributed throughout the larger set [21].

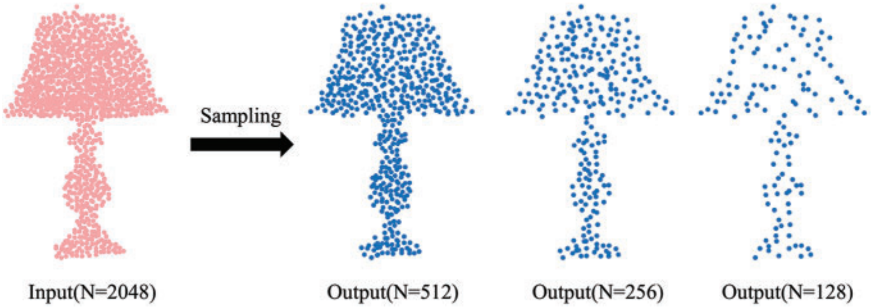

Figure 1: Results of FPS algorithm in different resolutions

The algorithm begins by selecting an initial point at random from the larger set. From there, it iteratively adds the point farthest from any point already in the subset until the desired number of points is reached. This process can be efficiently implemented using spatial data structures such as k-d trees or ball trees to quickly identify the farthest point at each iteration. FPS has numerous applications, including selecting representative samples for clustering, selecting keyframes in video summarization, and selecting points for mesh simplification. Its ability to efficiently select well-distributed points makes it a powerful tool in a wide range of fields.

The suggested approach of treating nodes in a P2P network as points within a point cloud image and utilizing the farthest point sampling algorithm to select feature nodes offers numerous advantages. First, It enables a reduction in the number of nodes needed to represent the distribution of the network, which makes it easier for upper-layer traffic to better perceive the underlying physical topology structure. Second, the feature nodes selected using the algorithm can provide a more comprehensive representation of the network, allowing for more accurate analysis and predictions of network behavior. Moreover, this approach can be particularly useful in large-scale P2P networks, where traditional methods of network analysis may be impractical due to the sheer volume of nodes involved. By using farthest-point sampling, we can obtain a more condensed and manageable representation of the network's topology. Overall, this approach has the potential to greatly enhance our understanding and management of P2P networks, particularly in the areas of routing, load balancing, and fault tolerance.

2.2 Blockchain Scalability: Network Transport Layer (Layer 0) Solutions

Layer 0 solutions are also called network transport layer solutions. Compared to Layer 1 (also known as on-chain scaling, refers to scaling solutions implemented at the blockchain's base layer protocol.) and Layer 2 (also known as off-chain scaling, refers to increasing transaction processing speed without altering the underlying blockchain protocol and fundamental rules, using methods such as state channels and sidechains.) approaches, there has been relatively little research on the expansion of blockchain networks at layer 0. Layer 0 refers to the underlying infrastructure of a blockchain, this includes the network layer responsible for validating transactions, creating blocks, and implementing consensus protocols. Therefore, enhancements or optimizations implemented at this level can significantly impact the overall performance of a blockchain network.

One promising area of research in layer 0 scaling is the development of protocols that enable more efficient message propagation across the network. We are specifically examining new proposals that have been introduced recently and categorizing them as layer 0 solutions.

Erlay [22] is a protocol specifically crafted to minimize the bandwidth necessary for transaction propagation within the Bitcoin network. This is accomplished by aggregating transactions into "compact blocks" and transmitting only block headers and the variations between the blocks to nodes that require this information. Such an approach significantly diminishes the volume of redundant data nodes required to transmit, thereby enhancing the overall efficiency.

Kadcast [23] is a protocol that uses a P2P overlay network to improve message propagation in a blockchain network. It builds on the idea of Kademlia, a popular P2P routing protocol, to enable faster and more efficient message routing between nodes. By using Kadcast, nodes in a blockchain network can more effectively discover and connect, which can significantly reduce the latency and bandwidth requirements of the network. Kadcast is a structured transmission network that builds a broadcast tree. In this network, each node is allocated a distinct identifier called a node ID. Nodes broadcast messages they receive with an attached transmission scope to peer nodes within the receiving scope. As shown in Fig. 2, if a local node initiates a broadcast, it partitions the entire network into four ranges. The node then selects a specific number of peer nodes in each range as its neighboring nodes and forwards the data. Subsequently, the node receiving data repeats the above process to complete the forwarding task.

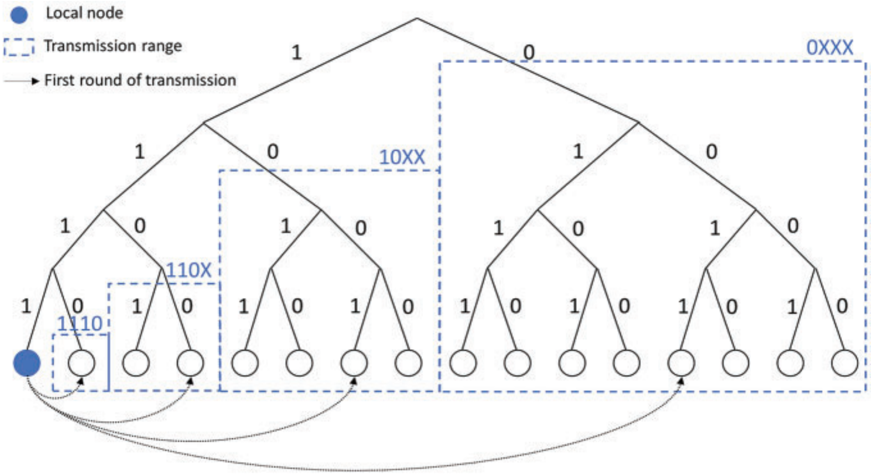

Figure 2: Schematic diagram of kadcast transmission process

BloXroute [24] built a protocol that Blockchain Distribution Network (BDN), aims to improve the scalability of blockchain networks by providing a dedicated, global, and decentralized network for block propagation. BloXroute operates as a layer-0 infrastructure that sits underneath the blockchain network and provides a set of APIs for nodes to interact with it. Nodes can submit transactions to BloXroute, which then aggregates them into blocks and broadcasts them to the network. By using BloXroute, blockchain networks can achieve higher throughput and lower latency, while also reducing the risk of network congestion and forked chains.

All three of these protocols demonstrate the potential for significant improvements in blockchain layer 0 scaling. By reducing the bandwidth required for transaction propagation, improving message routing between nodes, and providing a dedicated global network for block propagation, these protocols offer a way to achieve higher throughput and scalability in blockchain networks without sacrificing decentralization or security. However, these solutions still have their limitations. Erlay and Kadcast are researchers aimed at the Bitcoin blockchain network and have certain compatibility issues. Although BloXroute is compatible with various forms of blockchain architecture, its use of BDN technology to establish a cut-through routing incurs high costs. Therefore, researching a cost-effective and highly adaptable layer 0 blockchain network expansion solution becomes necessary.

3 System Design

3.1 FPSblo Management Modeling

Kadcast blockchain network node management model uses an application-layer topology to build a neighbor node management model, enabling application-layer multicast and facilitating node management. However, this approach lacks awareness of the underlying physical links, resulting in a mismatch between upper-layer traffic and lower-layer links. For example, if Node A in Asia wants to send data to Node B in Asia, it may need to be forwarded through Node C in North America, greatly reducing network transmission performance.

The FPSblo node management model is based on geographic location and utilizes the farthest point sampling algorithm mentioned above. The roles of FPSblo are classified into:

a) Local Node: the node that initiates the broadcast;
b) Starting Node: The node responsible for providing the FPSblo sampling node forwarding table to the local node, and running the FPSblo algorithm to calculate and select FPSblo sampling nodes;
c) FPSblo Sampling Node: a set of target nodes selected based on the FPS algorithm for the first-time transmission;
d) Transmission Node: ordinary nodes participating in message reception and forwarding;
e) FPSblo Alternative Node: serves as a candidate for the next update of the "FPSblo sampled node" set, contributing to the reduction of algorithmic complexity by providing nodes awaiting selection by the FPSblo algorithm during the subsequent update.

As depicted in Fig. 3, we consider the nodes within the blockchain network as points in point cloud images, and we have customized the FPS algorithm for effective node management in blockchain networks. The transmission process is as described below: The starting node will run the FPSblo algorithm to calculate the set of FPSblo sampling nodes starting from itself and write it into the FPSblo forwarding table. Then, when a local node has a broadcasting requirement, it will connect to the nearest starting node in terms of physical distance and fetch the forwarding table from the starting node, prioritizing the selection of nodes from it for broadcasting. Finally, nodes receiving the broadcast message will forward the message through the transmission nodes and repeat this process until the message covers the entire network.

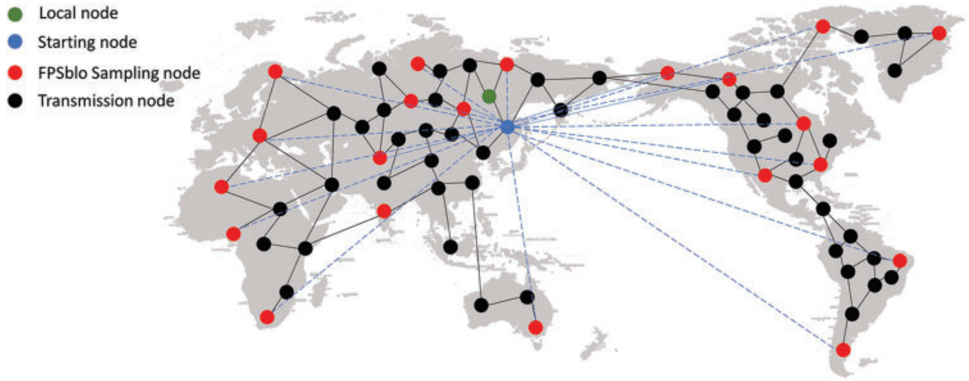

Figure 3: Schematic network topology of the FPSblo node management model

By employing the FPSblo node management model, we can propagate information quickly to the network's edge by selecting the farthest nodes. This approach offers several advantages:

a) Reducing Redundant Transmission: Choosing the farthest nodes as the starting points for propagation minimizes redundant transmission. This is because selecting nodes far from the initial node reduces the number of intermediary nodes during message propagation, thereby reducing redundant transmission.

b) Even Distribution of Messages: The FPSblo model ensures that the selection of nodes is uniform, helping to avoid concentration of messages in specific areas and enhancing the speed of message coverage across the entire network. This even distribution aids in fully utilizing network resources, improving the efficiency of the entire transmission process.

c) Swift Response to Demands: Propagating messages by selecting the farthest nodes means that messages can quickly reach the network's edge. This is particularly important for discrete nodes or those far from the core of the network. Rapid arrival at the network edge implies that messages can swiftly cover the entire network, thereby improving real-time transmission and efficiency.

In the next sections, we will describe this process in detail.

3.2 FPSblo Sampling Cluster Generation

The basic concept behind the FPSblo model is to select few representative sampling nodes from an extensive pool of nodes in the blockchain network (typically consisting of thousands of nodes). These sampling nodes should be evenly distributed throughout the network. The FPS algorithm selects nodes that meet this requirement, and we will first introduce how this sampling operates.

Kadcast blockchain network node management model is a kind of structured distributed hash table (DHT) algorithm, each node has a universally distinctive identifier known as "node ID", and the ID is random and meaningless, it does not care about the geographical location of the nodes, but the location largely affects the efficiency of message transmission. We treat the P2P network as a point cloud image, adapting traditional FPS algorithms to blockchain network transmission.

As shown in Fig. 4, suppose we have a set of N nodes $\{P_1, P_2, \cdots, P_n\}$, where each node represents a potential location for a peer in a P2P network. We want to select a subset of k nodes $\{Q_1, Q_2, \cdots, Q_n\}$

such that they cover the entire space with minimal overlap, i.e., each point in the space is within a certain distance d of at least one of the selected nodes.

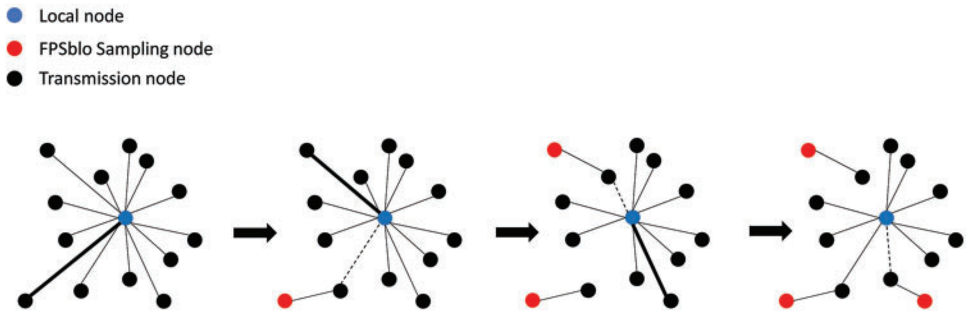

Figure 4: FPSblo node sampling process

To find the farthest point, we select a starting node Q_1 randomly. Then, we need to find Q_2, the farthest point from Q_1. Mathematically, Q_2 can be found as follows:

$$Q_2 = argmax\ dist\,(P, Q_1)\,, P \in \{P_1, P_2, \cdots, P_n\} \tag{1}$$

where $dist\,(P, Q_1)$ represents the geographical distance between node P and Q_1. We can repeat this process to find $Q_3, Q_4, \cdots, Q_k$, where each node is selected as the farthest from the previously selected nodes. This can be expressed as:

$$Q_k = argmax\ dist\,(P, Q_1 \cup Q_2 \cup \cdots \cup Q_{k-1})\,, P \in \{P_1, P_2, \cdots, P_n\} \tag{2}$$

where $Q_1 \cup Q_2 \cup \cdots \cup Q_{k-1}$ represents the union of the previously selected points. We can repeat this process until k nodes are selected, which will form the subset that covers the space with minimal overlap.

Algorithm 1: Node Sampling

```
Input: Node Range Q of shape (M, B)
// Start the sampling process
Dis[M] = {f};
Selected [n_node] = {a};
// Randomly pick an initial node index
initial_idx = random_integer(0, M);
// Store the initial node in the selected array
Selected [0] = initial_idx;
for idx = 1 to n_node - 1 do // Update distances array
    for j = 0 to M-1 do // Calculate geometric distance
        distance = geo_distance(P[j, :], P[initial_idx, :], D);
// If the distance is less than or equal to the current value in the distances array, update the distances array
        if distance ≤ Dis [j] then
            Dis [j] = distance;
        end
    end
// Choose the next node to be sampled, i.e., the node with the maximum distance
```

(Continued)

Algorithm 1 (continued)

```
    $initial\_idx = argmax(Dis)$;
    Selected[$idx$] = $initial\_idx$;
end
```

3.3 Distance Calculation

As mentioned in Section 3.2, the distance between nodes, $dist\,(P, Q)$, is one of the key input parameters in this algorithm. There are multiple ways to measure the "distance" between nodes, such as the number of hops required for data transmission, transmission delay, TTL (Time To Live) of the information, etc. The distance measurement used in this paper's management model is based on the physical location of nodes, this approach provides a more visual representation of the physical distribution of the network nodes. Currently, the mainstream network positioning methods include relying on location services provided by other systems, IP address-based positioning and ping-based active testing. For easy implementation, this paper chooses IP address-based positioning technology, ignoring the service bottleneck problem of centralized points and the problem of locating hosts on different subnets.

FPSblo protocol employs the haversine formula to assess geographical distance. In particular, this formula is utilized to compute the actual separation of two nodes located on the land face of the Earth, specified in terms of longitude and latitude. Consequently, we extract the latitude and longitude from the node's IP address using the *MaxMind GeoLite* City database. There are two nodes A (λ_1, φ_1), B (λ_2, φ_2) on the earth, where λ is the longitude and φ is the latitude, d is the spherical distance between nodes A and B, and R is the Earth's radius. Haversine is defined as:

$$hav\,(\theta) = sin^2(\theta/2) = ((1 - \cos(\theta))/2) \tag{3}$$

The relationship between d and R is:

$$hav\,(d/R) = hav\,(\Delta\varphi) + \cos(\varphi_1)\cos(\varphi_2)\;\; hav(\Delta\lambda) \tag{4}$$

So, we have

$$d = 2R\arcsin(\sqrt{sin^2(\Delta\varphi/2) + \cos(\varphi_1)\cos(\varphi_2)\,sin^2(\Delta\lambda/2)} \tag{5}$$

During communication between nodes, basic information like IP is included. The *MaxMind GeoLite* City database facilitates the retrieval of geographical latitude and longitude coordinates associated with individual IP addresses. By applying the formula mentioned above, the physical distance measurement between nodes can be obtained. The specific process of node discovery will be detailed in subsequent chapters. It is worth noting that this method assumes the usage of public IP addresses and does not consider scenarios involving proxy servers or network address translation (NAT).

3.4 Node Joining

As a new node tries to join the network, it usually needs to establish a connection with other nodes to participate in the distributed ledger system. One way to do this is to link with a seed node, which is an existing node that has been designated as a starting peer for new nodes joining the network, providing new nodes with information about the blockchain. We offer two options for nodes to join the network, join as a transmission node or join as an FPSblo alternative node.

3.4.1 Join as a Transmission Node

FPSblo is an algorithm that consumes resources, and it operates on the premise that the capabilities of nodes within the blockchain P2P network are not uniformly distributed. To save computational expenses, nodes can choose to join the network as transmission nodes. From the perspective of data circulation, these nodes only need to handle data reception and forwarding and do not participate in the task of sampling nodes for FPSblo. Fig. 5a shows the procedures for a new node to become a transmission node in the FPSblo network as follows:

1) Connect to the seed node: In a blockchain network, seed nodes serve as the initial points of contact for new nodes joining the network. Seed nodes are typically maintained by the blockchain's development team or community, and they provide a list of IP addresses or domain names for nodes to connect to when they first join the network;
2) Request starting node: As mentioned in Section 3.2, the seed node randomly picks a node in its neighbor list as the starting node of FPSblo. As shown in Fig. 5, there are 3 seed nodes in this network, each seed node has a randomly generated starting node, the local node will request information about all starting nodes from the seed node and calculate the geographical district of local node from the starting node, then select the starting node with the shortest distance.
3) Request to join FPSblo range: Once the new node has linked with the seed node, and chosen the appropriate starting node. It can request a randomly generated FPSblo table of nodes from the starting node. This table contains a list of nodes that have been selected using the FPS algorithm, based on their distance and correlation to the new node.

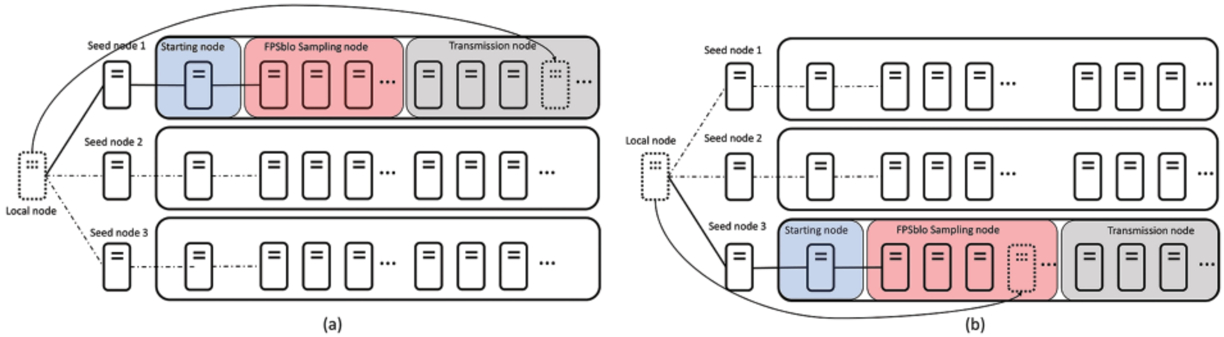

Figure 5: New node joining process

3.4.2 Join as FPSblo Alternative Node

FPSblo alternative node refers to a choice or candidate for the next node to be added to the subset being selected by the FPSblo algorithm. The new node can also choose to join the network as an FPSblo alternative node, as part of the construction of the FPSblo sampling table.

As shown in Fig. 5b, same as joining the network as a transmission node, the new node first connects to the seed node randomly and gets the current FPSblo table of the seed node. Subsequently, the new node sends a joining request to the seed node, seeking to integrate into its FPSblo alternative table, when the seed node starts the next round of FPSblo sampling node generation, the nodes that were previously requested to be added to the alternative list and vetted will be used as the alternative set.

By using this model, the new node can establish connections with a set of nodes that are most likely to provide useful and diverse information, while avoiding connections with nodes that are too similar or too closely connected to other nodes. This can help to ensure that the new node is integrated into the network in a way that promotes decentralization and resilience, while also optimizing its performance and efficiency. Once the new node has established connections with other nodes in the network, it can begin participating in the validation and verification of new transactions, contributing to the overall security and integrity of the blockchain. We designed an incentive mechanism to motivate nodes in the FPSblo alternative node-set based on service time and quality. This way, nodes will strive to provide services, contributing to the enhancement of network performance.

3.5 Node Maintenance

As mentioned in Section 3.2, FPSblo is a greedy, iterative algorithm that incurs significant computational overhead during the sampling point generation process. In the blockchain P2P network, nodes can join and exit flexibly, and node identities are peer-to-peer, with no performance restrictions on admission mechanisms, making the performance of nodes in the entire network uneven. Therefore, it is difficult to achieve the participation of all nodes in the generation and maintenance of sampling points in practical operations. In this model, the generation of the sampling table is entirely undertaken by the seed nodes in the blockchain network. Seed nodes are also called boot nodes, They serve as an initial point of contact for new nodes joining the network, and provide new nodes with information about the blockchain. This helps new nodes quickly synchronize with the network and begin participating in the blockchain's consensus process.

There are five seed nodes in this model, as shown in Fig. 6. Each seed node randomly generates five initial nodes and uses the FPSblo algorithm to generate five sampling tables. Upon establishing a connection with the seed node, the local node receives the seed node's current list of initial nodes. Subsequently, the local node computes the distance between itself and each initial node, selecting the one with the shortest distance. It then adds the nodes from the selected initial node's sampling table to its forwarding list.

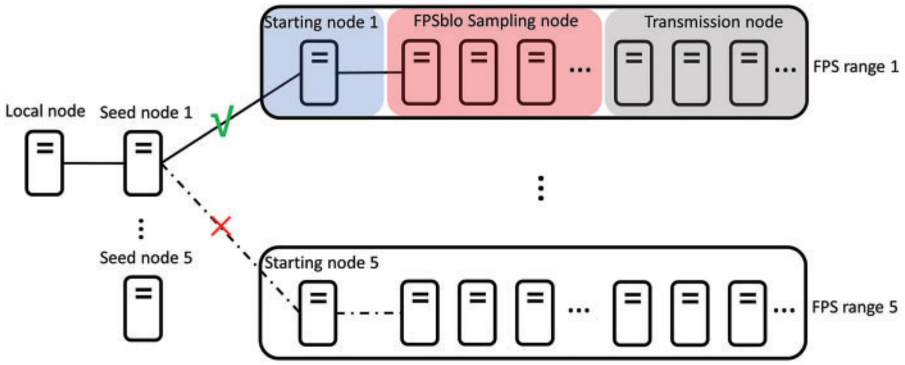

Figure 6: Optimized node joining process

From the perspective of resource consumption, each seed node maintains 5 FPSblo forwarding tables in this model, and there are 5 seed nodes in total, which means there are 25 FPSblo forwarding tables in the entire network. FPSblo forwarding tables are not fixed and unchanging. Fig. 7 displays the comprehensive process of generating and sustaining the FPSblo tables. At the outset, the Table object is initialized by utilizing either external or default parameters. Following this, the seed nodes are loaded into the system and the FPSblo tables are initialized. Once the initialization is completed, the adding loop is initiated, which continually adds new nodes to the FPSblo table. The system ensures that the FPSblo table remains up-to-date by periodically removing expired nodes using the *expireNodes ( )* function. This function refreshes the database and eliminates nodes that have exceeded their expiration time. The event listener *loop( )*, is responsible for ensuring the stability of the FPSblo table. This loop operates by continuously validating the FPSblo table nodes to ensure that they are still valid and reliable, it also maintains the FPSblo table by writing stable sampling nodes into it. By performing these actions, the event listener *loop( )* ensures that the FPSblo table remains an accurate representation of the underlying data and that it is reliable for use in various applications.

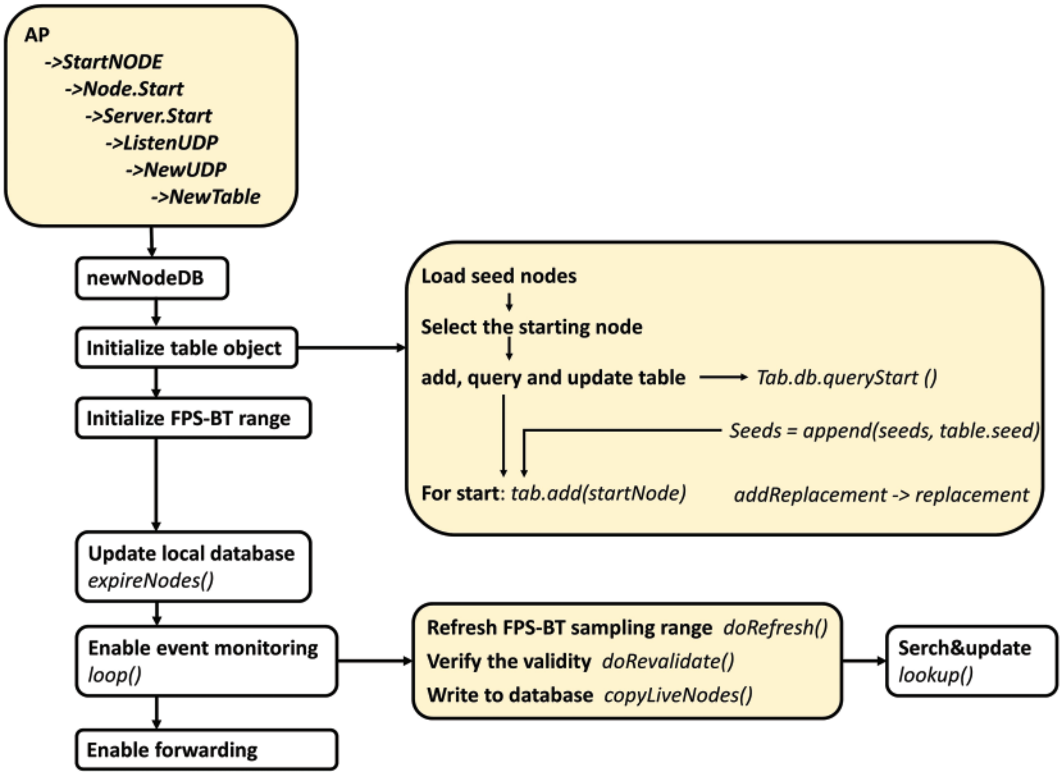

Figure 7: Node discovery and maintenance process

3.6 Transmission Convergence

The network transmission coverage is one of the important indicators for assessing the effectiveness of a node management model. The performance of the same algorithm may vary in different operating environments, which is influenced by various factors such as network bandwidth, operation and maintenance, and malicious nodes. In the following Section 4.2 on security, we will demonstrate through simulated experiments how the transmission coverage rate will be affected in the event of the existence of malicious nodes. This chapter discusses the convergence problem of the network coverage of the FPSblo model under the assumption of no external influence, i.e., all nodes are peer-to-peer and reachable.

Suppose there are n nodes within the P2P network of the blockchain, we need to select k nodes as FPSblo representative nodes. First, we define some symbols and concepts:

- S_k: The set of k FPSblo representative nodes selected by this model.
- s_k: The latest node in set S_k.

- $d(p_i, S_k)$: The shortest distance between node p_i and the representative node set S_k.
- $D_{max}(S_k)$: The minimum distance from the farthest node in the representative node set S_k to the remaining nodes.

We can then use the following lemma to prove the convergence of coverage:

Lemma1: For any $k \geq 1$, we have $S_k \subseteq S_{k+1}$, i.e., the selected set of points gradually increases with the number of iterations.

Proof: Assuming that after the *i-th* iteration, the selected node set is S_i, then we have:

$$d_i = \min_{j=1,\cdots,i-1} \left| s_i - s_j \right| \tag{6}$$

According to the definition of the FPSblo algorithm, for $2 \leq i \leq k$, s_i is selected as the point that satisfies the maximum d_i. Therefore, we can obtain:

$$d_i \geq d_{i+1}, \forall i = 1, \cdots, k-1 \tag{7}$$

This means that as the number of iterations of the algorithm increases, each newly selected sampling point becomes farther and farther away from the previously selected sampling points. Thus, we can conclude that S_k is monotonically increasing. Thus, $S_k \subseteq S_{k+1}$, the proof of Lemma1 is completed.

Lemma2: For any $k \leq n$, we have $d\left(p_i, S_k\right) \leq D_{max}\left(S_k\right), \forall i \in [1, n]$.

Proof: Assume there exists $i_0 \in [1, n]$ such that:

$$d(p_i, S_k) > D_{max}(S_k) \tag{8}$$

According to the definition of the FPSblo representative node set, we can select a representative node p_j in S_k that satisfies $d\left(p_j, p_{i_0}\right) = D_{max}(S_k)$. According to the triangle inequality, we have:

$$d\left(p_{i_0}, S_k\right) \leq d\left(p_{i_0}, p_j\right) + d\left(p_i, S_k\right) = D_{max}\left(S_k\right) + d\left(p_{i_0}, p_j\right) \tag{9}$$

Also, $d\left(p_{i_0}, p_j\right) \leq D_{max}\left(S_k\right)$. Therefore, we have:

$$d\left(p_{i_0}, S_k\right) \leq 2D_{max}\left(S_k\right) < d\left(p_{i_0}, S_k\right) \tag{10}$$

The statement formula (10) is contradictory to the assumption condition formula (8). Thus, the lemma is proved. According to the lemma, we can obtain the following conclusion:

Conclusion: When $k \to n$, the representative node set S_k is capable of encompassing all nodes in the network.

Proof: According to the lemma, for any node p_i, we have $d\left(p_i, S_k\right) \leq D_{max}\left(S_k\right)$. Also, the shortest distance between the farthest node in the representative node set S_k and the other nodes is $D_{max}\left(S_k\right)$ Therefore, the representative node set S_k is able to encompass all nodes within the P2P network. Thus, the conclusion is proved. When $k < n$, according to the conclusion, the representative node set can still cover all nodes, but may require more nodes to achieve complete coverage. In summary, when k approaches n, the coverage of the representative node-set obtained by the FPSblo model converges.

4 Execution and Evaluation

4.1 Design of the Prototype System

Our FPSblo transport model simulation research simulation platform is anchored on the Blockchain Network Simulator (BNS) [23] designed by Rohrer et al. BNS is based on ns-3 [25], an open-source discrete event network simulator designed for simulating. It offers a flexible framework for users to customize network topologies, configure protocol parameters, and simulate network behavior using an event-driven model. We have partially modified the topology and transmission method to meet the experimental requirements. Table 1 displays the configured parameters within the prototype system.

Table 1: Configuration of prototype system parameters

Parameter	Value
Node distribution	Power-law model, exponent of 2
Node coordinates	Latitude [−90, 90] Longitude [−180, 180]
Number of nodes	256, 512, 1024, 2048
Percentage of FPSblo nodes	0.3
Average network bandwidth	10–50 Mbps
Number of requests a node can process per unit of time	10–100

The experiments were performed with the help of a SIMT accelerator. The cluster comprises numerous nodes, each equipped with 1 CPU and 4 accelerators. The CPU is structured with four NUMA nodes, and each NUMA node incorporates 8 X86-based processors. The accelerator employs a GPU-like architecture, featuring a 16 GB HBM2 device memory and multiple compute units. The connection between the accelerators and the CPU is established via PCI-E, with a peak bandwidth of 16 GB/s for data transfer between main memory and device memory.

We assessed the transmission performance using a prototype system and compared FPSblo to Kadcast's network transmission systems.

4.2 Transmission Performance

4.2.1 Load Balance

P2P networks that use structured Distributed Hash Tables (DHTs) for information transfer inherently face load balancing challenges, primarily stemming from the O (log n) bound introduced by consistent hashing. Bottlenecks can manifest in the form of query overflow, where a node receives an excessive number of queries simultaneously, or data overflow, where the node is tasked with forwarding an overwhelming volume of data.

Two metrics are employed in this experiment to gauge the model's performance with respect to load balancing.

(1) Load balancing state of nodes.

First, we define some parameters. C_i (normal distribution between 10–100): Ability of nodes to handle queries requested by other nodes; S_i: The load threshold; L_i: Number of received and forwarded messages by the node to its neighbors in time T. We classify a node as an overloaded node if $L_i/C_i > S_i$, otherwise, it is categorized as a light node.

Fig. 8 displays a comparison of load averaging distribution of nodes for Kadcast and FPSblo. The results show that the average load value of the maximum loaded node is reduced, while the minimum loaded node remains the same. The FPSblo algorithm selects nodes with the maximum spatial distance as sampling nodes, which means that the selected nodes are more dispersed and evenly distributed in space compared to other nodes. By distributing the load evenly across different nodes, it avoids the situation where some nodes are overloaded, thereby improving load-balancing performance.

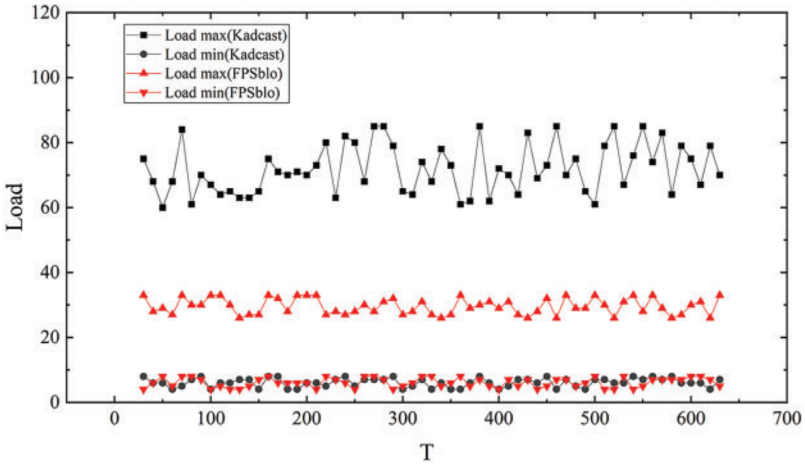

Figure 8: Load balancing state of network nodes

(2) Overload rate of nodes.

$R_{overlode}$ describes the percentage of overloaded nodes in the entire network, the formula is:

$$R_{overlode} = N_{overlode}/N_{node} \tag{11}$$

As shown in Fig. 9, compared to Kadcast, the overload rate of nodes in the network with the FPSblo model is reduced by 37.6%. the FPSblo model ensures that the load is distributed evenly across the network. This reduces the chances of any node being overloaded and ensures that the network resources are used efficiently.

The FPSblo model selects nodes with the maximum spatial distance as sampling points. This selection process ensures that the workload is evenly distributed among the nodes in the network. By avoiding situations where a few nodes are overloaded with excessive tasks, the algorithm helps achieve a more balanced distribution of the workload across the network.

4.2.2 Transmission Redundancy

Minimizing the redundancy in blockchain network transmissions can yield significant benefits to it. This reduces the amount of data that needs to be propagated, thus increasing the efficiency and throughput of the network and validated by nodes, resulting in faster transaction processing times and reduced latency. Additionally, reduced redundancy can lead to lower transaction fees and better scalability by freeing up network bandwidth and reducing the computational load on nodes.

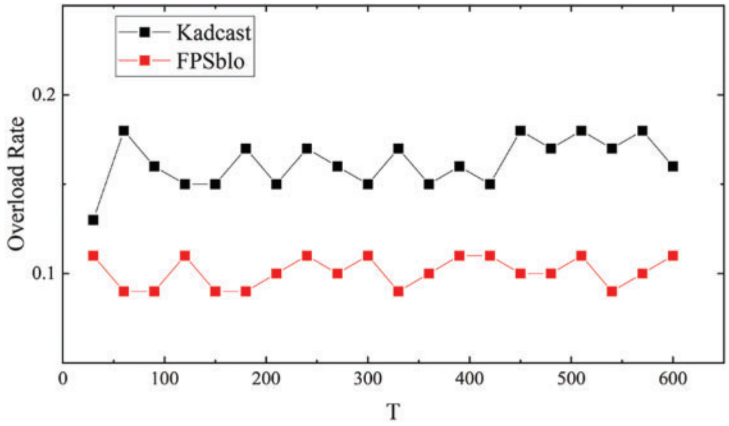

Figure 9: Overload rate of network nodes

As show in Fig. 10, we observed the number of messages received per node per unit time for the FPSblo model and Kadcast at different node scales, denoted by α. The comparison assumes that both models achieve the same network coverage in terms of message transmission. Taking a node scale of 256 as an example, the maximum and minimum values of α for the FPSblo model are 10 and 1, respectively, while for the Kadcast model, the maximum and minimum values of α are 19 and 3, respectively. The line chart representing the α values of different nodes in the FPSblo model is consistently below the line chart of α values for the Kadcast model. To illustrate this trend more intuitively, LE-smoothing was applied to both lines, showing consistent results. Further observations at node scales of 512, 1024, and 2048 reveal a similar pattern as observed at a node scale of 256.

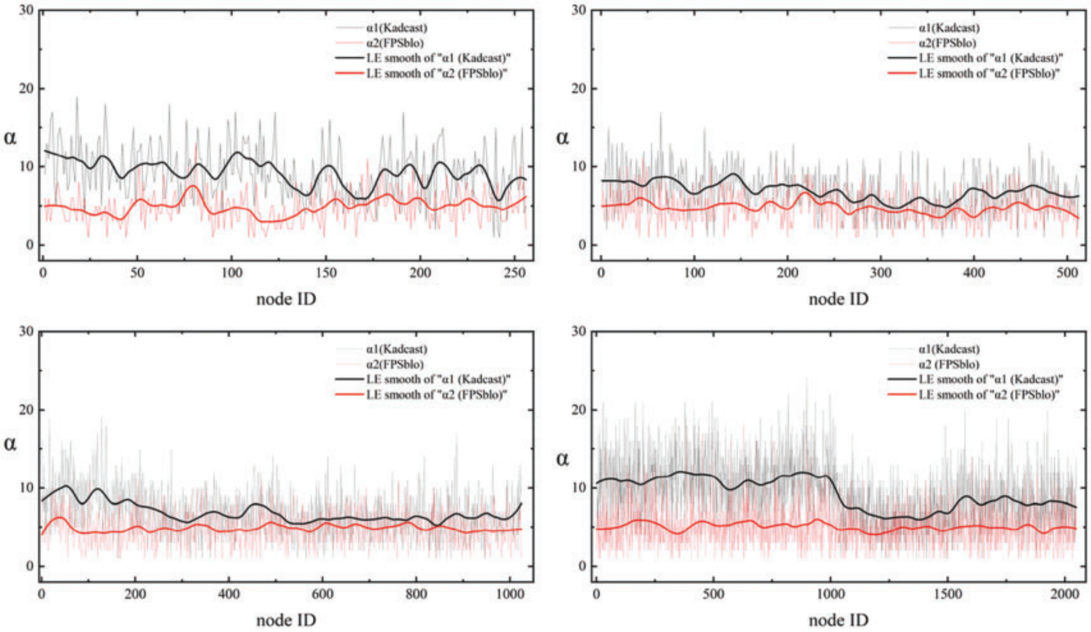

Figure 10: The number of message transmissions per unit of time for a single node at different network scales

Message transmission rounds refer to the frequency of blockchain network transmission. It represents the number of times data is transmitted or broadcasted within the network. The transmission rounds are determined by the broadcast protocol used in the blockchain network. Fig. 11 displays the amount of redundant messages generated in the network and the number of message transmission rounds for both models when completing a single network-wide broadcast at different node scales. The FPSblo model has almost the same number of transmission rounds as the Kadcast model, but as the node scale increases, the redundancy of messages is greatly reduced.

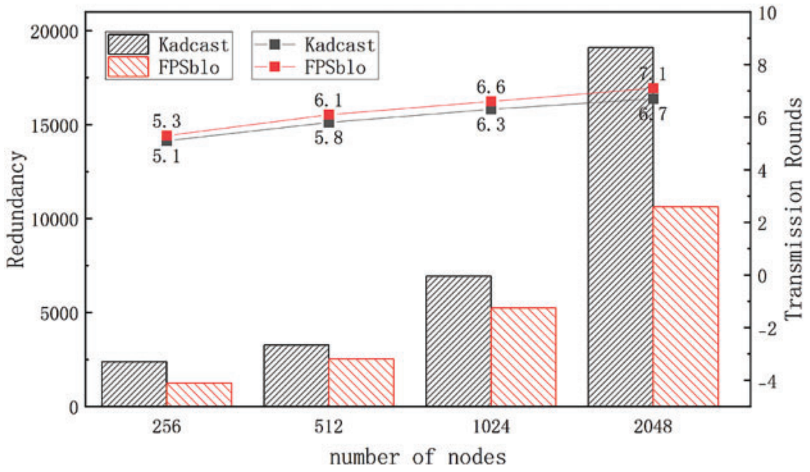

Figure 11: Transmission redundancy and rounds during a single broadcast in the network

The selected nodes by the FPSblo model have a lower correlation with other nodes. Transmitting data with nodes that are far away, reduces redundant data transmission between nodes, and data distribution across the network becomes more extensive, reducing redundant data along transmission paths. This helps to decrease transmission redundancy and optimize the overall data transmission efficiency of the network. This helps to decrease transmission redundancy and avoids unnecessary data duplication.

The experimental results from Figs. 10 and 11 indicate that, with the same message coverage, the FPSblo model exhibits lower transmission redundancy compared to the Kadcast model. This is attributed to the fact that the FPSblo model considers the physical locations of neighboring nodes when selecting nodes. With the assistance of FPSblo sampled nodes, broadcast messages can cover the entire network more rapidly and evenly. The FPSblo forwarding table generated using the farthest point sampling algorithm helps optimize the selection of paths for message transmission. As a result, under the same node scale, the FPSblo model avoids more redundant paths in message transmission, leading to a more streamlined message processing for each node.

4.3 Security

Denial of Service Attack (DoS) [26]. The purpose of the broadcast protocol in the blockchain system is to distribute blocks efficiently to all nodes. Malicious affect the normal broadcasting of messages by stopping services. To demonstrate the operation of the FPSblo model in the presence of a denial of service by malicious nodes, we conducted experiments. We set a certain number of nodes to refrain from forwarding messages upon receipt, simulating malicious behavior. This quantity is represented as R_m, a proportion of the total nodes in the network. Additionally, different levels of

FPSblo sampling ratio (R_f) were configured. The relationship between these two factors is illustrated in Fig. 12. Clearly, When this network is not always honest with all the nodes, the final network coverage approaches 1 due to the convergent nature of FPSblo transmission. As the proportion of nodes engaging in denial-of-service increases, the network coverage decreases. However, with an FPSblo sampling ratio exceeding 0.3, and when the proportion of malicious nodes is less than 0.3, the network message coverage can still reach 0.9. This is attributed to the early-stage effectiveness of the FPSblo model in spreading messages further, thereby enhancing the system's robustness.

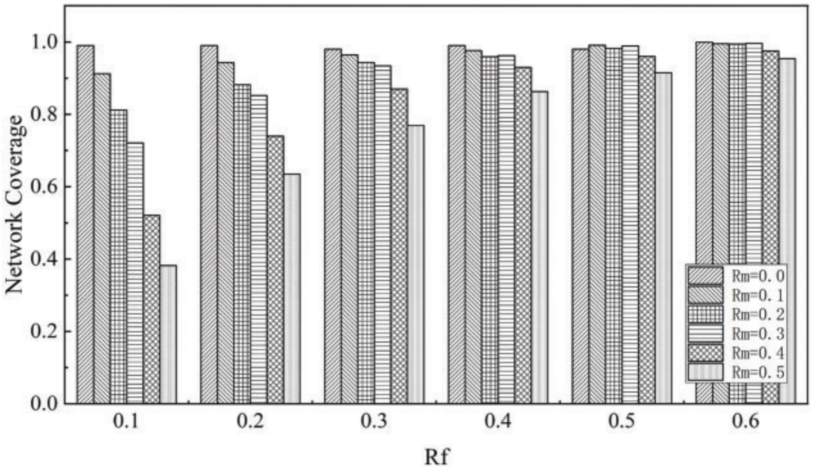

Figure 12: Network coverage under different levels of malfeasance

Routing Table Attack [27]. The working principle of a routing table attack involves predicting paths through the analysis of the collected network topology and relationships between nodes by attackers using predictive algorithms. As outlined in Section 3.5, in the FPSblo model, each seed node will generate multiple FPSblo sampled forwarding tables from a randomly generated starting node and periodically change the starting node according to the rules. Hence, messages generated by the source node can undergo more dispersed transmission with the assistance of the FPSblo model, forming multiple non-intersecting paths and regions. This makes it challenging for attackers to predict the route of message transmission.

5 Conclusion and Future Work

P2P networks are an essential component of blockchain systems, and their transmission performance directly affects the reliability of the blockchain system. This article proposes a blockchain p2p network transmission model named FPSblo, based on the farthest point sampling algorithm. The model addresses the problem of mismatch between upper-layer traffic and lower-layer physical link topology in traditional P2P networks. While ensuring high coverage, it reduces the redundancy of network transmission, improves load balancing, and enhances network transmission performance. However, the FPSblo model also slightly increases nodes' computational and storage costs. This study has only investigated the scalability of blockchain networks from the perspective of network topology optimization. It could be beneficial to explore comprehensive solutions to network scalability issues by incorporating advantageous technologies from blockchain systems, such as smart contracts, consensus mechanisms [28,29].

In future work, we will focus on a more profound analysis of scalability challenges in blockchain networks and discuss potential solutions. This may involve challenges related to more complex network topologies, the intricacies of smart contracts, and large-scale transaction processing. There are still some research gaps in the current exploration of the scalability of the underlying blockchain network. It is worth exploring network architectures that are better suited for blockchain systems, such as the content-addressable Named Data Networking [30] (NDN) architecture. This approach could potentially replace the current TCP/IP network paradigm entirely. However, there may be challenges in the adoption and deployment of such architectures.

Acknowledgement: Not applicable.

Funding Statement: This present research work was supported by the National Key R&D Program of China (No. 2021YFB2700800) and the GHfund B (No. 202302024490).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Longle Cheng, He Zhao; data collection: Xiru Li, Shiyu Fang; analysis and interpretation of results: Longle Cheng, Wansh Pan; draft manuscript preparation: Longle Cheng; project administration: Haibo Tan, Xiaofeng Li. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used to support the findings of this study are included within the article.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Deepa *et al.*, "A survey on blockchain for big data: Approaches, opportunities, and future directions," *Future Gener. Comput. Syst.*, vol. 131, pp. 209–226, 2022. doi: 10.1016/j.future.2022.01.017.

[2] H. Tan *et al.*, "Archival data protection and sharing method based on blockchain," *J. Softw.*, vol. 30, no. 9, pp. 2620–2635, 2019. doi: 10.13328/j.cnki.jos.005770.

[3] A. A. Monrat, O. Schelén and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117134–117151, 2019. doi: 10.1109/ACCESS.2019.2936094.

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf (accessed on 04/07/2010).

[5] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Proj. Yellow Paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[6] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Workshop Distrib. Cryptocurrencies Consens. Ledger.*, Chicago, IL, 2016, pp. 1–4.

[7] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "A survey on the scalability of blockchain systems," *IEEE Netw.*, vol. 33, no. 5, pp. 166–173, 2019. doi: 10.1109/MNET.001.1800290.

[8] Z. Sun, X. Zhang, F. Xiang and L. Chen, "Survey of storage scalability on blockchain," *J. Softw.*, vol. 32, no. 1, pp. 1–20, 2021.

[9] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16440–16455, 2020. doi: 10.1109/ACCESS.2020.2967218.

[10] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016. [Online]. Available: https://static1.squarespace.com/static/6148a75532281820459770d1/t/61af971f7ee2b432f1733aee/1638897446181/lightning-network-paper.pdf (accessed on 07/12/2021).

[11] J. Poon and V. Buterin, "Plasma: Scalable autonomous smart contracts," White paper, pp. 1–47, 2017. [Online]. Available: https://www.plasma.io/plasma-deprecated.pdf (accessed on 01/06/2020).

[12] H. Kalodner, S. Goldfeder, X. Chen, S. M. Weinberg, and E. W. Felten, "Arbitrum: Scalable, private smart contracts," in *27th USENIX Secur. Symp. (USENIX Secur. 18)*, 2018, pp. 1353–1370.

[13] J. Kwon and E. Buchman, "Cosmos whitepaper," *A Netw. Distrib. Ledger.*, pp. 27, 2019. [Online]. Available: https://wikibitimg.fx994.com/attach/2020/12/16623142020/WBE16623142020_55300.pdf (accessed on 19/06/2023).

[14] D. Ding, X. Jiang, J. Wang, H. Wang, X. Zhang and Y. Sun, "Txilm: Lossy block compression with salted short hashing," arXiv preprint arXiv:1906.06500, 2019.

[15] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *13th USENIX Symp. Netw. Syst. Des. Implement. (NSDI 16)*, 2016, pp. 45–59.

[16] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. 2016 ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 17–30.

[17] Y. Wu and J. Li, "The evolution process of blockchain p2p network protocol," *Comput. Appl. Res.*, vol. 36, no. 10, pp. 2881–2886, 2019.

[18] 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 Peer Netw. Appl.*, vol. 5, pp. 1–13, 2012. doi: 10.1007/s12083-011-0101-y.

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

[20] A. Liu, X. Du, N. Wang, and S. Li, "Research progress of blockchain technology and its application in information security," *J. Softw.*, vol. 29, no. 7, pp. 2092–2115, 2018. doi: 10.13328/j.cnki.jos.005589.

[21] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Adv. Neur. Inf. Process. Syst.*, vol. 30, 2017.

[22] G. Naumenko, G. Maxwell, P. Wuille, A. Fedorova, and I. Beschastnikh, "Bandwidth-efficient transaction relay for bitcoin," arXiv preprint arXiv:1905.10518, 2019.

[23] E. Rohrer and F. Tschorsch, "Kadcast: A structured approach to broadcast in blockchain networks," in *Proc. 1st ACM Conf. Adv. Financ. Technol.*, 2019, pp. 199–213.

[24] U. Klarman, S. Basu, A. Kuzmanovic and E. G. Sirer, "bloxroute: A scalable trustless blockchain distribution network whitepaper," 2018. [Online]. Available: https://bloxroute.com/wp-content/uploads/2019/11/bloXrouteWhitepaper.pdf (accessed on 31/08/2020).

[25] G. F. Riley and T. R. Henderson, "The *ns-3* network simulator," *Model. Tools Netw. Simul.*, pp. 15–34, 2010. doi: 10.1007/978-3-642-12331-3_2.

[26] G. Carl, G. Kesidis, R. R. Brooks, and S. Rai, "Denial of service attack detection techniques," *IEEE Internet Comput.*, vol. 10, no. 1, pp. 82–89, 2006. doi: 10.1109/MIC.2006.5.

[27] T. Baumeister, Y. Dong, G. Tian, and Z. Duan, "Using randomized routing to counter routing table insertion attack on freenet," in *2013 IEEE Glob. Commun. Conf. (GLOBECOM)*, IEEE, 2013, pp. 754–759.

[28] J. Liu, M. Xie, S. Chen, C. Ma, and Q. Gong, "An improved DPoS consensus mechanism in blockchain based on PLTS for the smart autonomous multi-robot system," *Inf. Sci.*, vol. 575, pp. 528–541, Oct. 1, 2021. doi: 10.1016/j.ins.2021.06.046.

[29] M. Xie, J. Liu, S. Chen, G. Xu, and M. Lin, "Primary node election based on probabilistic linguistic term set with confidence interval in the PBFT consensus mechanism for blockchain," *Complex Intell. Syst.*, vol. 9, no. 2, pp. 1507–1524, Apr. 2023. doi: 10.1007/s40747-022-00857-9.

[30] L. Zhang *et al.*, The NDN project team, "Named data networking (NDN) project," Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC, Palo Alto, CA, USA, 2010. [Online]. Available: https://named-data.net/techreport/TR001ndn-proj.pdf (accessed on 7/01/2013).