Towards a Trust-Enhanced Blockchain P2P Topology for Enabling Fast and Reliable Broadcast

Weifeng Hao, *Student Member, IEEE*, Jiajie Zeng, *Student Member, IEEE*, Xiaohai Dai, *Student Member, IEEE*, Jiang Xiao, *Member, IEEE*, Qiang-Sheng Hua, *Member, IEEE*, Hanhua Chen, *Member, IEEE*, Kuan-Ching Li, *Senior Member, IEEE*, and Hai Jin *Fellow, IEEE*

***Abstract*—Blockchain technology offers an intelligent amalgamation of distributed ledger, *Peer-to-Peer* (P2P), cryptography, and smart contracts to enable trustworthy applications without any third parties. Existing blockchain systems have successfully either resolved the scalability issue by advancing the distributed consensus protocols from the control plane, or complemented the security issue by updating the block structure and encryption algorithms from the data plane. Yet, we argue that the underlying P2P network plane remains as an important but unaddressed barrier for accelerating the overall blockchain system performance, which can be discussed from how fast and reliable the network is. In order to improve the blockchain network performance about enabling fast and reliable broadcast, we establish a trust-enhanced blockchain P2P topology which takes transmission rate and transmission reliability into consideration. Transmission rate reflects blockchain network speed to disseminate transactions and blocks, and transmission reliability reveals whether transmission rate changes drastically on unreliable network connection. This paper presents *BlockP2P-EP*, a novel trust-enhanced blockchain topology to accelerate transmission rate and meanwhile retain transmission reliability. BlockP2P-EP first operates the geographical proximity sensing clustering, which leverages K-Means algorithm for gathering proximity peer nodes into clusters. It follows by the hierarchical topological structure that ensures strong connectivity and small diameter based on node attribute classification. Then we propose establishing trust-enhanced network topology. On top of the trust-enhanced blockchain topology, BlockP2P-EP conducts the parallel spanning tree broadcast algorithm to enable fast data broadcast among nodes both *intra- and inter-* clusters. Finally, we adopt an effective node inactivation detection method to reduce network load. To verify the validity of BlockP2P-EP protocol, we carefully design and implement a blockchain network simulator. Evaluation results show that BlockP2P-EP can exhibit promising network performance in terms of transmission rate and transmission reliability compared to Bitcoin and Ethereum.**

***Index Terms*—Blockchain, peer-to-peer network, network clustering, trust-enhanced topology, broadcast algorithm.**

I. Introduction

TODAY blockchain technology has attracted increasing attention as the cornerstone of trust across a wide realm of society sectors from finance, industrial logistics to healthcare. Its beneficial characteristics including traceability, decentralization, and transparency spout out the massive proposals of blockchain systems and projects. Unfortunately, the real-world blockchain adoption experiences serious technical challenges that impede its further development, especially from the aspect of the overall system performance. State-of-the-art researches mainly focus on advancing the consensus algorithms in the consensus layer [1], as well optimizing the data storage in the data layer [2]. However, few studies have been conducted from the network layer (i.e., lying between the two layers) that can update the topology under consensus guarantee, while adapting the dynamic on-chain blockchain data traffic.

In particular, current blockchain P2P network performance is limited by two impact factors (i.e., transmission rate [3] and transmission reliability [4]). On the one hand, transmission rate reflects how fast blockchain P2P network can disseminate transactions and blocks for ensuring distribution and fairness. On the other hand, transmission reliability shows whether transmission rate changes drastically on unreliable network connection. In particular, the lack of consideration in transmission rate can not only lead to poor consensus performance, meanwhile bring about high risks of double spending attacks. According to the experiment insights, it normally takes on average 6 seconds to ensure a block is received by 50% of the total nodes in the Ethereum network, and up to 10 seconds, on average, for 90% of the nodes. Since the generation time of a new block is only 15 seconds in Ethereum, this network-level latency becomes a major barrier that limits the blockchain performance (i.e., *Transactions Per Second* (TPS)), leading to high potential of forks. Therefore, it is urgent to reduce blockchain network latency, so as to improve the overall performance of blockchain systems with stronger security. Besides this, transmission reliability can also affect transmission rate, because low transmission reliability aggravates the

Manuscript received August 7, 2019; revised December 20, 2019 and February 23, 2020; accepted February 27, 2020. Date of publication March 12, 2020; date of current version June 10, 2020. This work is supported by the Technology Innovation Project of Hubei Province of China under Grant No. 2019AEA171, National Key Research and Development Program of China under Grant No. 2018YFB1004800. The associate editor coordinating the review of this article and approving it for publication was R. Pasquini. *(Corresponding author: Jiang Xiao.)*

Weifeng Hao, Jiajie Zeng, Xiaohai Dai, Jiang Xiao, Qiang-Sheng Hua, Hanhua Chen, and Hai Jin are with the National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Laboratory, Huazhong University of Science and Technology, Wuhan 430074, China, and also with the Cluster and Grid Computing Laboratory, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: jiangxiao@hust.edu.cn; hjin@hust.edu.cn).

Kuan-Ching Li is with the Department of Computer Science and Information Engineering, Providence University, Taichung 43301, Taiwan (e-mail: kuancli@pu.edu.tw).

Digital Object Identifier 10.1109/TNSM.2020.2980303

blockchain P2P network burden. Due to the frequent packet loss of network, nodes will retransmit redundant data to destination, which could cause serious network congestion and obstruct other normal links.

In order to improve blockchain network performance, much prior work has been conducted to accelerate the transmission rate, according to the different topologies and transmission protocols. On one hand, the fully distributed unstructured topology of Bitcoin network is optimized by shortening the network diameter. However, it will bring in huge computation overheads because each node needs to repeatedly calculate the network distance between it and all the rest ones [5], [6]. On the other hand, Ethereum employs the fully distributed structured topology that increases the connectivity of the entire network, but the rapid growth of Ethereum nodes will introduce high maintenance cost of such structured topology. Many blockchain systems adopt gossip protocol [7], [8] to disseminate data with redundant transmission rounds. In general, blockchain network consists of two phases [9]. Apart from the blockchain network transmission rate, the network reliability becomes a leading barrier to enhance the network performance. More specifically, it requires to enable the 'trust-aware' property of blockchain topology in two folds. First, trust value [4], [10] between nodes is introduced to facilitate reliable connection. Second, detection of inactive nodes should be incorporated during data transmission. Nevertheless, recent work still lacks of concerns about taking the above two influential factors simultaneously for performance enhancement.

In this paper, we propose a novel network protocol namely BlockP2P-EP. Blockchain network workflow can be divided into two phases. In the first phase, BlockP2P-EP achieves the goal of constructing the trust-enhanced P2P topology. In the second phase, BlockP2P-EP takes optimized broadcast algorithm while considering inactive node detection. Combining with the two network phases, five steps are included in the BlockP2P-EP protocol. First, BlockP2P-EP gathers the proximity peer nodes into clusters based on the K-Means algorithm. We then optimize the inter-cluster topology by organizing the nodes into a Harary-like graph with high connectivity and small diameter. In the meanwhile, we promote and maintain the node connection based on the trust value. Then a parallel spanning tree broadcast algorithm is designed to speed up the data broadcast, by eliminating the multiple rounds of message in a single communication process. Finally, BlockP2P-EP applies effective node inactivation detection mechanism into the second network phase. To facilitate the evaluation of performance in the large-scale network, we design and implement *BlockSim*, a simulator to simulate the running of blockchain network without affecting the accuracy of the evaluation results. With the help of BlockSim, we conduct several experiments to compare BlockP2P-EP protocol with the counterpart protocols in Bitcoin and Ethereum. The experimental results show that BlockP2P-EP protocol can effectively reduce blockchain network latency (i.e., transmission rate) and keep transmission reliability. In summary, this paper makes the following novel contributions:

- To the best of our knowledge, this is the first in-depth analysis of influential factors of blockchain performance about transmission rate and transmission reliability from underlying P2P network.
- We delve into the five key factors that limit the blockchain network performance from how fast and reliable network is.
- We introduce an optimized blockchain network protocol *BlockP2P-EP* to improve blockchain transmission rate and maintain transmission reliability.
- To verify the feasibility and efficiency of BlockP2P-EP protocol, we design and implement *BlockSim*, a blockchain simulator tailored for large-scale blockchain network.
- Experimental results demonstrate that BlockP2P-EP can effectively improve transmission rate from three different aspects compared to Bitcoin and Ethereum. In the meanwhile, BlockP2P-EP can also maintain transmission reliability from the experiment insights.

This paper is an extension of the paper "BlockP2P: Enabling Fast Blockchain Broadcast with Scalable Peer-to-Peer Network Topology [9]", which provides a more enhanced protocol for blockchain P2P network. Specifically, this paper considers more complex blockchain P2P network environments. Compared with BlockP2P, BlockP2P-EP not only takes into consideration the transmission rate, but also enhances transmission reliability, which shows whether the transmission rate changes drastically on unreliable network connections. In the meantime, BlockP2P-EP is composed of five stages based on the blockchain two-phase workflow, in comparison, BlockP2P consists of 3 stages. In order to verify the new blockchain network protocol, this paper adds extensive experiments to evaluate the transmission reliability including success rate of malicious nodes detection and load rate of inactive node detection based on original experiments. Results demonstrate that BlockP2P-EP can provide more reliable broadcast compared with the BlockP2P protocol. The rest of the paper is organized as follows. Section II presents the background knowledge about the network protocol of current blockchain systems. Related work on network optimization is stated in Section III. Section IV elaborates the design of BlockP2P-EP in five steps. Extensive experiments are conducted in Section V to evaluate the system performance in terms of reducing network latency and maintaining transmission reliability. Finally, we conclude the paper in Section VI.

II. Background

As mentioned in Section I, blockchain performance can be discussed from transmission rate and transmission reliability. Different phases in blockchain network have different impact parameters to ensure transmission rate and transmission reliability, and specific content is as follows.

A. Description of Transmission Rate and Transmission Reliability

P2P network enables direct information interaction between different nodes in the blockchain, so network performance seriously affects overall system running. Blockchain P2P network performance includes two dimensions (i.e.,

transmission rate and transmission reliability), and both of them have their impact parameters.

Transmission rate reflects the transmission speed of blockchain network in good condition, limited by network diameter, network connectivity and transmission rounds. Network diameter refers to the longest distance between any two nodes in the network, which is generally measured by the link tree. Generally, improving the transmission speed of blockchain network, means the network diameter will decrease. Network connectivity describes the extensive process of connecting various parts of a network to one another, for example, through the use of routers, switches and gateways, and how that process works. Transmission rounds directly equal to the frequency of blockchain network transmission, which is determined by the broadcast protocol.

Transmission reliability shows the ability and time change of maintaining transmission in case of blockchain network error, restricted by network trust connection and node inactivation detection. Network trust connection means any node should consider the possible problems of the evil data transmission (i.e., transaction and block) when selecting neighbors, so as to avoid the evil data generated from malicious nodes in network layer and reduce the algorithm complexity of the upper consensus protocol. Node inactivation detection represents the connection stability during network transmission. For example, since the connection status among nodes is relatively stable and the time taken to establish connection is usually very short, the most important component of the total network transmission rate is the broadcast latency in the phase of data transmission. However, configurations of both phases can have effects on the transmission rate.

B. Two Phases of Blockchain Network Workflow

As shown in Figure 1, the process of information interaction between two nodes can be divided into two phases: connection establishment marked by gray circles, and data transmission marked by red circles. To ensure transmission rate and transmission reliability discussed in the previous section, both of the two phases develop their own design about the five key impact parameters.

In the phase of connection establishment, different network topologies may be formed among the nodes. Different network topologies will have different effects on the broadcast latency, leading to high transmission rates, which can be measured by network connectivity and network diameter [11]. The network connectivity refers to the number of neighbor nodes connected to each node in the network. The larger the network connectivity is, the more neighbors a node can broadcast the data each time. In this way, the overall time spent on the network broadcasting can be reduced. The smaller the network diameter, the shorter the average broadcast time between any two nodes, thus accelerating the overall broadcast time across the network. As a result, optimizing the network topologies of nodes including the network connectivity and diameter can effectively reduce the broadcast latency. On the other hand, lack of mutual trust between nodes easily leads malicious nodes to transmit fake data, which in turn affects blockchain P2P network

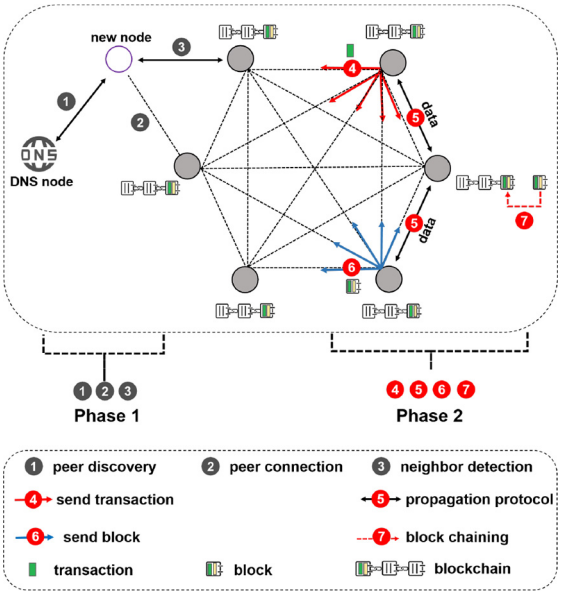


Fig. 1. Workflow of blockchain P2P network.

performance. So we come up with the method for evaluating the trust value in the phase of connection establishment.

In the phase of data transmission, the Gossip algorithm is used to broadcast the data from a node to its neighbors [12]. In distributed systems, Gossip is a common synchronization algorithm, mainly composed of time model and message update model. According to the time model, Gossip can be divided into synchronous Gossip and asynchronous Gossip. According to the message update model, Gossip can be divided into unicast-based Gossip and broadcast-based Gossip. For blockchain P2P network (e.g., Bitcoin and Ethereum), the Gossip algorithm usually adopts asynchronous and broadcast-based methods. In each interval, one node will be awakened to randomly select neighbors for data exchange. During the process of broadcast, a node firstly selects the nodes from its neighbors to disseminate the data using the propagation protocol. The node which receives the data repeats the process above until all the nodes in the network have received the data. More specifically, the propagation protocol [3], [13] used by Gossip algorithm further includes three steps. First, a node (i.e., sender) sends an *INV* message to its neighbor node before sending one piece of data (namely a transaction or a block in the context of blockchain). Second, the neighbor node determines whether it has received the data before. If not, it returns a *getdata* message back to the sender; otherwise, it ignores the *INV* message. Finally, before the end of timeout set by the sender, if the sender receives the *getdata* message, it sends the piece of data to the neighbor. It should be noted that a node only broadcasts data to its directly connected neighbors. The broadcast process will run in many rounds by each node, until each node in the network has received the data. As a result, Gossip protocol may lead to large data broadcast latency due to many rounds of broadcast. Besides, three steps of the propagation protocol bring extra communication rounds, which exacerbate the problem of large broadcast latency. In the meanwhile, for most blockchain systems, e.g.,

Bitcoin and Ethereum, they simply rely on the heartbeat mechanism to detect whether node is survival. So the most direct embodiment of transmission reliability in the phase of data transmission is the node inactivation detection.

III. Related Work

In blockchain P2P network, information interaction between two nodes consists of two phases: the connection establishment and the data transmission. We will present the existing optimization work in these two phases.

A. Connection Establishment

In the stage of connection establishment, optimization works mainly contains topology construction and trust connection.

Network diameter and network connectivity: The information propagation delay reveals the transmission rate of blockchain systems [1], since the high latency increases the time for all the nodes to reach a consensus. The high network latency makes the system more vulnerable to malicious attacks [14]. The topological structure of the mainstream blockchain systems can be divided into two categories: one is the unstructured topology in Bitcoin [15], the other is the structured topology in Ethereum [16] (i.e., Kademlia) and NKN [17] (i.e., Chord). To measure the quality of network topology, two metrics including network diameter and connectivity are adopted. Nodes in unstructured topology are randomly connected, which results in a large network diameter. In order to decrease the network diameter, BCBPT protocol utilizes the proximity clustering algorithm based on the number of network hops between nodes, and then connects the nodes that are physically proximal [18]. However, BCBPT brings in great algorithm complexity, as each node needs to calculate the network hops to all other nodes. Croman *et al.* revealed that Bitcoin cannot fully utilize the bandwidth in the network, which has serious impact on transactions processing. Then they proposed to reduce the network latency of blockchain by starting with optimizing the network topology [19]. Compared with the unstructured topology, the structured topology has a good network connectivity. But its network diameter is also very large, since the network latency between nodes is not taken into consideration when they try to establish a connection. Moreover, creating the structured network topology brings in huge computation cost, because of the large size of blockchain network (e.g., the size of nodes in Ethereum has almost reached to $10k$). The cost will increase significantly as the network size further increases, the same with the network latency.

Network trust connection: As stated in the previous section, transmission reliability is composed of two key factors (i.e., network trust connection and node inactivation detection). In this phase, transmission reliability specifically refers to network trust connection. Xiong and Liu propose PeerTrust framework to quantify the trustworthiness of peers in the P2P online communities, which can minimize threats in the network [10]. This model can effectively identify malicious nodes and decrease inauthentic files in the network, but it brings high communication cost. Only few work formulated the trust value in the blockchain network, since blockchain technology adds many new characteristics to the node compared with the traditional P2P network. The IFT protocol [4] aimed to select high credible nodes as their neighbors to improve the communication efficiency of the blockchain system. However, it calculates the trust value based on the nodes behavior in the application layer without considering the network features of nodes in the network layer. Network trust will be affected at different network layers (i.e., network, transport, application) and even cross layer. Compared with the IFT protocol, BlockP2P-EP adopts more effective trust value calculation method from P2P network layer, while can reduce malicious data dissemination.

B. Data Transmission

In the stage of connection establishment, optimization work contains transmission optimization and node inactivation detection.

Transmission rounds: The blockchain network is the broadcast channel for data. Some efficient broadcast protocols [7], [8] are proposed to speed up the progress of broadcast. Bitcoin employs the flood-based [20] algorithm to broadcast the data, while Ethereum adopts the gossip-based [21] broadcast algorithm. Both of these two algorithms bring huge redundant data in broadcast, because the data will be sent multiple rounds before it meets the termination conditions of the broadcast. Besides, the multi-message transfer in the propagation protocol greatly lowers the speed of data broadcast [13], [22]. An attempt to solve the problem above is conducted by Decker and Wattenhofer [3], which tries to optimize the Bitcoin network by removing the process of verification and pipelining the process of block propagation. However, their ideas are only at the conceptual stage and further experiments are needed to prove it.

Node inactivation detection: Many blockchain systems take heartbeat mechanism [11] to detect whether a node is alive when network failure happens. However, heartbeat mechanism cannot ensure real-time performance of node inactivation detection. The heartbeat we mentioned here refers to the keep-alive mechanism at the application layer. Heartbeat at the application layer requires additional development stages, and users need to build complex application layer logic to keep the blockchain P2P network running. As a result, redundant development process at the application layer causes a large amount of network traffic consumption, which affects the network's real time performance.

In order to improve blockchain P2P network performance (i.e., transmission rate and transmission reliability), BlockP2P-EP starts from the two phases (i.e., connection establishment and data transmission) to ameliorate the performance problems. First, BlockP2P-EP constructs a hierarchical structured topology after the node clustering. Compared with the unstructured topology in Bitcoin [15], structured topology in Ethereum [16] (i.e., Kademlia) and NKN [17] (i.e., Chord), BlockP2P-EP has smaller network diameter with strong network connectivity. Second, BlockP2P-EP calculates the trust value. Compared with the IFT protocol [4], BlockP2P-EP adopts more effective trust value calculation method from

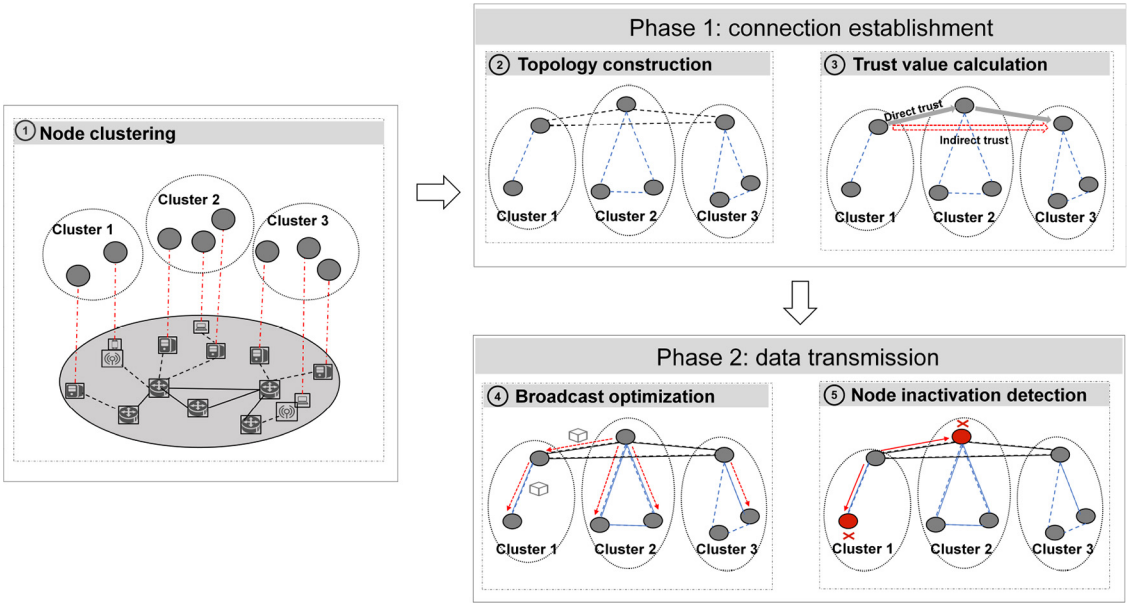


Fig. 2. Overview of the BlockP2P-EP.

P2P network layer to reduce malicious data dissemination. Third, BlockP2P-EP optimizes the broadcast algorithm, which can effectively reduce transmission rounds compared with the flood-based [20] algorithm. Finally, BlockP2P-EP performs node inactivation detection to reduce the communication overhead brought by the heartbeat mechanism [11].

IV. Design

Figure 2 gives an overview of how the BlockP2P-EP protocol operates, which is composed of five parts: node clustering, topology construction, trust value calculation, broadcast optimization, and node inactivation detection. First, to reduce the complexity of building the network topology for the whole network and ensure parallel broadcast between clusters, a *Geographical Proximity Sensing Clustering* (GPSC) method based on the K-Means algorithm [23] is devised. Second, a *Structured Hierarchical Network Topology* (SHNT) approach is proposed to construct the topology of node connection with a high network connectivity and a small diameter. Third, we propose a *Distributed Feedback Trust Value* (DFTV) method to establish and maintain strong connection based on the trust value calculated by every node. Fourth, we design a *Parallel Spanning Tree Broadcast* (PSTB) mechanism to parallelize the broadcast processes in both intra-cluster and inter-cluster nodes. Last, a *Subscribed Node Inactivation Detection* (SNID) algorithm is adopted to ensure network reliability.

A. Node Clustering

To guarantee proximal and coequal clustering, BlockP2P-EP implements the GPSC method to organize the nodes across the network into several clusters, based on the well-known K-Means algorithm. The average number of nodes in a cluster is the key parameter in the K-Means algorithm that requires careful design. On one hand, such number can not be set too large, otherwise, it will bring in high communication latency between two intra-cluster nodes. On the other hand, a small value may increase the communication cost between two inter-cluster nodes, since it enlarges the number of clusters. According to the previous studies in [11] and [24], the optimal setting of the number of nodes in a cluster should be *logN*. After the number of nodes in a cluster is set, GPSC organizes all the nodes into several clusters in three stages as depicted in Figure 3.

1) Selection of Cluster Centers: First, we describe how GPSC selects the nodes as cluster centers. One simple way is to perform iterative computation of K-Means algorithm continuously. However, it brings in huge computation costs since it requires each node to measure the network latency between it and all the other nodes, whose computation costs are too high. To reduce the computational complexity, BlockP2P-EP pre-designates the candidate subsets based on distribution density of the blockchain nodes, so the network distance is equal to the network latency. GPSC creates a candidate subset for selecting cluster centers in advance. In particular, with network latency as the Euclidean distance between two nodes, GPSC selects the cluster centers in three steps as follows:

- **Step 1:** Calculate the Euclidean distance $T(n_i, n_j)$ $(i \neq j)$ between any two nodes in the candidate subset. Find the nodes pair with the furthest distance to form a new set $S_m (1 \leq m \leq K)$, where K represents the number of network clusters, and then delete the two nodes from the candidate subset;
- **Step 2:** Add the node which is furthest from the new set to update S_m, and then remove the node from the original candidate set;
- **Step 3:** If the number of nodes in S_m is smaller than K, repeat Step 2; otherwise, the nodes in the set S_m are taken as the cluster centers.

2) Choice of Aggregation Nodes: To ensure all the cluster centers are evenly distributed, assistant nodes named *aggregation nodes* C_{aggre} are chosen, each of which is located at the

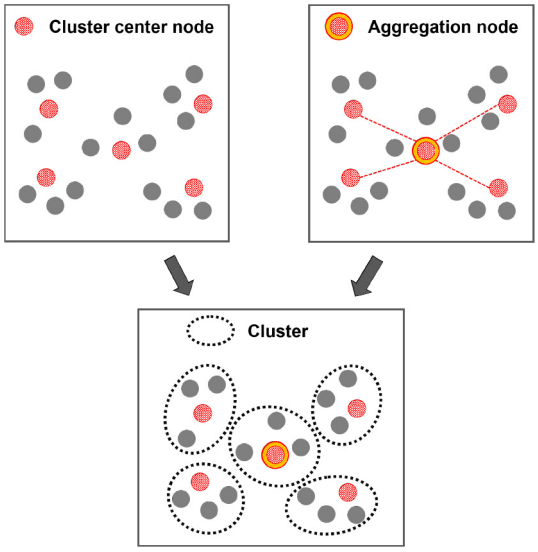


Fig. 3. Three stages of node clustering.

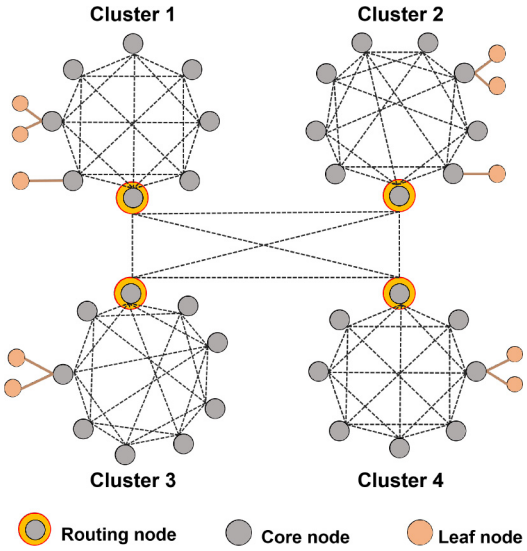


Fig. 4. Structured hierarchical network topology.

geometric center of a cluster. It is difficult to calculate C_{aggre} only according to the network latency between two nodes. Therefore, GPSC adopts the method of *network coordinate system* (NCS) to figure out C_{aggre} [25]. First, GPSC sets a network coordinate for each cluster center node according to Equation (1).

$$F\left(H_{S_1},\ldots,H_{S_k}\right)=\sum_{S_i,S_j\in\{S_1,\ldots,S_k\},i>j}\varepsilon\left(d_{S_iS_j},\bar{d}_{S_iS_j}\right) \quad (1)$$

where S_i and H represent the cluster centers and the network coordinates of the cluster centers respectively, d and $\bar{d}$ represent the network distances between two nodes in the actual system and network coordinate system separately, and ε represents the error function. After getting the geometric center coordinates $\bar{C}_{aggre}$, GPSC chooses the cluster aggregation node according to Equation (2).

$$\varphi\left(C_{aggre}\right)=\bar{D_{min}}\left(H_{S_i},\bar{C}_{aggre}\right),\quad S_i\in\{S_1,\ldots,S_k\} \quad (2)$$

where φ represents the matching function of the cluster aggregation node, $\bar{D_{min}}$ represents the minimum network distance between two nodes in NCS, and $\bar{C}_{aggre}$ represents the geometric center coordinates.

3) Network Clustering: Relying on the above prerequisites, GPSC finally clusters all the nodes according to an objective function $D(X_i,S_j)$ in Equation (3).

$$D\left(X_i,S_j\right)=\omega_1\times d_1\left(X_i,S_j\right)+\omega_2\times d_2\left(S_j,C_{aggre}\right),\quad \omega_1+\omega_2=1 \quad (3)$$

where X_i and S_j represent the general node and the center node respectively, d_1 represents the distance between a node and a center while d_2 represents the distance between a center and an aggregation node. Besides, ω_1 and ω_2 are two weight factors. Compared to the $O(N^2)$ complexity of BCBPT [18], GPSC can decrease algorithm complexity to $O(K\cdot N)$, which enables the fast re-clustering of nodes in response to the possible network change, thus promoting the system's robustness.

B. Topology Construction

The execution of the GPSC algorithm could result in hundreds of nodes in a cluster. In this way, a node has to select a small subset from the cluster to constitute its neighbors, thus constructing the network topology. As previously mentioned in Section II, network connectivity and cluster diameter can have significant effects on the blockchain broadcast performance. To enable each cluster to have an optimal network connectivity and diameter, we introduce the SHNT approach to construct the network topology as shown in Figure 4. More precisely, SHNT consists of network initialization and maintenance processes.

1) Network Initialization: The nodes can be divided into SPV nodes and full nodes according to their roles in the blockchain network. SPV refers to a simplified payment verification method that can quickly and securely verify payments even without the complete transaction record in blockchain P2P network. Compared with the full nodes, SPV nodes only need to download all block headers to verify the payment, so the data broadcast by SPV nodes is much less and lighter. Due to their different behaviours in the network, SHNT regards the SPV nodes and full nodes as leaf nodes and core nodes respectively. Besides, SHNT selects one core node from each cluster as the routing node, according to the node ID randomly, which ensures the security and randomness. Routing nodes allow the data transmitted from one cluster to another. Once a piece of data is transmitted from one routing node to another, the data can concurrently broadcast in these two clusters, thus speeding up the data transmission across the whole network. The detailed description of different nodes are listed as follows.

- **Leaf node:** consisting of SPV nodes, periodically sending node information to the core node and initiating a transaction.
- **Core node:** consisting of mining nodes, maintaining and managing leaf nodes of the cluster which they are located at, and forwarding transactions or blocks among nodes in the cluster.

- **Routing node:** selected from core nodes, storing routing node information about other clusters, and forwarding transactions or blocks among nodes in the cluster.

Based on the classification of blockchain network nodes, the construction process of blockchain network topology by SHNT can be divided into three steps. First, a leaf node is connected directly to a core node that is closest to it. Second, we construct the topology among the core nodes as a Harary-like graph, which has high connectivity and small diameter [11]. Third, a node from the core nodes is selected randomly as the routing node. Once being selected as the routing node, it will contain all the network information of other clusters. If the routing node breaks down, one of the other core nodes will replace it.

2) Network Maintenance: After initializing the network topology in Section IV-B1, a natural and important concern is how to maintain the stability of network topology, especially when facing the dynamic changes, i.e., the join of new nodes and leave of old nodes. Recall that the average number of nodes in a cluster is of great importance. It will incur huge communication overhead, no matter the number of nodes is set too large or too small. To overcome the challenges brought by dynamic network, we adopt an automatic adjustment mechanism that can keep the size of each cluster stable within $O(\log N)$. In particular, the mechanism merges small clusters when many nodes churn to leave, and splits large clusters if a large number of nodes join in the same clusters. The minimal threshold to trigger *cluster merge* and maximal threshold to trigger *cluster split* are set to $\frac{\log N}{l}$ and $l \times \log N$ (l represents the cluster adjustment parameter, and N represents the number of nodes in the entire network), respectively.

C. Trust Value Calculation

BlockP2P-EP uses distributed P2P trust model to motivate node transmission, and enhance network reliability through the reward mechanism. In BlockP2P-EP protocol, frequency of successful blockchain data transmission (i.e., transaction) between any two nodes is used to reflect the trust relationship. Based on this influence factor, BlockP2P-EP sets up two different trust models: Intra-cluster trust model and Inter-cluster trust model, and the detailed calculations are listed as follows.

1) Intra-Cluster Trust Model: Trust value in intra-cluster model consists of two parts: direct trust value DTr_{ab} and indirect trust value $InDTr_{ab}$ between any two nodes. First, direct trust value between node a and node b is defined in Equation (4):

$$DTr_{ab} = \begin{cases} \frac{S_{ab} - US_{ab}}{S_{ab} + US_{ab}} & S_{ab} + US_{ab} \neq 0 \\ 0 & S_{ab} + US_{ab} = 0 \end{cases} \tag{4}$$

In Equation (4), DTr_{ab} represents the detailed quantitative trust degree between node a and node b, S_{ab} indicates the number of successful data transmission, US_{ab} indicates the number of unsuccessful data transmission times. BlockP2P-EP stipulates that the direct trust value will be set to 0 if there is no interaction history between the two nodes.

While the direct trust value cannot fully evaluate the honesty of nodes with only one neighbor's subjective evaluation, so BlockP2P-EP introduces the indirect trust value $InDTr_{ab}$ in Equation (5). Indirect trust value can indicate the long-term historical behavior of the nodes and reflect the trust value more objectively and fairly.

$$InDTr_{ab} = \sum_{m \in A(b)} DTr_{am} DTr_{mb} \tag{5}$$

$InDTr_{ab}$ is the indirect trust value of the node b calculated by node a with plenty of recommendation information. DTr_{mb} represents the local direct trust value between the recommended node m and node b in one network cluster. DTr_{am} indicates the direct trust value between node a and the recommended node m. $A(b)$ is the recommended nodes' set of node b. As can be seen from Equation (6), node a gives higher weight to local evaluations from nodes with high reliability.

In the end, global trust value in intra-cluster model can be calculated in Equation (6):

$$R_{ab} = \alpha \times DTr_{ab} + (1 - \alpha) InDTr_{ab} \tag{6}$$

α is the confidence factor of direct trust value, and its value is limited by the number of interaction times between node a and node b. The more interaction times, the larger the value of α.

2) Inter-Cluster Trust Model: Routing node carries the inter-cluster information, so BlockP2P-EP takes the trust degree between clusters into trust value calculation for routing node. First, BlockP2P-EP defines the global trust value between any other nodes and the routing node in Equation (7):

$$R_{SP_i} = \sum_{p,q \in I(G_i), p \neq q} \left(DTr_{pq} DTr_{qSP_i} \right) \tag{7}$$

R_{SP_i} indicates the global trust value of the routing node SP_i. $I(G_i)$ is the node set in which the SP_i is located. At a certain moment, the trust of the super node is unique for the entire group, which is not influenced by other nodes.

Then BlockP2P-EP defines the global trust value between two different inter-cluster routing nodes in Equation (8).

$$R_{SP_i SP_j} = \begin{cases} R_{SP_i} R_{SP_j} \times \frac{S_{G_i G_j} - US_{G_i G_j}}{S_{G_i G_j} + US_{G_i G_j}} & S_{G_i G_j} + US_{G_i G_j} \neq 0 \\ \tau & S_{G_i G_j} + US_{G_i G_j} = 0 \end{cases} \tag{8}$$

R_{SP_i} and R_{SP_j} are the global trust value of the routing node SP_i as well SP_j correspondingly. $S_{G_i G_j}$ is the number of successful data transmission times between nodes from group G_i and nodes from group G_j. τ represents the global trust value, when some nodes become the routing nodes in the initialization phase.

In order to calculate the trust value between nodes, one node initiates multiple query requests to obtain feedback information from another node within the $O(\log M)$ hops (M is the size of network cluster), when these two nodes belong to one cluster. If these two nodes are located in different clusters, node i needs $O(2 \log M) + O(\log N/M)$ hops, where N is the size of the entire P2P network. Therefore, BlockP2P-EP has good scalability with low communication overhead.

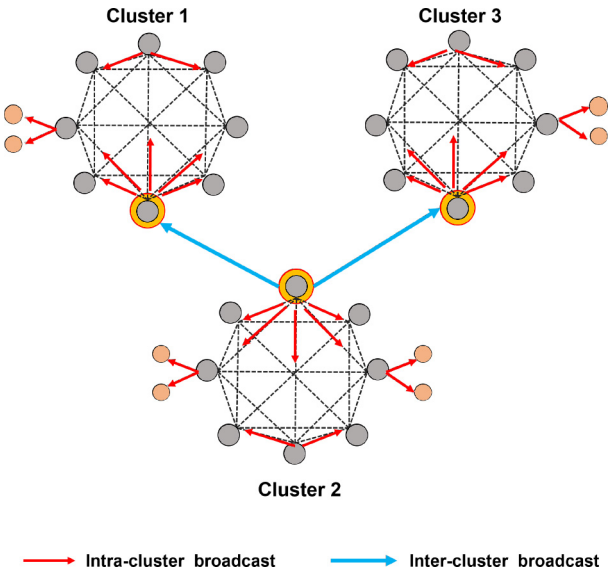


Fig. 5. Parallel spanning tree broadcast.

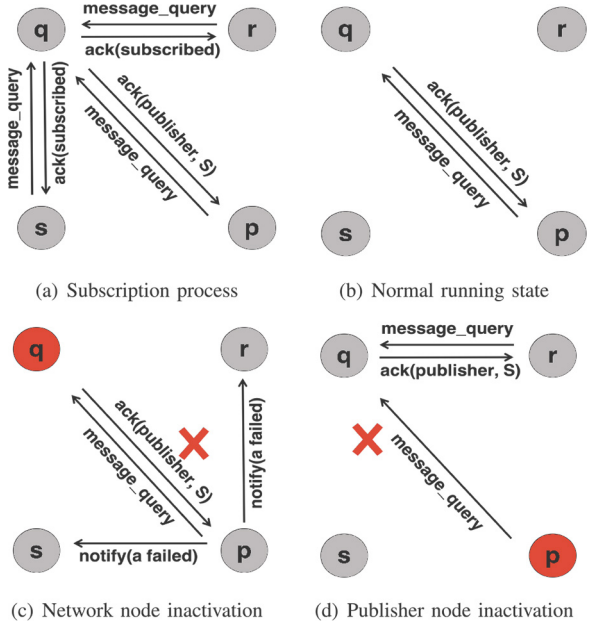


Fig. 6. Node inactivation detection algorithm.

D. Broadcast Optimization

In order to improve the broadcast efficiency, a *Parallel Spanning Tree Broadcast* (PSTB) method is adopted. As shown in Figure 5, the data will be sent by one routing node to the rest according to the route table. Once a routing node receives the data from other clusters, it will send the data to the nodes in the same cluster along a spanning tree.

1) Inter-Cluster Broadcast: The inter-cluster broadcast of data is based on the routing nodes of different clusters. Each routing node stores a route table, which records the information of all the other routing nodes. As shown in Figure 5, once the routing node receives the data, it will not only broadcast in its cluster, but also forward the data to other routing nodes as well, according to the route table. Hence, this method can enable parallel and fast data broadcast. In order to reduce the security problems caused by the crash of routing nodes, PSTB proposes a backup mechanism for routing table, which randomly selects a node from the core nodes as the backup node.

2) Intra-Cluster Broadcast: Only the routing node in a cluster will broadcast the data, which effectively avoids the huge network overheads brought by Gossip protocol [21]. If the broadcast source in the cluster is not a routing node, the source node will firstly send the data to the routing node. Once the routing node receives the data, it will broadcast the data in the cluster along the spanning tree. As for a spanning tree table, PSTB adopts the center-based approach to build it. First, the protocol selects a central node, and then all other nodes propagate the *INV* message to the central node to join the tree directly. In order to deal with the interference caused by the dynamic network changes, each routing node will update the spanning tree table periodically to enable the timeliness of the algorithm.

E. Node Inactivation Detection

Node inactivation under the blockchain P2P network is inevitable with at least one inactivated node every minute according to the statistical results [6]. In the meanwhile, node inactivation can affect the normal network transmission process, thereby reducing blockchain performance and affecting network reliability. Traditional node inactivation detection algorithms [4], [6] mainly focus on the heartbeat detection periodically, which results in many problems. On the one hand, sudden departure of nodes caused by high perturbation makes the heartbeat detection very frequent. On the other hand, every node needs to maintain a large number of connections (i.e., $O(\log N)$), so the failure detection overhead will reach $O(N \log N)$, which greatly affects the scalability of the entire Blockchain P2P network. In order to reduce network communication overhead and maintain node inactivation detection, BlockP2P-EP proposes a lightweight neighbor detection algorithm.

BlockP2P-EP adopts the PULL as the basic failure detection strategy, that is, detection node p first sends a query message *message_query* (*mq*) to the detected node q, and then q returns the response message (ack) to p in normal working state. Each node x only detects the neighbor node from the set $IV(x)$ recorded in its own routing table. After receiving *mq*, there are two types of response messages *ack*. As shown in Figure 6(a), the node q returns the message *ack* (*publisher*, *s*) to the *mq* originally received, and assigns the *mq* issuer as the publisher of the detection result (node p in Figure 6(b)). The detected node adds the nodes that send the detection message afterwards to the subscriber set *s*, and passes the response message *ack* (*subscribed*) to notify them that they are in subscription state (nodes r and s in Figure 6(b)). When no failure occurs in the blockchain network, nodes in the subscription state will no longer send any detection message. Only the publisher will continue to send the *mq* in clock cycle Δ, and then the detected node will aggregate the subscribers that are constantly updated and attached to the *ack* (*publisher*, *s*) message as well as update the the subscriber set *s* saved by p.

It can be seen in Figure 6(c) that when the node P does not receive the *mq* from q within the Δ time, it can be concluded that the node q has failed. After that, p will modify the detection result, and the subscriber set will be initiated to notify all the detectors of node q. Figure 6(d) shows the failure condition of publisher node. A two-way detection mechanism is adopted between the publisher node and the detected node, by which the node q starts a timer after issuing the response message *ack* (*publisher*, *S*). Since the interval of the detection message from the publisher P is known to be Δ, for this reason, when the new detection message *mq* is not received, then node q will find that the publisher has failed. At this time, node q selects a node r with a long online time from the set S, and assigns it as a new issuer by sending a message *ack* (*publisher*, *S*). What's more, publisher node is the reliability bottleneck of the node inactivation algorithm. To this end, BlockP2P-EP sets backup nodes of publisher to ensure that the publisher node has a longer online time than the detected one.

V. Evaluation

In this section, we conduct several experiments to evaluate BlockP2P-EP. First of all, we introduce the experimental setup, including the platform configuration, implementation, and the evaluation metrics. Then, we analyze the experimental results from comprehensive perspectives, and compare BlockP2P-EP with Bitcoin and Ethereum.

A. Experimental Setup

1) Platform Configuration: We conduct our experiments on two machines, each of which has two eight-core Intel Xeon E5-2670 2.60Hz CPUs, 64GB DRAM, 300GB HDD, and InfiniBand network card, with CentOS 7.0 as the operating system.

2) Implementation: Blockchain P2P network can be of a very large scale with millions of dynamically changing nodes, which typically join or leave stochastically. It is not feasible to evaluate a new protocol in a real environment, especially in its early stages. Optionally, without sacrificing the accuracy of experimental results, the simulation methods are adopted. In this paper, we design a generic blockchain network simulator named BlockSim based on PeerSim [26]. BlockSim has been developed with extreme scalability and support for network dynamically change in mind. It is composed of the event-driven engine, which is supported by many simple, extendable, and pluggable components (i.e., *simulation-network*, *simulation-consensus*, and *simulation-data*). BlockSim supports structured and unstructured Blockchain P2P network simulation. To simulate different network environments in the blockchain, developers can implement the interfaces provided by BlockSim as needed, which include topological connection, latency setting, network broadcast algorithm and so on.

Now we will introduce the specific implementation of the three blockchain network protocols (i.e., Bitcoin, Ethereum, and BlockP2P-EP) and evaluate their performance based on BlockSim. First, the simulation-network part contains three interfaces, including protocol, transport, and linkable. Protocol interface can simulate the underlying network-level protocol to define the communication methods between any two nodes. Transport interface simulates transport-level communication in overlay networks. Linkable interface defines the connection between any two nodes, by which one node records their neighbors to form different topologies. Of course, users can generate a fixed network topology by calling the graph model interface (random and small-world models, etc.). Furthermore, the network latency between nodes can be set through a configuration file to simulate a real network latency environment. Second, the simulation-consensus part defines the consensus mechanism of the blockchain network. Users can set different parameters to choose whether to use Bitcoin's PoW mining protocol or Ethereum's Ghost protocol (an improvement of PoW) to determine different outcomes about the block mechanism and spacing. Third, in terms of simulation-data, common blockchain data structures can be customized, such as transactions and blocks. The speed of transaction and block generation depends on the specific blockchain system. For example, in the Bitcoin network, only one block is generated every 10 minutes, which can be set through the configuration interface exposed by BlockSim.

3) Evaluation Metrics: In order to observe the network optimization effect of BlockP2P-EP compared to Bitcoin and Ethereum, we establish the evaluation metrics about blockchain network performance from five aspects:

- **General performance:** static performance with the number of nodes fixed;
- **Network scalability:** dynamic performance with the number of nodes changing;
- **Network stability:** stable performance with the number of nodes joining and leaving;
- **Success rate of malicious nodes detection:** the ability of network to detect malicious nodes and the rate of successful transmission of transactions, while in the presence of different malicious nodes and time cycles;
- **Load rate of inactive node detection:** network load generated by detection of inactive node under different network size and time cycles.

Network transmission rate is defined as the transmission speed of blockchain network, which mainly includes general performance, network scalability, and network stability. First, general performance means how much the broadcast time of transactions and blocks consumes with the fixed nodes, when different synchronization ratios are reached. Here we consider the blockchain network scale in reality to find a reasonable maximum network size. In the real blockchain network, Bitcoin has 10,561 nodes [27], and Ethereum currently has 8,485 nodes [28]. Therefore, the maximum blockchain network size is fixed as 14,000 to fit in with the actual blockchain network size. Second, network scalability means how the broadcast time changes when the number of network nodes increase from 2,000 to 14,000, with the fixed synchronization ratio. In the case of increasing network size, different speed of the synchronization time change can reflect the blockchain system scalability. In the end, we evaluate the robustness of BlockP2P-EP, by investigating the fluctuation of the time when lots of nodes join or leave in one data synchronization process for the evaluation of network stability.

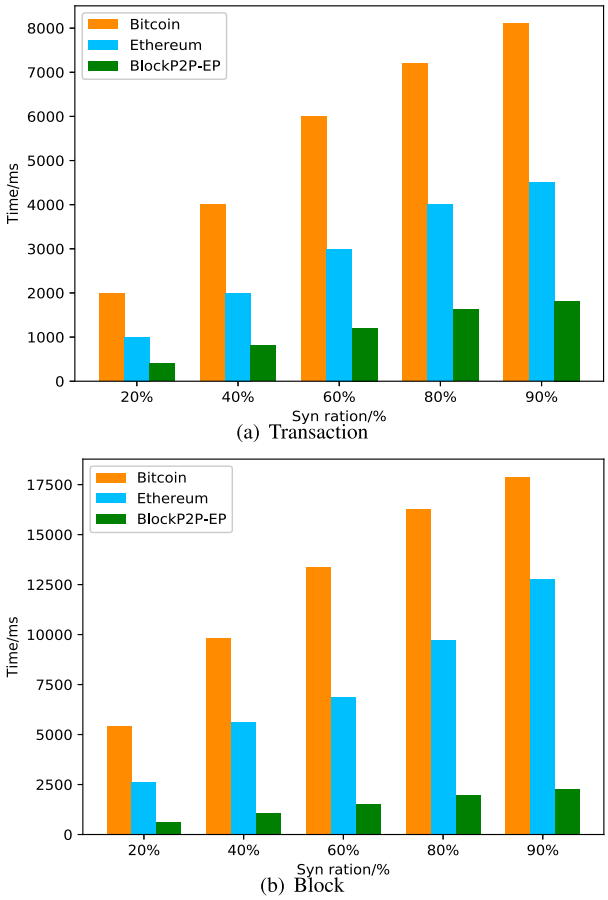


Fig. 7. Time spent on broadcasting the transaction/block.

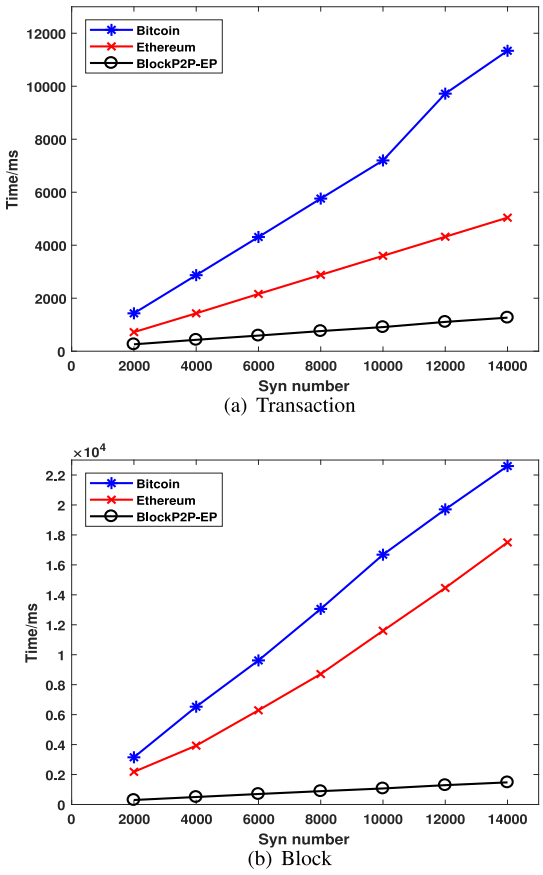


Fig. 8. Broadcast time with different number of nodes.

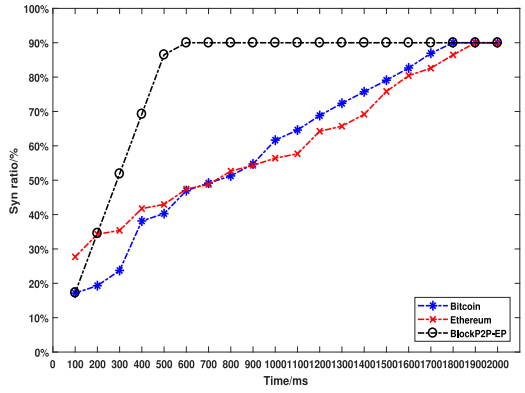
In more detail, we evaluate the network stability by measuring broadcast time when stochastic nodes following Poisson distributions join in or leave from the network every 100 milliseconds.

Network transmission reliability is defined as the ability to maintain transmission when failures occur in blockchain network, which mainly includes success rate of malicious nodes detection and Load rate of inactive node detection. The success rate of malicious nodes detection denotes whether the blockchain network can detect the nodes that send bogus transactions effectively. We assume that interactive node sends trusted transactions with a probability of 90%, and the network size is set as 1,000. Then we change the ratio of malicious nodes to explore how different blockchain network protocols perform on the success rate of malicious nodes detection. Besides this, we measure the influence of distinct blockchain network protocols on the success rate of malicious nodes detection under multiple simulation cycles. Load rate of inactive node detection indicates the network communication load when detecting inactive nodes. We assume that every node stores trust information for ten neighbors locally, and all nodes will deactivate with stochastic probability following Poisson distributions. Based on the above assumptions, we conduct experiments to explore the changes of network communication load caused by the detection protocols in different network scale and time cycles.

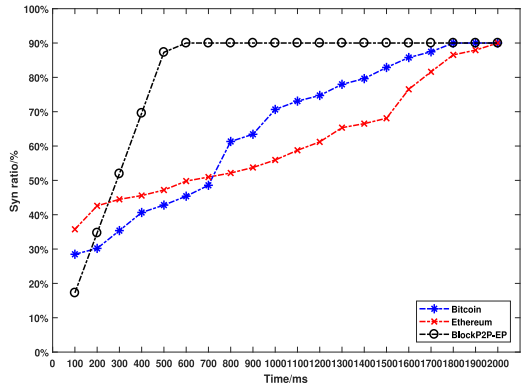
B. Results of Transmission Rate

1) General Performance: We first measure the time used to broadcast the data to different synchronization ratios of nodes. With the total number of nodes fixed as 14,000, two group of experiments for transactions and blocks are conducted respectively. As shown in Figure 7, the experimental results demonstrate that the broadcast time of BlockP2P-EP is less than Bitcoin and Ethereum both in transaction and block synchronization at different synchronization ratios. To be specific, when the block synchronization ratio reaches 90%, Bitcoin takes 17,880 milliseconds, while BlockP2P-EP only takes 2,270 milliseconds, which can reduce the network broadcast latency by 90%. At the same time, network synchronization time of BlockP2P-EP changes very little at different network synchronization ratios compared to Bitcoin and Ethereum. When the block synchronization ratio changes from 20% to 90%, synchronization time change for Bitcoin takes 12,430 milliseconds, while BlockP2P-EP only takes 1,630 milliseconds. To sum up, BlockP2P-EP can promote the network performance apparently.

2) Network Scalability: Now we fix the synchronization ratio at 90% and increase the number of nodes in each blockchain simulation network, to evaluate the network scalability of BlockP2P-EP. As shown in Figure 8, as the number of the network node increases, the data synchronization time required also gradually increases, both in transaction and block synchronization. However, as for the same size of network,

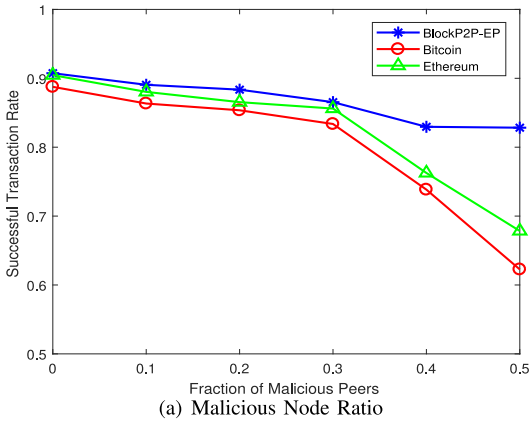


(a) Nodes join randomly

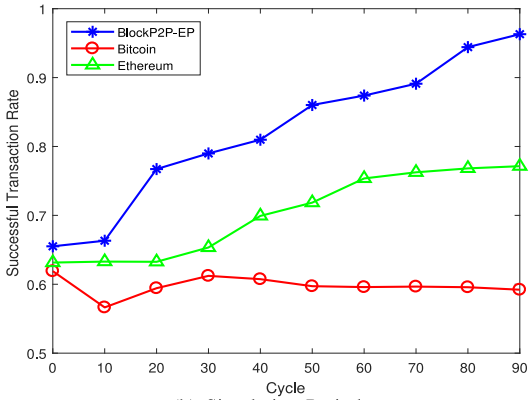


(b) Nodes leave randomly

Fig. 9. Broadcast time when nodes randomly join or leave.



(a) Malicious Node Ratio



(b) Simulation Periods

Fig. 10. Success rate of malicious nodes detection.

the synchronization time taken by BlockP2P-EP is smaller than Bitcoin and Ethereum. Specifically, when the number of nodes is 14,000, it takes 11,338 milliseconds for a transaction to propagate to 90% of the total nodes in Bitcoin. By contrast, it only takes 1,267 milliseconds for BlockP2P-EP. The similar phenomenon takes place in terms of block synchronization. In the meanwhile, network synchronization time of BlockP2P-EP changes very little at different network synchronization numbers compared to Bitcoin and Ethereum. When the network synchronization number changes from 2,000 to 14,000, synchronization time change for Bitcoin takes 11,000 milliseconds, while BlockP2P-EP only takes 1,620 milliseconds. As a result, we can conclude that BlockP2P-EP protocol can provide a higher system scalability.

3) Network Stability: In this section, we try to verify if BlockP2P-EP can maintain stability of latency when the number of nodes changes dynamically. With total number of nodes initialized as 14,000 and the synchronization ratio fixed as 90%, in one process of network synchronization, we increase or decrease stochastic nodes following Poisson distributions every 100 milliseconds to observe the time and fluctuation of the network. From Figure 9, we can find that BlockP2P-EP only takes 600 milliseconds, and the synchronization time fluctuates slightly. Compared with the original network scale, network synchronization time has basically not changed, and fluctuation of time is minor. While Bitcoin reaches the final synchronization ratio of 90%, Bitcoin takes 2,000 milliseconds. Compared with the original network scale, the synchronization time increases obviously, so the fluctuations are dramatic. Therefore, it shows that BlockP2P-EP can maintain better network stability than Bitcoin and Ethereum, when large number of nodes leave from or join in the network. At the same time, we also found an interesting phenomenon. In the process of network nodes dynamic leaving and joining, although Ethereum shows relatively good performance at the beginning stage, its performance is not as good as that of Bitcoin at the later stage, indicating that the unstructured topology is more resistant to disturbances than the structured topology.

C. Results of Transmission Reliability

1) Success Rate of Malicious Nodes Detection: Figure 10(a) shows that Bitcoin and Ethereum can effectively identify the malicious nodes which send untrustworthy transactions, when the proportion of malicious nodes is small. Therefore, the successful transaction rate of the cooperative node decreases slowly as the proportion of malicious nodes increases. Compared with other two network protocols, BlockP2P-EP shows greater advantages with the malicious nodes increases, for which nodes in Bitcoin and Ethereum cannot obtain the trust information of all neighbor nodes. Then we set the proportion of malicious nodes as 50%, and observe the changes of successful transaction rates under different simulation cycles. It can be seen that BlockP2P-EP performs much better, indicating that BlockP2P-EP has strong resistance to malicious nodes.

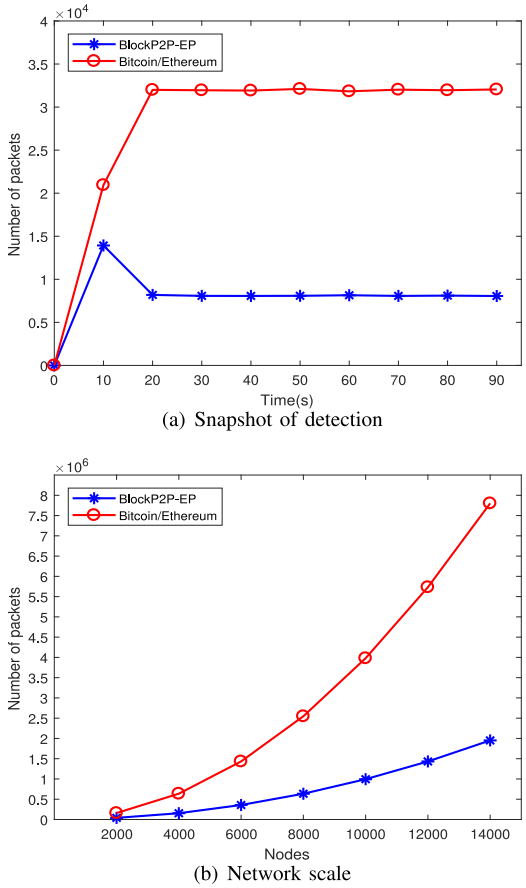


Fig. 11. Load rate of inactive node detection.

2) Load Rate of Inactive Node Detection: We compare the BlockP2P-EP protocol with the inactivation detection algorithm of Bitcoin and Ethereum in various network scales. Figure 11 (a) demonstrates the detection load comparison between the two algorithms in an execution fragment. Results shows that the conventional inactivation detection algorithm has 4 times network load than the BlockP2P-EP algorithm at the same network scale (e.g., N = 2,000). We configure several node sizes (from 2,000 to 14,000) to investigate the discrepancy in inactivation detection load at different node sizes. Figure 11 (b) shows that as the network scale becomes larger, both BlockP2P-EP algorithm and traditional blockchain inactivation detection algorithm will increase, but the network load of the latter is 4 times that of the former.

VI. Discussion

This paper focuses on improving both transmission rate and transmission reliability of blockchain P2P protocol, and addressing the associated scalability and security challenges. However, there remain several open problems to be solved in the future work:

- From the simulation results, BlockP2P-EP has improved blockchain network performance over the predominant permissionless blockchain systems (i.e., Bitcoin and Ethereum). However, the situation will be different for the consortium and private blockchain since the nodes are densely distributed in a specific geographical area with low network latency. Therefore we need to consider more diverse blockchain platforms to convince network optimization;
- Although BlockP2P-EP has optimized the current blockchain network performance, it only considers transmission speed and reliability. The metrics for evaluating the network performance include many aspects such as network capacity, forwarding rate, and transmission security, each metric will affect the network performance in varying degrees. To mature the network protocol, more complex factors need to be added in the future work;
- The actual network environment is complex, the network performance of nodes is various due to the heterogeneous hardware configuration (e.g., CPU, RAM, and bandwidth) and communication link status. A more universal blockchain simulator should be designed to set relevant parameters for complex network environments, which can provide a reliable verification tool for designing network protocols;
- The experimental measurement of BlockP2P-EP is performed in network simulation. Although the experimental results show that BlockP2P-EP has good performance, it still needs to be verified in a real network environment to manifest the practical significance of BlockP2P-EP;
- BlockP2P-EP tries to accelerate the data transmission rate while considering the network reliability, but the proportion of the factors effecting on network performance is obscure. Further research should be conducted to establish a model which can formalize the relevance of network factors, and finding the key issue in obstructing network performance.

We will continue studying the blockchain P2P network in depth and exploring the factors that restrict the blockchain P2P network performance from multiple aspects for future work.

VII. Conclusion

Blockchain performance has become the key challenge that impedes the development of blockchain ecosystem. Most of the researches draw their attention on the optimization of consensus layer or data layer, while lacking consideration of the underlying P2P network optimization. To improve blockchain performance, we take steps towards the network layer. In this article, we argue that blockchain network performance consists of two different dimensions including network rate and network reliability. So we first comprehensively analyze the influential factors of the entire blockchain network propagation, from the connection establishment phase and the data transmission phase, respectively. Then we summarize that network reliability is limited by the network trust connection and node inactivation detection. Based on our key findings, we then carry out a novel network protocol *BlockP2P-EP* to optimize the topology. To verify the feasibility and efficiency of BlockP2P-EP, we design and implement a unified blockchain network simulator *BlockSim* to evaluate the performance in terms of network rate and network reliability. The experimental results demonstrate that in comparison to existing Bitcoin and Ethereum, BlockP2P-EP can provide

better network performance for data broadcast, and maintain network scalability and stability.

REFERENCES

[1] I. Eyal, A. E. Gencer, E. G. Sirer, and R. V. Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *Proc. Usenix Conf. Netw. Syst. Design Implement.*, 2016, pp. 45–59.

[2] Z. Xu, S. Han, and L. Chen, "Cub, a consensus unit-based storage scheme for blockchain system," in *Proc. IEEE 34th Int. Conf. Data Eng. (ICDE)*, Paris, France, 2018, pp. 173–184.

[3] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Proc. IEEE 13th Int. Conf. Peer-to-Peer Comput.*, Trento, Italy, 2013, pp. 1–10.

[4] J. Li, G. Liang, and T. Liu, "A novel multi-link integrated factor algorithm considering node trust degree for blockchain-based communication," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 8, pp. 3766–3788, 2017.

[5] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *Proc. IEEE Eur. Symp. Security Privacy (EuroS P)*, Saarbrücken, Germany, 2016, pp. 305–320.

[6] T. Neudecker, P. Andelfinger, and H. Hartenstein, "Timing analysis for inferring the topology of the bitcoin peer-to-peer network," in *Proc. Ubiquitous Intell. Comput. Adv. Trusted Comput. Scalable Comput. Commun. Cloud Big Data Comput. Internet People Smart World Congr.*, Toulouse, France, 2016, pp. 358–367.

[7] C. Georgiou, S. Gilbert, R. Guerraoui, and D. R. Kowalski, "Asynchronous gossip," *J. ACM*, vol. 60, no. 2, pp. 1–42, 2013.

[8] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking, "Randomized rumor spreading," in *Proc. 41st Annu. Symp. Found. Comput. Sci.*, Redondo Beach, CA, USA, 2000, pp. 565–574.

[9] W. Hao *et al.*, "BlockP2P: Enabling fast blockchain broadcast with scalable peer-to-peer network topology," in *Proc. 14th Int. Conf. Green Pervasive Cloud Comput.*, 2019, pp. 223–237.

[10] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, Jul. 2004.

[11] P. Bhabak, H. Harutyunyan, and P. Kropf, "Efficient broadcasting algorithm in harary-like networks," in *Proc. 46th Int. Conf. Parallel Process. Workshops*, Bristol, U.K., 2017, pp. 162–170.

[12] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Gener. Comput. Syst.*, Aug. 2018.

[13] S. K. Kim, Z. Ma, S. Murali, J. Mason, A. Miller, and M. Bailey, "Measuring ethereum network peers," in *Proc. Internet Meas. Conf.*, 2018, pp. 91–104.

[14] N. Papadis, S. Borst, A. Walid, M. Grissa, and L. Tassiulas, "Stochastic models and wide-area network measurements for blockchain design and analysis," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Honolulu, HI, USA, 2018, pp. 2546–2554.

[15] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf/

[16] *Ethereum*. [Online]. Available: https://www.ethereum.org/

[17] *Nkn*. [Online]. Available: https://www.nkn.org/

[18] M. F. Sallal, G. Owen, and M. Adda, "Proximity awareness approach to enhance propagation delay on the bitcoin peer-to-peer network," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, Atlanta, GA, USA, 2017, pp. 2411–2416.

[19] K. Croman *et al.*, "On scaling decentralized blockchains," in *Proc. Int. Conf. Financ. Cryptography Data Security*, 2016, pp. 106–125.

[20] J. A. D. Donet, C. Pérez-Sola, and J. Herrera-Joancomartí, "The bitcoin P2P network," in *Proc. Int. Conf. Financ. Cryptography Data Security*, 2014, pp. 87–102.

[21] S. Sourav, P. Robinson, and S. Gilbert, "Slow links, fast links, and the cost of gossip," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Vienna, Austria, 2018, pp. 786–796.

[22] S. Delgado-Segura, C. Pérez-Solà, J. Herrera-Joancomartí, G. Navarro-Arribas, and J. Borrell, "Cryptocurrency networks: A new P2P paradigm," *Mobile Inf. Syst.*, vol. 2018, p. 16, 2018.

[23] S. Datta, C. Giannella, and H. Kargupta, "*K*-means clustering over a large, dynamic network," in *Proc. SIAM Int. Conf. Data Min.*, 2006, pp. 153–164.

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

[25] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network," in *Proc. Inf. Process. Sensor Netw.*, 2003, pp. 333–348.

[26] A. Montresor and M. Jelasity, "PeerSim: A scalable P2P simulator," in *Proc. IEEE 9th Int. Conf. Peer-to-Peer Comput.*, Seattle, WA, USA, 2009, pp. 99–100.

[27] *Bitnodes*. [Online]. Available: https://bitnodes.earn.com/nodes/

[28] *Ethernodes*. [Online]. Available: https://www.ethernodes.org/network/

Weifeng Hao (Student Member, IEEE) received the B.E. degree from the School of Computer Science and Technology, Shandong University, Jinan, China, in 2017. He is currently pursuing the M.S. degree with the School of Computer Science and Technology, HUST. His current research interests include blockchain and clouding computing.

Jiajie Zeng (Student Member, IEEE) received the B.E. degree from Software College, Northeastern University, Shenyang, China, in 2018. He is currently pursuing the M.S. degree with the School of Computer Science and Technology, Huazhong University of Science and Technology. His current research interests include P2P network, blockchain, and distributed system.

Xiaohai Dai (Student Member, IEEE) received the M.S. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2017, where he is currently pursuing the Ph.D. degree. His current research interests include blockchain and distributed system.

Jiang Xiao (Member, IEEE) received the B.Sc. and Ph.D. degrees from the Hong Kong University of Science and Technology in 2009 and 2014, respectively. She is currently an Associate Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. She has been engaged in research on blockchain, distributed computing, big data analysis and management, and wireless indoor localization. She was a recipient of the several awards, including the Hubei Dawnlight Program 2018, the CCF-Intel Young Faculty Research Program 2017, and the Best Paper Awards from IEEE ICPADS/GLOBECOM 2012.

Qiang-Sheng Hua (Member, IEEE) received the B.Eng. and M.Eng. degrees from the School of Computer Science and Engineering, Central South University, China, in 2001 and 2004, respectively, and the Ph.D. degree from the Department of Computer Science, University of Hong Kong, Hong Kong, in 2009. He is currently an Associate Professor with the Huazhong University of Science and Technology, China. He has published more than 60 papers in mainstream international conferences and journals on network and parallel distributed computing. His interested in parallel and distributed computing, including algorithms and implementations in real systems.

Hanhua Chen (Member, IEEE) received the Ph.D. degree in computer science and engineering from the Huazhong University of Science and Technology, China, in 2010, where he is currently a Professor with the School of Computer Science and Technology. His research interests include distributed systems and bigdata processing systems. He received the National Excellent Doctorial Dissertation Award of China in 2012.

Kuan-Ching Li (Senior Member, IEEE) is currently a Professor with the Department of Computer Science and Information Engineering, Providence University, Taiwan. He was a Department Chair in 2009. He has been the Special Assistant with the University President since 2010, and appointed as a Vice-Dean for Office of International and Cross-Strait Affairs (OIA) since 2014. He has been involved actively in conferences and workshops as a program/general/steering conference chairman positions, numerous conferences and workshops as a program committee member, and has organized numerous conferences related to high-performance computing and computational science & engineering. His research interests include GPU/many-core computing, big data, and cloud. He was a recipient of awards from Nvidia, the Ministry of Education, Taiwan, and the Ministry of Science and Technology, Taiwan, as also a Guest Professorship from universities in China, that includes Xiamen University, the Huazhong University of Science and Technology, Lanzhou University, Shanghai University, the Anhui University of Science and Technology, and Lanzhou Jiaotong University. He is a fellow of the IET.

Hai Jin (Fellow, IEEE) received the Ph.D. degree in computer engineering from the Huazhong University of Science and Technology in 1994. He worked with the University of Hong Kong from 1998 to 2000, and a Visiting Scholar with the University of Southern California from 1999 to 2000. He is a Cheung Kung Scholars Chair Professor of computer science and engineering with the Huazhong University of Science and Technology and the Chief Scientist of ChinaGrid, the Largest Grid Computing Project in China and the National 973 Basic Research Program Project of Virtualization Technology of Computing System, and Cloud Security. He has coauthored 22 books and published over 800 research papers. His research interests include computer architecture, virtualization technology, cluster computing and cloud computing, peer-to-peer computing, network storage, and network security. He received the Excellent Youth Award from the National Science Foundation of China in 2001. He received German Academic Exchange Service Fellowship to visit the Technical University of Chemnitz, Germany, in 1996. He is a fellow of CCF and a member of ACM.