# Self-Diagnosing and Healing System for Reliable Hybrid P2P-based Blockchain Against Abnormal Node Failure

Ho-se Moon
Department of Computer Science and Engineering
POSTECH
Pohang, Republic of Korea
mhs0272@postech.ac.kr

Seunggyu Ji
Department of Computer Science and Engineering
POSTECH
Pohang, Republic of Korea
jiseunggyu@postech.ac.kr

Hyunmin Noh
Division of Electronic Engineering
Jeonbuk National Univ.
Jeonju, Republic of Korea
hmnoh@jbnu.ac.kr

Hwangjun Song
Department of Computer Science and Engineering
POSTECH
Pohang, Republic of Korea
hwangjun@postech.ac.kr

***Abstract*—This work proposes a self-diagnosing and healing system for a fast and reliable blockchain service in a multi-cloud environment. Basically, the proposed whole system is implemented to concurrently satisfy both scalability and reliability in blockchain systems. At first, a hybrid P2P-based structure is adopted for block transmission to enhance the scalability. However, it may be vulnerable to abnormal nodes that are experiencing network layer attacks and vulnerabilities. Thus, the proposed system is designed to frequently check as many nodes as possible in order to detect abnormal nodes and quickly exclude them from the block transmission path. The proposed system is fully built by using various open-source tools (e.g. Mininet, Docker, and Containernet). Finally, the experimental results show that the proposed system improves the reliability of blockchain service compared with P2P tree-based systems at the cost of slightly increased block reception delay, and significantly enhances the scalability compared with existing Gossip-based systems.**

***Keywords—blockchain, P2P network, Hyperledger Fabric, self-diagnosing, self-healing.***

## I. INTRODUCTION

Blockchain [1] functions as a distributed ledger that preserves both data transparency and security. Despite its rapid growth, blockchain still encounters technical hurdles. One of the biggest challenges is the difficulty of achieving scalability, security, and decentralization simultaneously. This trade-off is often referred to as the blockchain trilemma [2]. Blockchain systems are typically divided into two types: permissioned blockchains, which sacrifice decentralization to enhance scalability, and permissionless blockchains, which sacrifice scalability to ensure decentralization. In the permissioned blockchain, only authorized nodes are allowed to participate in the network, whereas in the permissionless blockchain, any node can participate.

Recently, many researchers have worked on enhancing peer-to-peer (P2P) network [3] performance for more efficient blockchain systems. The Gossip protocol [4], which is commonly used in blockchain networks, propagates blocks randomly, which causes scalability issues. The P2P tree topology [5] enables high performance but it may be vulnerable to various network layer attacks such as IP spoofing, routing table poisoning and denial-of-service, etc. Actually, these may lead to abnormal node failure, where nodes are unable to normally propagate blocks over P2P tree topology. Lin et al. [6] addressed this by relocating abnormal nodes toward the leaves to improve reliability. Solana [7] is known as a high-performance blockchain. It adopts a P2P tree-based routing to quickly propagate blocks to a large number of validators. The P2P mesh topology [5] offers greater robustness but is constrained by the node bandwidth. Consequently, hybrid P2P tree-based networks that blend P2P tree and P2P mesh characteristics have been widely studied. Noh et al. [8] proposed a resilient and fast block transmission system based on a hybrid P2P structure. Their approach considers network conditions such as available bandwidth, congestion, and delay. Rathee et al. [9] proposed a hybrid blockchain mechanism to enhance security in the Internet of Things.

This work proposes efficient self-diagnosing and self-healing processes for reliable hybrid P2P-based blockchain systems. The proposed system includes the diagnosing node set and block height exchanging interval determination method, the abnormal node detection algorithm, the abnormal node localization algorithm, and the self-healing process. The details of the proposed self-diagnosing and healing system is presented in Section II. The experimental results are given in Section III, and finally, the concluding remarks are provided in Section IV.

## II. PROPOSED SELF-DIAGNOSING AND HEALING SYSTEM

The proposed system aims to achieve both scalability and reliability by combining a hybrid P2P-based topology with the proposed self-diagnosing and self-healing processes although a few abnormal nodes are suffering from the network layer attacks and vulnerabilities. Gossip protocol ensures robustness to abnormal nodes by randomly propagating blocks, which we define as proactive spatial randomness in this work. However, this approach reduces the transactions per second (TPS) owing to redundant block propagation and increased message overhead. The proposed self-diagnosing and self-healing processes are implemented to frequently monitor many nodes to diagnose the abnormal nodes and quickly exclude them from the block transmission path. Consequently, the system can maintain reliability regardless of where several abnormal nodes exist in the hierarchy. It is defined as reactive temporal randomness in this work. While proactive spatial randomness prevents the undesirable effects of abnormal nodes in advance, reactive temporal randomness quickly responds to them after detection.

Basically, the proposed system uses the HP$^2$B (hybrid P2P-based blockchain) [8, 10] as its base platform although the self-diagnosing and self-healing components can be applied to any other platform with minor modifications. The overall architecture is shown in Fig. 1, which includes the Network Monitoring and Management (NMM) module, Block Transmission (BT) module, Self-Diagnosing (SD) module, Self-Healing (SH) module, and Hyperledger Fabric as the underlying blockchain platform. The NMM and BT modules are inherited from HP$^2$B. In HP$^2$B, the original Gossip-based block transmission module of Hyperledger Fabric was replaced with hybrid P2P network. The NMM module measures the network delay among nodes and constructs a hybrid P2P Overlay Multicast Tree (OMT) for efficient block dissemination. The BT module transmits blocks over this tree and sends a pull request to neighboring nodes to recover the missing blocks. The SD module detects the existence of abnormal nodes and localizes their positions in the tree. The SH module uses this information to exclude the abnormal nodes from the block transmission path and recover it. In the proposed system, all participating nodes are categorized as diagnosing nodes, covered nodes, or uncovered nodes. Covered nodes have at least one diagnosing node among their descendants, while uncovered nodes do not. The working procedure of the proposed whole system is shown in Fig. 2. As shown in the figure, the NMM module of the manager node sends control messages to all the participating nodes as well as the block generating node to create a 3D coordinate based on the observed network delays. Then, the block generating node sends the estimated current transaction rate to the manager node. Using this information, the clustering is performed and OMT is constructed by the NMM module of the manager node. Based on the constructed OMT, the diagnosing node set and block height exchanging interval are also determined by the NMM module of the manager node. The resulting OMT, neighbor node information, diagnosing node set, and block height exchanging interval are then shared with the block generating node and all the participating nodes via control messages. The BT modules of the block generating node and the participating nodes use the received OMT information to update their forwarding tables. From the control message, the SD module of each participating node can identify whether the node itself is determined as a diagnosing node and recognize the other diagnosing nodes in the network. While blocks are being transmitted, the SD module monitors whether blocks have been arriving normally by either comparing block heights exchanged among diagnosing nodes or inspecting block reception intervals. If abnormal nodes are detected, the SD module localizes their exact positions within the network and then notifies the SH module. Upon receiving this information, the SH module excludes the abnormal nodes from the block transmission path and performs block transmission path recovery.

We now describe the self-diagnosing and self-healing processes. Beforehand, the message formats required to explain these processes are defined in Fig. 3.

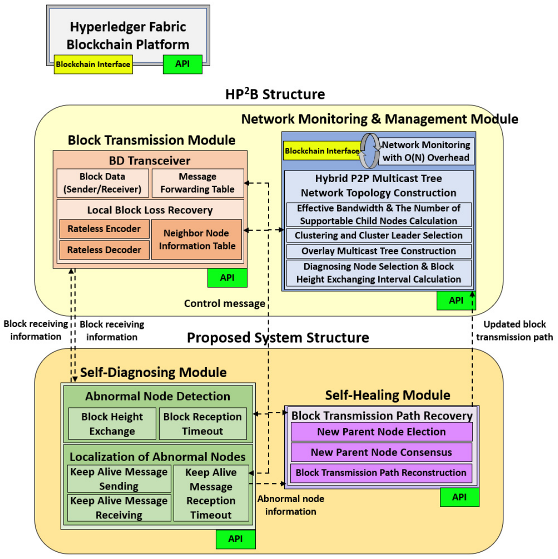


Fig. 1. Overall system architecture.

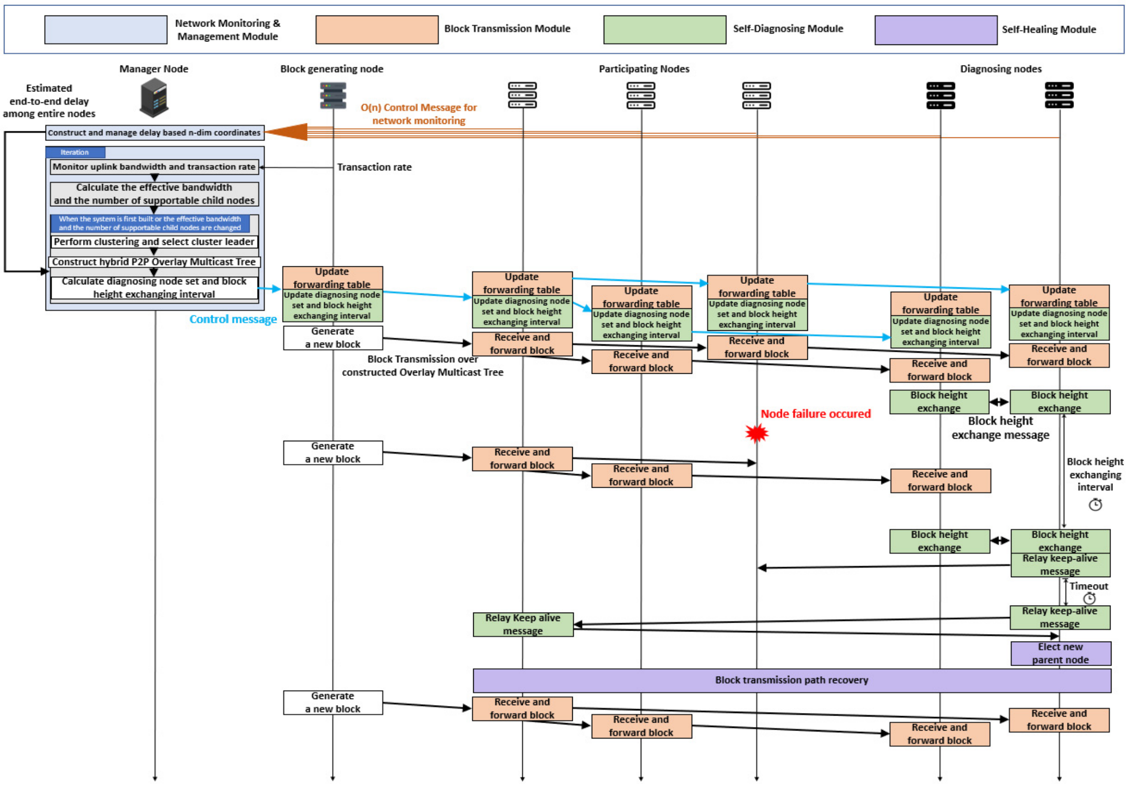


Fig. 2. The working procedure of the proposed system.

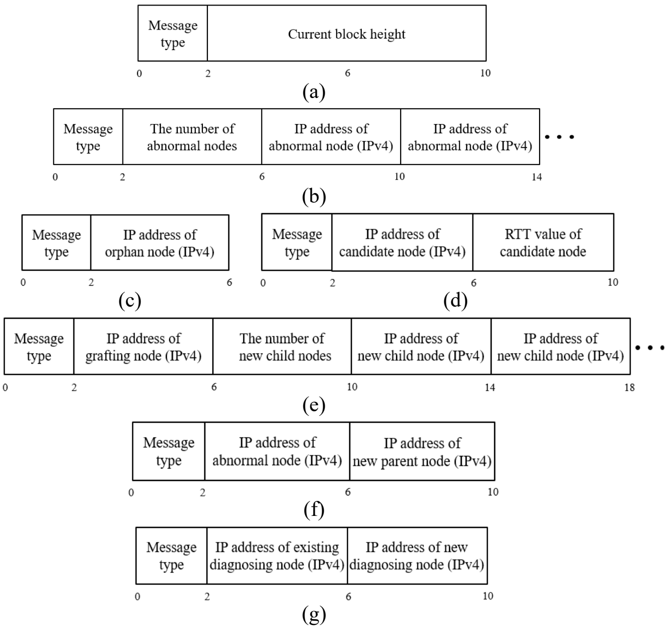


Fig. 3. Message formats for self-diagnosing and healing processes (the numbers denote bytes): (a) block height exchange message, (b) abnormal node notification message, (c) candidate discovery message, (d) candidate information exchange message, (e) block transmission path notification message, (f) new parent node notification message, and (g) diagnosing node delegation message.

### *A. Self-Diagnosing Process*

The proposed diagnosing process includes not only detecting the abnormal node existence, but also localizing abnormal nodes in the block transmission path. We consider two approaches for detecting abnormal node existence, i.e. abnormal node detection among the covered nodes by exchanging block heights among diagnosing nodes and abnormal node detection based on the block reception timer at each uncovered node. After an abnormal node is detected, it is localized by iteratively relaying keep-alive messages. In the hybrid P2P-based system, when an abnormal node exists among the covered nodes, the diagnosing node fails to receive blocks normally. This allows the diagnosing node to detect abnormality by comparing block heights with other diagnosing nodes. However, due to limited bandwidth, it is crucial to optimally determine the number and positions of the diagnosing nodes as well as the block height exchange interval. Now we can formulate the problem of determining the control variables as follows.

**Problem Formulation:** Determine $S_{diag}$ and $t_{interval}$ with the given $c_{BW} \cdot BW_{eff}$ to maximize the following cost function

$$C\left(S_{diag},t_{interval}\right)=\alpha\cdot R_{coverage}\left(S_{diag}\right) + (1-\alpha)\cdot\left(1-\frac{t_{interval}}{t_{interval}^{\max}}\right), \tag{1}$$

$$\text{subject to } bw_{detect}\left(S_{diag},t_{interval}\right)\leq c_{BW}\cdot BW_{eff}, \tag{2}$$

where $\alpha$ is the weighting factor $(0\leq\alpha\leq 1)$, $t_{interval}$ is a block height exchanging interval, the coverage $R_{coverage}\left(S_{diag}\right)$ with the given diagnosing node set $S_{diag}$ is defined as

$$R_{coverage}\left(S_{diag}\right)=\frac{\left|\sum_{i\in S_{diag}} A(i)\right|}{|N|}, \tag{3}$$

where $|A(i)|$ denotes the cardinality of the ancestor node set at the node *i,* and $|N|$ denotes the cardinality of the set of all nodes, the bandwidth $bw_{detect}\left(S_{diag},t_{interval}\right)$ required for detection is calculated by

$$bw_{detect}\left(S_{diag},t_{interval}\right)=\frac{B_{msg}\cdot\left|S_{diag}\right|\cdot N_{fo}}{t_{interval}}, \tag{4}$$

where $c_{BW}$ denotes the bandwidth constraint ratio, $B_{msg}$ is the message size for exchanging block heights, $N_{fo}$ is the number of nodes that a diagnosing node contacts to exchange the block height, and $t_{inteval}^{\max}$ is the block height exchanging interval when all leaf nodes are selected as diagnosing nodes, which is obtained by

$$t_{interval}^{\max}=\frac{B_{msg}\cdot|L|\cdot N_{fo}}{c_{BW}\cdot BW_{eff}}, \tag{5}$$

where $|L|$ denotes the cardinality of the entire leaf node set. Theoretically, the optimal diagnosing node set and block height exchanging interval of the above problem is obtained by

$$\left(S_{diag}^{*},t_{interval}^{*}\right)=\underset{\left(S_{diag},t_{interval}\right)}{\arg}\max_{\substack{S_{diag}\in P(L)-\{\varnothing\}\\ t_{interval}\geq t_{\min}\left(S_{diag}\right)}} C\left(S_{diag},t_{interval}\right), \tag{6}$$

where $P(L)$ is the power set of entire leaf node set and $t_{\min}\left(S_{diag}\right)$ is the lower bound of the feasible range for $t_{interval}$ satisfying the bandwidth constraint for the given $S_{diag}$.

**Practical Parameter Determination:** As shown in Eq. 6, we need a full search-based optimization method to obtain the optimal solution. The diagnosing node set is discrete while the block height exchanging interval is continuous. For the fast processing, $\left(S_{diag}^{*},t_{interval}^{*}\right)$ can be achieved in two steps. Firstly, the continuous range of $t_{interval}$ is discretized and exhaustively searched to find the optimal value that maximizes the cost function Eq. 7 with a given diagnosing node set $\hat{S}_{diag}$, i.e.

$$t_{interval}^{*}\big|_{\substack{with\ given\\ \hat{S}_{diag}}}=\arg\max_{t_{interval}\geq t_{\min}\left(\hat{S}_{diag}\right)} C\left(\hat{S}_{diag},t_{interval}\right). \tag{7}$$

Secondly, the above step is repeated for every element in the power set of entire leaf node set, and the optimal diagnosing node is determined by

$$S_{diag}^{*}=\arg\max_{\hat{S}_{diag}\in P(L)-\{\varnothing\}} C\left(\hat{S}_{diag},t_{interval}^{*}\big|_{\substack{with\ given\\ \hat{S}_{diag}}}\right). \tag{8}$$

Although the above full search-based method guarantees the globally optimal diagnosing node set and block height exchanging interval, it requires the computational complexity of $O\left(2^{n}\right)$, which may be a big burden for the real-time processing as the number of nodes increase. Thus, a greedy-based method with $O(n)$ complexity is adopted as a scalable and efficient alternative for large-scale networks. In contrast to the full search-based method that explores all elements in the power set of entire leaf node set, the greedy-based method sequentially selects the leaf nodes that yield the largest increase in the coverage.

**Detection of Abnormal Node Existence**: Now, we present a method for detecting the abnormal node existence using the diagnosing node set and the block height exchanging interval determined above. Each diagnosing node periodically and randomly selects a subset of the other diagnosing nodes and exchanges block height exchange messages shown in Fig. 3 (a). Upon receiving these messages, a diagnosing node compares the received block heights with its own. If the difference exceeds a predefined threshold, it infers that an abnormal node exists among its ancestor nodes. This algorithm enables the distributed and efficient detection of block transmission failures with minimal overhead.

Actually, it is also necessary to detect abnormal nodes for even the other nodes not covered by the diagnosing nodes. Hence, we consider a lightweight detection algorithm, in which every node independently monitors its block reception intervals in the sliding window and computes their average value. When a new block is received, the node sets the timeout to this average value. If the timeout expires before the next block arrives, the node suspects abnormal behavior in its ancestor nodes. To confirm this, the node randomly selects a set of peers and sends messages requesting their block heights. As soon as response messages arrives, the node compares the reported block heights with its own. If the difference exceeds a predefined threshold, the node infers the abnormal node existence among its ancestors.

**Localization of Abnormal Nodes**: After detecting the abnormal node existence among the ancestor nodes, the localization of the abnormal node on the OMT is performed by relaying lightweight messages containing only a message type, referred to as keep-alive messages between the parent and child nodes. That is, each node sequentially sends a keep-alive message to its parent node to verify its operational status. If the child node receives a response from its parent within a predefined timeout, the parent is considered to be in a normal status. Otherwise, it is assumed that the corresponding parent node is abnormal, and the child node sends the same keep-alive message to the grandparent node. If a response is then received from the grandparent, the parent is confirmed to be abnormal. Occasionally, consecutive ancestor nodes may be abnormal. In such cases, the child node continues to send keep-alive messages sequentially to its ancestor nodes along the block transmission path until it receives a response, thereby allowing all abnormal nodes to be localized. The first normally operating ancestor that responds is defined as the grafting node, and the set of localized abnormal nodes is defined as the healing zone, which is handled by the following self-healing process.

### B. *Self-Healing Process*

Once abnormal nodes are diagnosed, the self-healing process is executed. The block transmission path recovery algorithm is implemented to exclude the abnormal parent node and elect a normal node as the new parent node. Depending on the size of healing zone, it performs either the global updating or the local healing. If the healing zone size exceeds a predetermined threshold, the global update is performed by reconstructing the entire OMT. Otherwise, the local healing is performed. By the way, the resulting OMT by the local healing is only sub-optimal in terms of delay. During local healing, all the child nodes of the abnormal parent are defined as orphan nodes. Examples are presented in Figs. 4 and 5, where a single or multiple abnormal nodes are localized.

**Abnormality Notification**: The first orphan node that detects an abnormal parent notifies the other orphan nodes using abnormal node notification message shown in Fig. 3 (b).

**Candidate Discovery**: Each orphan node recommends a candidate independently from its descendant leaf nodes with no child nodes. To discover these candidates, each orphan node sends a candidate discovery message shown in Fig. 3 (c) to its child nodes while requesting that the message be relayed downward until a leaf node is reached. As shown in the figure, the message allows descendant leaf nodes to respond directly. When a leaf node receives the request, it responds to indicate it is available as a new parent node. Each orphan node recommends the first respondent as its candidate.

**New Parent Node Election:** Each orphan node measures the round-trip time (RTT) to its candidate, and exchanges the candidate information exchange message shown in Fig. 3 (d) with the other orphan nodes. Based on this shared information, they elect the candidate with the shortest RTT as the new parent node.

**Block Transmission Path Configuration:** The recommender sends block transmission path notification message shown in Fig. 3 (e) to the elected candidate to establish the new block transmission path. When the new parent node receives this message, it identifies the node from which to receive blocks and the child nodes to forward them to, and then reconfigures its forwarding role accordingly.

**Block Transmission Path Redirection:** After configuring its role, the new parent node must receive blocks from the grafting node. However, because the grafting node is unaware of the path change, it continues to forward blocks to the abnormal node. To resolve this problem, the new parent node sends new parent node notification message shown in Fig. 3 (f) to the grafting node. This prompts the grafting node to forward blocks to the new parent node instead. In addition, the new parent node informs its own parent using new parent notification message to stop forwarding blocks to it, thereby avoiding duplication. Any missing blocks are retrieved by the BT module, which requests the missing blocks from neighbor nodes through a pull mechanism.

**Diagnosing Node Delegation:** If the new parent node has previously served as a diagnosing node, it must delegate that role to ensure continued monitoring of the network. The new diagnosing node is selected from among the non-diagnosing descendants of the recommender, with preference given to the node that increases the coverage the most. The new parent sends diagnosing node delegation messages shown in Fig. 3 (g) to the new and existing diagnosing nodes to update the configuration.

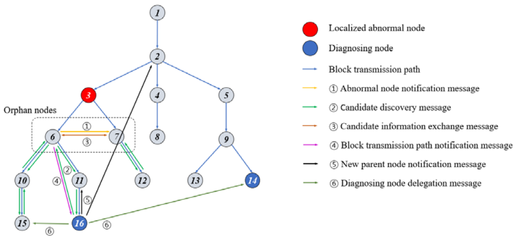


Fig. 4. Message exchange example for self-healing process when a single abnormal node is localized.

Some modifications are required to handle the multiple consecutive abnormal node case shown in Fig. 5. In this context, a generation refers to a level in the OMT where all nodes share the same depth from the root. In this case, the same number of new parent nodes as abnormal nodes should be elected. There are two major differences compared to the single abnormal node case. First, the first node to detect multiple consecutive abnormal nodes must notify not only orphan nodes of its generation but also orphan nodes across other parts of the tree using abnormal node notification message shown in Fig. 3 (b). This ensures that all affected nodes can elect new parent nodes. Second, after the new parent nodes are elected, their recommenders exchange the candidate information across generations using new parent node notification message shown in Fig. 3 (f), and it allows new parent nodes in different generations to configure their block transmission paths.

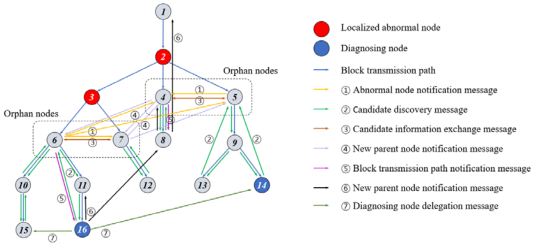


Fig. 5. Message exchange example for self-healing process when multiple consecutive abnormal nodes are localized.

TABLE 1. EXPERIMENTAL PARAMETER SETTINGS.

| Symbol | Description | Value |
|---|---|---|
| $N_{trx}^{blk}$ | The number of transactions in a block | 25 |
| $B_{TX}$ | Transaction size | 4000 bytes |
| $B_{blk\_hd}$ | Block header size | 1000 bytes |
| $\lambda$ | Transaction rate | 15 |
| $P_{fin}$ | Success probability | 97.72% |
| $N_{child}$ | The number of supportable child nodes | 3 |
| $N_{fo}$ | The number of messages generated by a diagnosing node for block height exchange | 5 |
| $c_{BW}$ | Constraint rate on bandwidth | 0.05 |
| $TTL$ | Time-to-live value in Enhanced Gossip | 5 |
| $k^{conn}$ | The minimum degree of connectivity in Block P2P-EP | 5 |
| $\lvert N^{neighbor} \rvert$ | The number of neighbor nodes of Rate-based algorithm | 7 |
| $\chi^{init}$ | Initial transmission rate of Rate-based algorithm | 0.5 |
| $\chi^{threshold}$ | Threshold of transmission rate of Rate-based algorithm | 0.8 |
| $T^{diag}$ | Threshold of the block height discrepancy | 3 |
| $W$ | Sliding window size | 5 |
| $M$ | Multiplier for average stored block reception interval values | 10 |
| $\lvert N_{fo}^{uncovered} \rvert$ | The number of nodes for block height exchange for abnormal node detection among uncovered nodes | 3 |

## III. EXPERIMENTAL RESULTS

The proposed system was implemented using Mininet [11], Docker [12], and Containernet [13], along with Go, C, and Python on an Ubuntu 20.04 workstation. A realistic multi-cloud environment was emulated, and Fig. 6 shows the network topology with delay settings based on real Internet RTTs measured by Amazon WorkSpaces Health Check [14]. The system includes a block generating node, a manager node, and 100 participating nodes due to practical reasons, which was the maximum number supported by our hardware (i.e. VM workstation with Intel 9-13900KS CPU, 32GB Memory, and 2TB SSD). The experimental parameter setting is given in Table 1.

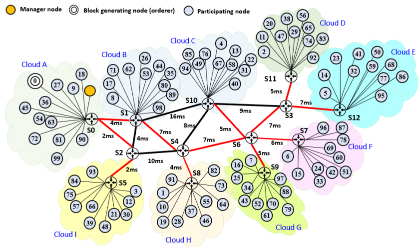


Fig. 6. Network topology of multi-clouds under consideration (the red lines are activated links and the black lines are deactivated links by the Spanning Tree Protocol of routers).

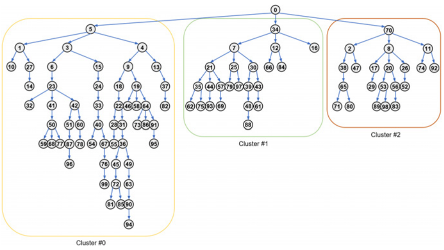
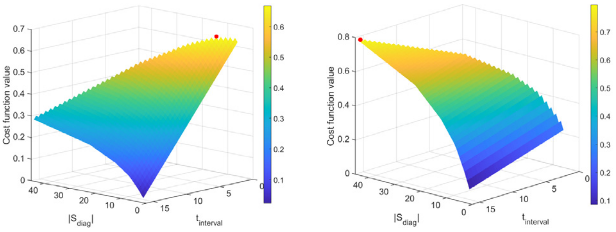


Fig. 7. OMT for block transmission.

### A. *Performance Verification of the Proposed System*

In this section, we present the performance of the self-diagnosing and self-healing processes in the proposed system. Fig. 6 shows the network topology used for the test. Fig. 7 shows the OMT used in the experiment, where the numbers above each node denote their node IDs.



(a) (b)

Fig. 8. The cost function value in terms of coverage, block height exchanging interval and $\alpha$ : (a) when $\alpha$ = 0.3 and (b) $\alpha$ = 0.8.

Based on the OMT, the diagnosing node set and the block height exchanging interval were determined. The cost function values calculated with respect to each diagnosing node set and block height exchanging interval are shown in Fig. 8. Thus, only the regions satisfying the constraint are shown in the figures. In Fig. 8 (a), where a smaller value of $\alpha$, the cost function becomes more sensitive to changes in the block height exchanging interval. As a result, a shorter block height exchanging interval was selected at the expense of reduced coverage. In contrast, in Fig. 8 (b), where a larger value of $\alpha$, the cost function becomes more sensitive to changes in the coverage. As a result, more diagnosing nodes were selected at the expense of a longer block height exchanging interval. During the experiment, $\alpha$ was set to 0.538. The resulting $S_{diag}$ and $t_{interval}$ were

{#14, #29, #32, #52, #55, #59, #61, #62, #66, #69, #71, #74, #81, #82, #83, #88, #94, #95, #96, #99} and 7.9 s, respectively.

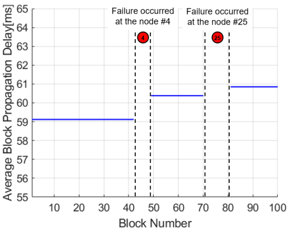


Fig. 9. Average block propagation delay for 100 nodes from the first to 100th blocks.

Fig. 9 shows the average block propagation delay across all nodes during the dissemination of first to the 100th blocks when Nodes #4 and #25 sequentially failed. Notably, the average block propagation performance degradation before and after the two healing processes was negligibly less than 3%. During the self-diagnosing and self-healing processes, some nodes could not receive blocks, rendering the average E2E delay data invalid.

## B. Performance Comparison with Existing Systems

In this section, the block propagation performance of the proposed system is compared with various block propagation algorithms, including Gossip, Enhanced Gossip [15], BlockP2P-EP [16], Rate-based algorithm [17], and HP$^2$B [8, 10]. Fig. 10 shows a histogram of 1,800 block arrivals recorded for the first to 18th blocks across 100 nodes. Gossip showed a broad block transmission range, with many nodes failing to receive blocks even after extended periods. Enhanced Gossip improved, with more nodes receiving blocks within 0 to 200 ms, but still had a notable number of unreceived blocks over long durations. BlockP2P-EP delivered most blocks within 0 to 300 ms, although some arrived between 1200 and 1300 ms. Rate-based algorithm delivered more blocks within 0 to 100 ms compared with Gossip, but still exhibited a long heavy tail. HP$^2$B received almost all blocks within 0 to 200 ms. The proposed system achieved the most block arrivals within 0 to 100 ms, with minimal performance degradation compared with HP$^2$B. Furthermore, as shown in the figure, HP$^2$B delivered only 1,592 block to all of its descendants when Node #4 fails. While, the proposed system successfully brought all blocks after the self-diagnosing and self-healing processes although it took some time for the self-diagnosing and self-healing processes of the failure at Node #4.

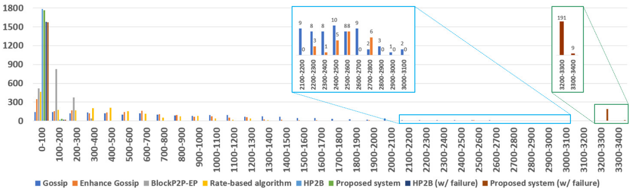


Fig. 10. Histogram of block arrival times when uplink bandwidth was 100Mbps and $N_{trx}^{blk}$ was 25.

# IV. CONCLUSION

In this work, we have proposed a self-diagnosing and healing system for a Hybrid P2P-based blockchain in a multi-cloud environment. The self-diagnosing process includes an abnormal node detection algorithm and an algorithm for the localization of abnormal nodes. In particular, for the detection algorithm among covered nodes, we have proposed a full search-based method and a greedy-based method for determining the diagnosing node set and the block height exchanging interval. The self-healing process includes a block transmission path recovery algorithm. In this work, we consider a single manager node and a single block generating node. As our future work, we are planning to adopt the Raft algorithm to operate multiple manager nodes and block generating nodes to enhance the reliability of the proposed system.

# ACKNOWLEDGMENT

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00484, Core Technologies for Hybrid P2P Network-based Blockchain Services and No. RS-2024-00404972, Development of 5G-A vRAN Research Platform).

# REFERENCES

[1] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, “Blockchain,” *Bus. Inf. Syst. Eng.*, vol. 59, no. 3, pp. 183–187, Mar. 2017.

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

[3] J. F. Buford, H. Yu, and E. K. Lua, *P2P networking and applications*, Burlington, MA, USA: Morgan Kaufmann, 2009.

[4] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic algorithms for replicated database maintenance," in *Proc. 6th Annu. ACM Symp. Princ. Distrib. Comput. (PODC),* Vancouver, Canada, 1987, pp. 1–12.

[5] R. Hekmat, *Ad-Hoc Networks: Fundamental Properties and Network Topologies*. Zoetermeer, Netherlands: Springer, 2006.

[6] C. Lin, W. Chen, and P. Lin, "A decentralized tree-based algorithm for reliable blockchain communications," in *Proc. IEEE Int. Symp. Netw. Comput. Commun. (ISNCC)*, Montreal, Canada, 2020, pp. 1–6.

[7] “Solana.” 2023. [Online]. Available: https://solana.com/

[8] H. Noh, S. Ji, Y. Go, G. Park, and H. Song, "Resilient and fast block transmission system for scalable hyperledger fabric blockchain in multi-cloud environments," *IEEE Trans. Netw. Service Manag.*, vol. 21, no. 5, pp. 5118–5134, Oct. 2024.

[9] G. Rathee, F. Ahmad, R. Sandhu, C. A. Kerrache, and M. A. Azad, “On the design and implementation of a secure blockchain-based hybrid framework for industrial Internet-of-Things,” *Inf. Process. Manage.*, vol. 58, no. 3, Art. no. 102526, May 2021.

[10] H. Song, D. Lee, and H. Oh, “Application layer multicast tree constructing algorithm for real-time media delivery,” *Comput. Commun.*, vol.29, no.9, pp. 1480–1491, May 2005.

[11] “Mininet.” 2023. [Online]. Available: http://mininet.org/

[12] “Docker.” 2023. [Online]. Available: https://www.docker.com/

[13] “Containernet.” 2023. [Online]. Available: https://containernet.github.io/

[14] “Amazon WorkSpaces Health Check.” Available: https://clients.amazonworkspaces.com/Health.html

[15] N. Berendea, H. Mercier, E. Onica, and E. Riviere, “Fair and efficient gossip in hyperledger fabric,” in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Singapore, Singapore, 2020, pp. 190–200.

[16] W. Hao, J. Zeng, X. Dai, J. Xiao, Q. S. Hua, H. Chen, K. C. Li, and H. Jin, “Towards a trust-enhanced blockchain P2P topology for enabling fast and reliable broadcast,” *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 2, pp. 904–917, June 2020.

[17] P. Zhang, Y. Tao, Q. Zhao, and M. Zhou, “A rate-and-trust-based node selection model for block transmission in blockchain networks,” *IEEE Internet Things J.*, vol. 10, no. 2, pp. 1605–1616, Jan. 2023.