SimBlock: A Blockchain Network Simulator

Yusuke Aoki*, Kai Otsuki*, Takeshi Kaneko*, Ryohei Banno† and Kazuyuki Shudo‡
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo, Japan
Email: *{aoki.y.au, ootuki.k.aa, kaneko.t.ay}@m.titech.ac.jp, †banno@computer.org, ‡shudo@is.titech.ac.jp

***Abstract*—Blockchain, which is a technology for distributedly managing ledger information over multiple nodes without a centralized system, has elicited increasing attention. Performing experiments on actual blockchains are difficult because a large number of nodes in wide areas are necessary. In this study, we developed a blockchain network simulator SimBlock for such experiments. Unlike the existing simulators, SimBlock can easily change behavior of node, so that it enables to investigate the influence of nodes' behavior on blockchains. We compared some simulation results with the measured values in actual blockchains to demonstrate the validity of this simulator. Furthermore, to show practical usage, we conducted two experiments which clarify the influence of neighbor node selection algorithms and relay networks on the block propagation time. The simulator could depict the effects of the two techniques on block propagation time. The simulator will be publicly available in a few months.**

***Index Terms*—blockchain, simulator, peer-to-peer**

I. Introduction

Blockchains, which are a core technology of cryptocurrency, are eliciting attention owing to its various application possibilities in numerous aside from cryptocurrency. Two of the features of blockchains are the possibility to manage ledger information without a centralized system even in a group including multiple malicious nodes and the difficulty in tampering previously obtained data. Due to these features, blockchains are used in many cryptocurrencies, and their applications are extensively investigated. Accordingly, various research subjects are discussed, such as approval time and scalability. In conducting the research on these issues, blockchain experiment are often necessary. However, excluding simple experiments completed at a single node, experiments on blockchains are costly. In case of preparing a node for the experiment in a public blockchain network built over a wide area, information on the entire network are hard to obtain. If a private experimental network is constructed, the information on the entire network can be obtained. Nevertheless, preparing a large number of nodes is costly, as well as experimental conditions and network configuration cannot be easily changed.

In this study, we developed a blockchain network simulator SimBlock for the research on blockchains. SimBlock is event-driven, that considering block generation and message transmission/reception as events. This simulator enables to easily implement the algorithm of the neighbor nodes selection. Given that block creation time is calculated from the success probability of block generation, reproducing the mining that requires a large calculation power is unnecessary, and a network that involves many nodes can be simulated. By modifying the block generation probability, simulating various mining algorithms is possible.

The rest of this paper is organized as follows. Section II provides an overview of the blockchain as background knowledge. Section III explains and evaluates SimBlock. Section IV discusses the performed experiments with SimBlock as the application example. Finally, Section V elaborates the summary and future work.

II. Blockchain

In this section, we outline the blockchain as background knowledge. A blockchain is a distributed ledger technology regarded as part of Bitcoin [1], proposed by Satoshi Nakamoto. The nodes involved in the blockchain constitute a peer-to-peer network, and a consensus algorithm is established to ensure that the nodes in the network have the same ledger information.

A. Transaction propagation and consensus

The data to be recorded in blockchains are called transactions and are broadcasted among the nodes involved in a blockchain network. This transaction is stored in the transaction pool of each node but has not been recorded in the ledger yet. To store the transaction in the ledger, a block where the plurality of transactions is collected should be generated. By broadcasting this block, the transactions included in the block are approved and recorded in the ledger. Each block contains a hash value of the immediately preceding block. Moreover, to change the transactions involved in the past block, rewriting all subsequent blocks is necessary. Therefore, by adopting a mechanism that appropriately determines the node generating a new block, the blockchains can be stored with difficulty in tampering transactions even in environments with possible multiple malicious nodes. As shown in Fig. 1, given that the blocks are connected in a row by including the immediately preceding hash, this technique is called a blockchain.

Several algorithms have been proposed to determine the node that generates the block. One of the most extensively known algorithms is Proof of Work (PoW) used in Bitcoin. In PoW, a node that generates a new block is determined based on the computing power of the node. Each block includes a value freely set by each node called a nonce, and each node locates a block whose hash value of the entire block is below a certain threshold while changing this nonce. Only the blocks below the threshold are regarded as formal blocks. Therefore, a node that discovers a nonce satisfying the previously mentioned

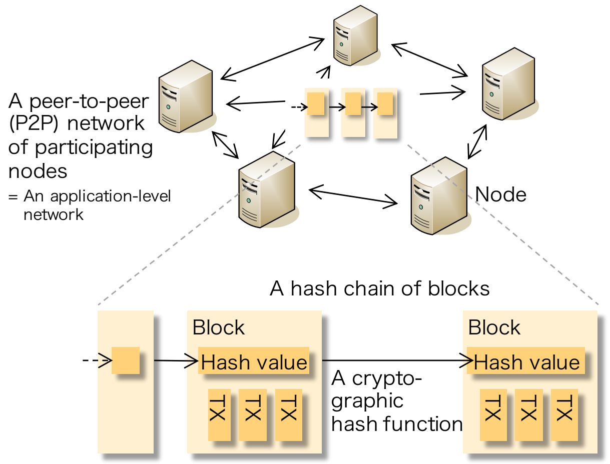
A peer-to-peer (P2P) network of participating nodes
= An application-level network
Node
A hash chain of blocks
Block
Hash value
TX
TX
TX
A crypto-graphic hash function
Block
Hash value
TX
TX
TX

Fig. 1. Overview of Blockchain

condition can generate a new block. The difficulty of the block generation can be adjusted by changing the threshold value. The process of calculating the hash value of the entire block while changing the nonce is called mining. In PoW, each node can generate a new block with a probability proportional to the computing power of the node.

B. Network

The nodes involved in the blockchain form a peer-to-peer network. Transactions and blocks are broadcasted in this peer-to-peer network.

The nodes participating in the network communicate periodically the obtained information of the nodes. The node selects a new neighbor node from this node information when a new neighbor node is required. In Bitcoin's reference implementation Bitcoin Core [2], new connections are generated at a limited occasion, such as when a node joins the network or when an existing neighbor node is disconnected. Therefore, the topology of Bitcoin's network does not significantly change in a short period of time [3].

In simple protocols, block transmission/reception is performed using the protocol shown in Fig. 2. If the node that received the INV message does not have the block, then it responds with a GETDATA message and waits for block reception. By using such protocol, unnecessary transmission of a block with a large data amount does not occur.

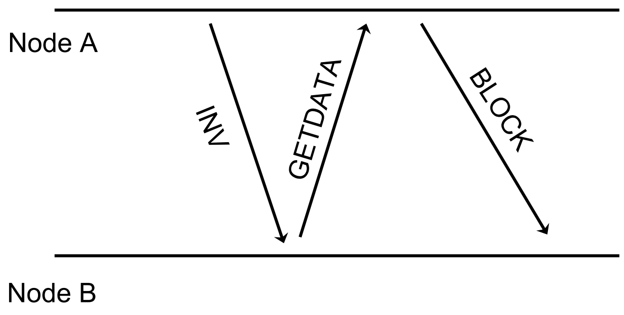

Fig. 2. Protocols when Node A sends block to Node B

C. Fork

A different block is generated prior to the overall propagation of one block to the entire network; hence, two different types of blocks are propagated on the network, which is called a fork. When a fork appears, each node has a different block as the latest block, and the data also lose consistency. To prevent this fork form appearing in existing blockchains, the difficulty of block generation is increased, and the block generation interval is lengthened to ensure that multiple blocks are not simultaneously generated. In the case of Bitcoin, the difficulty of block generation is adjusted to wherein one block to generate every 10 min.

III. Simulator

In this section, we explain the composition of our simulator named SimBlock and evaluate its validity by comparing it with the existing simulator.

A. Design and features

SimBlock is an event-driven simulator wherein each participating node generates the message and mining events. In SimBlock, the following values are used as parameters.

- **Block parameters**
  Block size: The size of the block generated by the node.
  Block generation interval: Block generation interval targeted by the blockchain.

- **Node parameters**
  Number of nodes: The number of nodes involved in the blockchain network.
  Number of neighbor nodes: The number of neighbor nodes of each node.
  Location of the node: The location of each node. Network parameters are determined by the region.
  Block generation capacity: The block generation capacity of each node. The block generation difficulty is obtained from the sum of the block generation capacity of all the nodes and the target of the block generation interval. For example in PoW, block generation capacity is a computing power.

- **Network parameters**
  Network bandwidth: The upstream and downstream bandwidths for each region. The bandwidth when sending messages from Region A to Region B is regarded as the minimum value of Region A's upstream bandwidth and Region B's downstream bandwidth.
  Network propagation delay: The average value of propagation delay between regions. By using this average value, the value according to the Pareto dispersion with a dispersion of 20% is regarded as the propagation delay.

To calculate the arrival time of a message, we use two parameters as follows: the propagation delay between the nodes and bandwidth. The transmission time is determined by the message size and bandwidth between the regions, and the message reception event is derived from the total time of the transmission time and propagation delay from the transmission event of the message. In this simulator, compared with the block message, the other messages are sufficiently small, and the message size is simulated as 0 byte.

When the actual mining is reproduced, the calculation time is high and the number of nodes to be simulated cannot be increased. Therefore, in the proposed simulator, no algorithm, such as actual hash calculation, is performed. The time when the mining becomes successful is calculated from the sum of the block generation capacity of all the nodes and block generation difficulty. In actual blockchains, each node individually determines the block generation difficulty from the generation interval of previous several blocks. However, considering that the difficulty of all the nodes is the same in usual cases, this simulator uniquely assigns the difficulty to all nodes. Each node calculates the success time distribution from its block generation capacity and block generation difficulty. Further, we simulate the time when the mining becomes successful by deriving a random number that follows that distribution. Therefore, the identification of the target block generation interval and block generation capacity of each node, it can be applied not only to PoW but also to various consensus algorithms. In the case of PoW, random number are generated by obtaining the geometric distribution from the difficulty of block generation and the hash rate of the node.

This simulator has the class for managing neighbor nodes. When a node becomes a target of transmission/reception in each message event or when a node succeeds in generating a block, the corresponding function of the adjacent node management class is called. Hence, modifying the neighbor node selection algorithm by changing this function becomes possible.

B. Evaluation

As an evaluation of the simulator, a comparative experiment was performed with the same condition as the existing simulator introduced by Grevais et al. [4]. Their simulator was designed mainly to evaluate its durability against double spoofing attack when modifying the block size and block generation interval. Therefore, changing the parameters such as block size is easy. However, altering the consensus algorithm of the node or the algorithm related to the network topology is challenging. Moreover, given that many of the actual blockchain protocols are reproduced, to change the algorithm of the node, adding all the algorithms that correspond to the existing protocol is necessary.

In the experiment, when reproducing the parameters used in the simulator of Grevais et al., we confirm the same simulation result. We reproduced the actual environment of Bitcoin, Litecoin, and Dogecoin, and evaluated the occurrence rate of the fork and time until the generated block reaches half of the nodes involved in the network. The parameters in Table I were reproduced similar to that of Grevais et al. 's simulator. The number of nodes, block size, and distribution area of the nodes is the result of Grevais et al.'s observation of the actual blockchain in 2015. We set up six regions, namely, Europe, North America, Asia, Australia, Japan, and South America, and the bandwidth and propagation delay of each region were set by reproducing the network at the time. The nodes were distributed in these six regions based on the actual data, and the bandwidth and propagation delay between the regions were applied to the nodes. The propagation delay followed a Pareto distribution with a mean value based on actual data and variance of 20%. The number of connection distribution of the nodes is based on the observation of Miller et al. [3]. Each node randomly selected the nodes from the entire network and set it as its own neighbor nodes. However, the distribution of the nodes' block generation capacity in the actual environment cannot be measured. In this experiment, the normal distribution with a standard deviation of one-third of the average was regarded as the distribution of the block generation capacity.

TABLE I
PARAMETERS OF GREVAIS ET AL.'S SIMULATOR

Parameter	Bitcoin	Litecoin	Dogecoin
# of the nodes	6000	800	600
Block interval	10 min	2 min 30 sec	1 min
Block size	534 KiB	6.11 KiB	8 KiB
# of the connection	Distribution according to Miller et al. [3]		
Geographical distribution	Distribution according to actual blockchains		
Bandwidth propagation delay	6 regional bandwidth and propagation delay		

We simulated it until 10,000 blocks were generated. The result of SimBlock and that of Grevais et al. and real data is shown in Table II.

Every result is close to the measured value and that of Grevais et al., and the proposed simulator can simulate the blockchains with good accuracy. Specifically, SimBlock can simulate values that are very close to that of Grevais et al. who used the same network parameters. Thus, a more accurate simulation of the actual value is possible by reviewing the parameters. Only the rate of the fork in Bitcoin obtained a large error because Bitcoin uses a relay network that is not reproduced in the experiment.

TABLE II
MEDIAN BLOCK PROPAGATION TIME (t_{MBP}) AND RATE OF THE FORK (r_f) IN ACTUAL NETWORKS AND SIMULATOR.

	Bitcoin	Litecoin	Dogecoin
Block interval	10 min	2.5 min	1 min
Measured t_{MBP}	8.7 s	1.02 s	0.98 s
Grevais et al. t_{MBP}	9.42 s	0.86 s	0.83 s
SimBlock t_{MBP}	8.94 s	0.85 s	0.82 s
Measured r_f	0.41 %	0.27 %	0.62 %
Grevais et al. r_f	1.85 %	0.24 %	0.79 %
SimBlock r_f	0.58 %	0.30 %	0.80 %

IV. Application Example of the Simulator

For the application examples of the simulator, we performed an experiment where the neighbor node selection algorithm and experiment in which the participation rate in the relay network were modified.

A. Purpose

One of the problems of existing blockchains is low transaction throughput. The throughput of the blockchain is the quotient obtained by dividing the number of transactions included in the block by the block generation interval. In the case of Bitcoin, the upper limit of the number of transactions included in one block is approximately 4000, and the block generation interval is 10 min. Therefore, the upper limit of the throughput is approximately seven transactions per second. This throughput is very small compared with approximately 1700 transactions per second [5], which is the average throughput of Visa, and approximately 290 transactions per second, which is the average throughput of PayPal [6]. One technique to address this problem is reducing the block propagation time. By shortening the propagation delay, we can safely reduce the block generation interval and improve the throughput [7]. We provide two examples of methods that can reduce the block propagation time and simulate blockchains by adopting these methods.

The first method is attempting to improve the efficiency of the network topology. As mentioned in Section II, a blockchain network is peer-to-peer without a central administrator. Therefore, the topology of the network is dependent on the manner of selecting the neighbor nodes of individual nodes.

The second one is preparing a block propagation-dedicated network different from the network used in the blockchain. As existing research, bloXroute [8] and Falcon [9] proposed a block propagation-dedicated network and attempted to improve the block propagation efficiency. To measure the effect of such a relay network, we observe the propagation time while changing the participation rate of the node to the relay network on the simulator.

B. Algorithm of the neighbor node selection

In the proposed algorithm, we designed each node to connect with the node that sent the INV message earlier to it. Each node scores the node that sent the INV message to it and determines the connection priority. Each time an INV message is received, each node records the elapsed time since the block creation time of the INV message. Each time a node receives 10 blocks, each node updates the neighbor node based on the average of the recorded elapsed time. For each node sent the INV message, sets the weighted average of the recorded elapsed time as a score. Up to the maximum number of connections, connections are established as a new neighbor node from the smallest to the higher score. However, to obtain information on the new node, one neighbor node is randomly selected from all the nodes.

We performed the experiments by using the same parameters that of Bitcoin, as discussed in Section III. We compared the median value of the block propagation time in networks composed of nodes with fixed neighbor nodes and a network consisting only of nodes adopting the proposed node selection algorithm. The former reproduces the ordinary Bitcoin node. The graph is shown in Fig.3 where the horizontal axis denotes the number of generated blocks and the vertical axis indicates the median of the block propagation time. The median of the block propagation time is averaged every 100 blocks and then plotted.

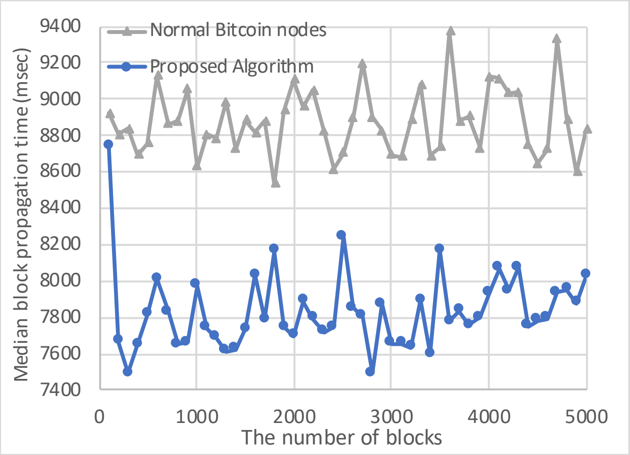

Fig. 3. Median of the block propagation time

The experimental result shows that the block propagation time of the entire network is improved using the proposed neighbor node selection algorithm. It also demonstrates that the propagation time is sufficiently improved in a small number of times by replacing the neighbor nodes. However, no improvement in the propagation time is further observed after 100 blocks are generated. In the proposed algorithm of this experiment, the number randomly selected as an adjacent node from all the nodes is set to 1. However, by changing this number, the improved speed and limit of the propagation time may change.

As observed in the figure, the propagation time in the nodes with smaller calculation power is larger than the propagation time in all the nodes. This observation seems to be because the nodes with large calculation power transitions to the center of the network as the result of the neighbor node selection. Although we could prove that the proposed algorithm can improve the block propagation time, we will also investigate the effect of such network bias on the security and benefits of individual nodes in the future.

C. Relay network

Constructing a relay network for block distribution is proposed as the method used in reducing the block propagation time in blockchains. The block propagation mechanism of the blockchains operates even in an environment where malicious nodes exist, but it is not always suitable when focusing on propagation efficiency. The relay network is a network for block propagation developed outside the mechanism of

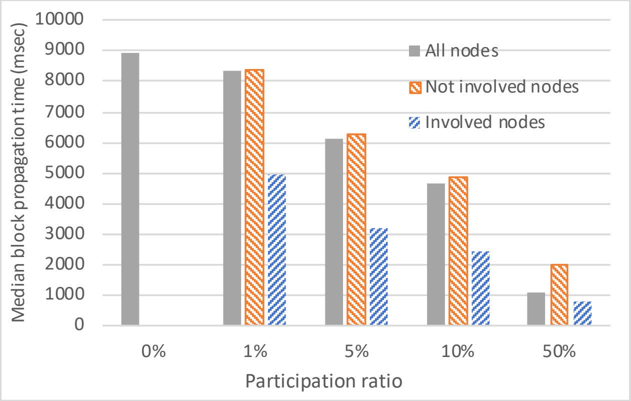

Fig. 4. Median of the block propagation time

blockchains, and sending and receiving blocks among the participating nodes earlier than the normal block propagation in blockchains are possible.

The mechanism for improving the efficiency of the block propagation in a relay network varies depending on the implementation, but we conceptualized this mechanism in the experiment. The nodes involved in the relay network are assumed to transmit the block by using the usual 10 times the bandwidth to the other nodes participating in the relay network. While changing the proportion of the nodes involved in the relay network, the median value of the block propagation time was measured. The median value of the propagation time was measured for the three types of node groups, namely, all the nodes in the blockchain, nodes involved in the relay network, and nodes not involved in the relay network. The parameters were set under the same conditions as that of Bitcoin, as discussed in Section III.

Even if the participation rate in the relay network is as low as 5%, the propagation time of all nodes is remarkably improved to less than 70% of the original propagation time. The propagation time among the nodes participating in the relay network is further improved remarkably. As the involved rate in the relay network increases, the difference in the propagation efficiency between the non-participating and participating nodes increases.

V. Summary and Future Work

In this study, we proposed a blockchain simulator named as SimBlock. We confirmed that the simulator could simulate an actual blockchain with good accuracy. We also presented the techniques in using the simulator and demonstrated that the simulator is useful in research.

Future work includes further expansion of the simulator. Although the current simulator simulates a simple block transmission protocol, we plan to add an implementation that also supports the latest transmission protocol, such as the compact block. We also want to include a simulation of transactions that have not been conducted this time. We believe that some mechanisms used for sending and receiving blocks can be partially added by sending and receiving transactions.

In a few months, the simulator will be released on the website after setting up the input method of the experimental scenario and data output method.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Numbers 16K12406, New Energy and Industrial Technology Development Organization (NEDO), and SECOM Science and Technology Foundation.

References

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[2] Bitcoin Core : Bitcoin. https://bitcoincore.org, (accessed Jan. 10, 2019).

[3] Andrew Miller, James Litton, Andrew Pachulski, Neal Gupta, Dave Levin, Neil Spring, and Bobby Bhattacharjee. Discovering bitcoin ’ s public topology and influential nodes. 2015.

[4] Arthur Gervais, Ghassan O Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 3–16. ACM, 2016.

[5] VisaNet — Electronic Payments Network. https://usa.visa.com/about-visa/visanet.html, (accessed Jan. 10, 2019).

[6] PayPal, Inc — PayPal Reports Second Quarter 2018 Results. https://investor.paypal-corp.com/news-releases/news-release-details/paypal-reports-second-quarter-2018-results?ReleaseID=1072972, (accessed Jan. 10, 2019).

[7] Christian Decker and Roger Wattenhofer. Information propagation in the bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–10. IEEE, 2013.

[8] Uri Klarman, Soumya Basu, Aleksandar Kuzmanovic, and Emin Gün Sirer. bloXroute: A scalable trustless blockchain distribution network whitepaper.

[9] Falcon - A Fast Bitcoin Backbone. https://www.falcon-net.org/, (accessed Jan. 10, 2019).