Blockchain Network Studies Enabled by SimBlock

Kazuyuki Shudo
Kyoto University

Tsuyoshi Hasegawa
Kyoto University

Akira Sakurai
Tokyo Institute of Technology

Ryohei Banno
Kogakuin University

***Abstract*—A blockchain network simulator SimBlock was developed to enable large-scale blockchain researches. Since it was released in June 2019, it has facilitated number of studies on blockchain performance and security. We demonstrate how such a simulator enables blockchain studies by showing experiments running on SimBlock.**

I. Introduction

A public blockchain is a large-scale decentralized distributed system operating over Internet. Studies on the network aspects of blockchain require a simulator. Either operating thousands of nodes or implementing a newly proposed technique to them are nearly impossible.

We developed a blockchain network simulator SimBlock [1], [2] and made it publicly available in June 2019. Since the release of SimBlock, it has facilitated number of blockchain researches by us as well as others. The main topic has been block propagation performance and how to link it to transaction throughput. Security has also been an important topic because it is closely related to performance of a blockchain.

In this demonstration, we show experiments for our studies running on SimBlock. The audience understand how the simulator facilitates blockchain studies.

II. SimBlock: A blockchain network simulator

SimBlock is an event-driven simulator that simulates a blockchain network wherein tens of thousands of nodes relay blocks over Internet. The followings describe Simblock.

- Simulating Internet
  - SimBlock treats Internet divided into six regions, Europe, North America, Asia, Australia, Japan, and South Africa.
  - The communication time is calculated based on the bandwidth and latency between two nodes. The bandwidth and latency depend on the regions where the sender and the receiver reside. The current SimBlock provides network parameters of Internet as of 2015 and 2019.
- Simulating blockchain nodes
  - A node generates and broadcasts a block at timings determined by the algorithm. If nodes run PoW, one of all the nodes generates a block once per about defined block interval.
  - A node relays a received block.

This work was supported by JSPS KAKENHI Grant Number JP21H04872.

978-8-3503-1019-1/23/$31.00 ©2023 IEEE

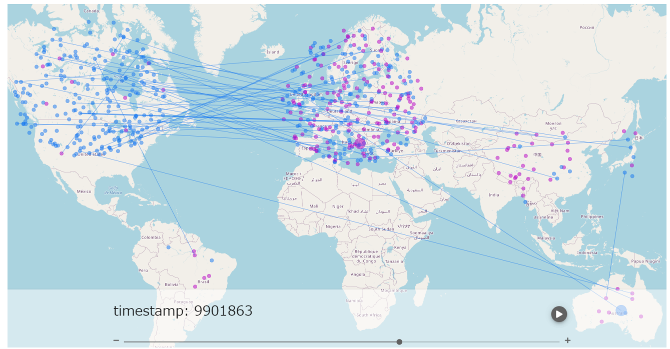

Fig. 1. SimBlock visualizer.

Fig. 1 is a screenshot of the Simblock visualizer. It reads a simulation log and shows a movie in which nodes communicate with each other. Though results of a study are found in a simulation log, such visualization allows us to confirm expected execution of an experience and it gives insights into experiments.

III. Research enabled by SimBlock

The followings are studies we conducted utilizing SimBlock.

- Improvement of SimBlock itself
  - Implementation of Compact Block Relay and evaluation of its impacts [3]
- Block propagation performance
  - Neighbor selection [4]
  - Evaluation of relay networks [5], [6]
  - Delivery tree [7], and delivery trees [8]
  - Sender switchover [9]
  - Block generation notification with a bloom filter [10]
- Transaction throughput
  - Block interval adjustment while keeping fairness [11]
  - Block interval adjustment for transaction throughput [12]
- Security
  - Simulation of selfish mining [13]
  - Saving attacks to Ethereum-like PoS blockchains [14]
  - Precise theoretical fork rate considering hash rate [15]

Proc. 2023 IEEE Int'l Conf. on Blockchain and Cryptocurrency
(IEEE ICBC 2023), May 2023

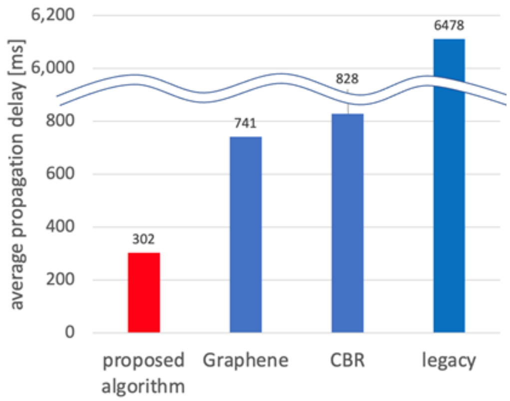

Fig. 2. Results of experiments enabled by SimBlock.

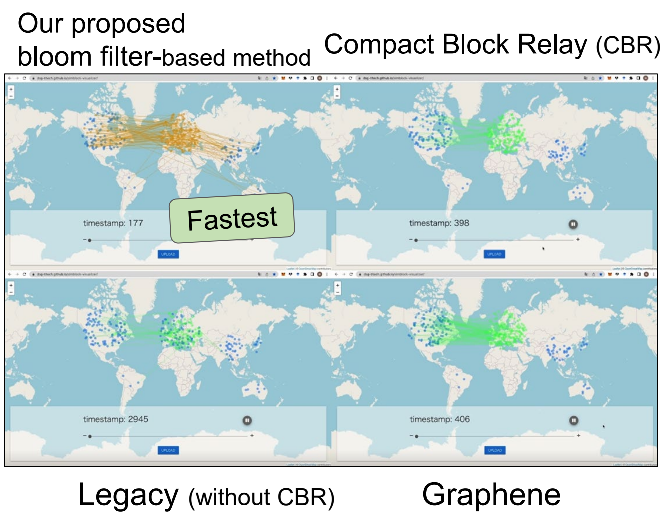

Fig. 3. Experiments visualized.

The so-called scalability problem of blockchains is shortage of transaction throughput. It is famous that initial Bitcoin provided only 7 TPS. Naive solutions to the problem are increasing block size and decreasing block interval. But they sacrifice security by increasing fork rate [16].

Increasing transaction throughput without sacrificing security requires faster block propagation. It has been one of our research topics. Our latest proposal is block generation notification with a bloom filter [10]. Fig. 2 shows the effect of the technique. It achieves a propagation delay of 302 millisecond, less than half that of state-of-the-art methods. SimBlock enabled the experiments and its visualizer demonstrates the differences of propagation performance as shown in Fig. 3.

Faster block propagation itself does not improve transaction throughput. It rather enables shorter block interval or larger block size without sacrificing security. So we next worked on shortening the block interval. The first work focuses on keeping fairness [11], and the next work focuses on keeping security [12].

SimBlock also enabled security studies [13], [14], [15]. Ethereum-like PoS blockchains do not care about block propagation performance as far as messages reach all the nodes within the time slot. But, once this premise is broken due to network troubles or attacks, convergence among multiple nodes becomes a problem. SimBlock could contribute to a study of such a situation [14].

IV. Conclusion

A blockchain network simulator SimBlock has enabled number of studies from performance to security.

The current SimBlock deals with blocks, not transactions. Simulation of transactions requires introduction of a transaction generation model, in which each node broadcasts transactions in its own frequency. Another possible improvement is consideration of Tor. In case of today's Bitcoin, over a half of nodes communicate over Tor [17]. Tor will make communication slow much and Bitcoin simulation could consider it.

References

[1] Y. Aoki, K. Otsuki, T. Kaneko, R. Banno, and K. Shudo, “SimBlock: A blockchain network simulator,” in *Proc. IEEE INFOCOM 2019 Workshops (CryBlock 2019)*, Apr. 2019.

[2] R. Banno and K. Shudo, “Simulating a blockchain network with SimBlock,” in *Proc. IEEE ICBC 2019*, May 2019, pp. 3–4.

[3] R. Nagayama, R. Banno, and K. Shudo, “Identifying impacts of protocol and Internet development on the Bitcoin network,” in *Proc. IEEE ISCC 2020*, Jul. 2020, pp. 506–510.

[4] Y. Aoki and K. Shudo, “Proximity neighbor selection in blockchain networks,” in *Proc. IEEE Blockchain 2019*, Jul. 2019, pp. 52–58.

[5] K. Otsuki, Y. Aoki, R. Banno, and K. Shudo, “Effects of a simple relay network on the Bitcoin network,” in *Proc. AINTEC 2019*, Aug. 2019, pp. 41–46.

[6] K. Otsuki, R. Banno, and K. Shudo, “Quantitatively analyzing relay networks in Bitcoin,” in *Proc. IEEE Blockchain 2020*, Nov. 2020, pp. 214–220.

[7] Y. Kitagawa, K. Shudo, O. Mizuno, and R. Banno, “Verification of applying Plumtree algorithm for blockchain networks,” in *Proc. ICETC 2021*, Dec. 2021.

[8] ——, “Simulation evaluation of multiple Plumtree application in blockchain networks,” in *IEICE technical report NS2022*, Mar. 2023, (unrefereed, and in Japanese).

[9] A. Sakurai and K. Shudo, “Accelerating block propagation by sender switchover in a blockchain,” in *IEICE technical report IA2021-67*, Mar. 2021, pp. 52–57, (unrefereed, and in Japanese).

[10] T. Hasegawa, A. Sakurai, and K. Shudo, “Optimizing block generation notification using bloom filter in a blockchain,” in *Proc. DEIM 2023*, Mar. 2023, (unrefereed, and in Japanese).

[11] R. Kanda and K. Shudo, “Block interval adjustment toward fair Proof-of-Work blockchains,” in *Proc. ICDE 2020 Workshops (BlockDM 2020)*, Apr. 2020, pp. 1–6.

[12] M. Arakawa and K. Shudo, “Block interval adjustment based on block propagation time in a blockchain,” in *Proc. IEEE Blockchain 2022*, Aug. 2022, pp. 202–207.

[13] R. Nagayama and K. Shudo, “Simulating Ethereum network with SimBlock,” lightning talk, Devcon 5, Ethereum Foundation, Oct. 2019.

[14] K. Otsuki, R. Nakamura, and K. Shudo, “Impact of saving attacks on blockchain consensus,” *IEEE Access*, vol. 9, pp. 133 011–133 022, Sep. 2021.

[15] A. Sakurai and K. Shudo, “Impact of the hash rate on the theoretical fork rate of blockchain,” in *Proc. IEEE ICCE 2023*, Jan. 2023.

[16] Y. Sompolinsky and A. Zohar, “Secure high-rate transaction in Bitcoin,” in *Proc. FC 2015*, Jan. 2015, pp. 507–527.

[17] “Bitnodes,” https://bitnodes.io (accessed Apr. 4, 2023).