Accelerating Block Propagation with Sender Switchover in a Blockchain

Akira Sakurai
Tokyo Institute of Technology
Tokyo, Japan

Kazuyuki Shudo
Kyoto University
Kyoto, Japan

***Abstract*—In a public blockchain, the block propagation time has a significant impact on the performance, security and fairness of mining. Reducing the propagation time can increase the transaction processing performance, reduce the fork rate, and increase security. We propose a method to improve block propagation with block sender switchover, even if a node is receiving a block. The method is not vulnerable to eclipse attacks because the neighboring nodes are not changed. Our simulation shows that the proposed method improves the 90th percentile value of the propagation time by up to 18% and the fork rate by up to 7.9%.**

***Index Terms*—blockchain, peer-to-peer, block propagation time, sender switchover**

I. Introduction

Since the cryptocurrency Bitcoin [1] was proposed in 2008, its underlying technology, the blockchain, has been used in numerous other cryptocurrencies, including Ethereum [2].

The block propagation time has a considerable influence on security and the fairness of mining. Reducing the block propagation time can reduce the fork rate, which results in enhanced security [3] [4]. It is also known that small propagation time leads to fairness in mining [5].

In addition, block propagation time is indirectly related to transaction processing perfomance. Bitcoin is currently said to process approximately 27 transactions per second [6]. For Bitcoin to be used more widely in the future, it is necessary to increase the transaction processing perfomance of the entire system. However, naive approaches such as increasing the block size and shortening the block generation interval will increase the fork rate and then compromise security. Therefore, we need to shorten the block propagation time to offset the loss of security.

Shortening the block propagation time could provide the abovementioned advantages, and many studies have addressed this problem. Specifically, neighbor selection [7] and relay network [8] [9] have been proposed to reduce the propagation time. However, neighbor selection makes a system vulnerable to eclipse attacks [10] [11].

In this paper, we propose a method in which a node switches from a block sending node to another node, even when receiving a block. The method shortens the block propagation time in the blockchain network. In addition, since this method does not require the selection of specific nodes, it is less susceptible to eclipse attacks than are traditional methods.

This work was supported by JSPS KAKENHI Grant Number JP21H04872.

Sections 2 and 3 present the study background and related work. Section 4 describes the proposed method sender switchover. In Section 5, the evaluation results are presented. A summary of the paper is given in Section 6.

II. Background

In this section, the background knowledge necessary to understand the paper is presented.

A. Block propagation time and security

In a blockchain, nodes generate blocks at regular intervals on average. After a block is generated, it is broadcast to the blockchain network. The block is spread throughout the network via repeated forwarding, but this process takes time. This time is called the block propagation time. As shown below, the block propagation time has a strong relationship with the blockchain fork rate and affects the blockchain in various ways. Notably, the block propagation time has a considerable influence on a blockchain system.

A fork is a branch of a blockchain. Such a fork can be caused by a malicious node attempting to tamper with past data, or by a large block propagation time even if the system is composed of nonmalicious nodes [4]. In other words, the larger the block propagation time is, the larger the fork rate is, and vice versa. A fork causes inconsistency in the system. A fork also wastes the computational resources of normal miners and increases the impact of various attacks [3]. Prior research suggests that a high forking rate also undermines the fairness of the mining process [5].

The block propagation time also indirectly affects the transaction processing performance of a blockchain. Currently, Bitcoin has lower transaction processing performance than other nonblockchain currencies. Therefore, it is considered simple to increase the transaction processing performance of Bitcoin by increasing the block capacity and shortening the block generation time. However, these solutions increase the fork rate.

In addition, it will take more time to propagate a block as the number of nodes participating in the blockchain system increases. As a result, the fork rate will increase, and the fairness of mining will diminish.

From the above perspectives, it is important to reduce the block propagation time to decrease the fork rate.

Proc. 28th IEEE Symposium on Computers and Communications
(IEEE ISCC 2023), July 2023

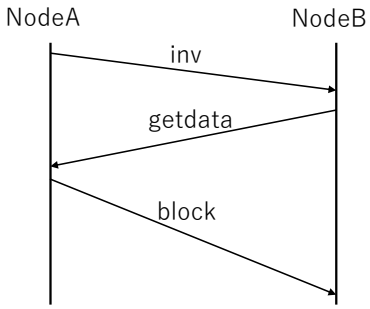

Fig. 1. Block propagation from node A to node B.

B. Compact block relay

Compact block relay, CBR, was proposed in BIP 152 [12] to reduce the bandwidth used for block transmission. In CBR, it is possible to decrease the bandwidth used for block transmission by sending a compact block containing the block header and the associated transaction identification information. If the block cannot be reconstructed after receiving the compact block, a node request is sent for the missing transaction information.

Bitcoin has shortened the block propagation time by introducing CBR.

III. Related work

There are several methods that aim to shorten the block propagation time by selecting neighboring nodes for each node. Some specific methods include Aoki et al.'s method based on past block transmission records [7] and Matsuura et al.'s method based on the area of neighboring nodes [13]. Our proposed method can coexist with these neighbor selection methods, and improve the propagation time furthermore. Although these methods are effective to some extent, since each node independently evaluates its neighboring nodes, an attacker may intentionally become a neighbor node. Thus, the system will become vulnerable to attacks, such as eclipse attacks.

The relay network approach can also reduce the block propagation time [14] by providing a network that can efficiently perform block propagation.

IV. Proposed method

Before explaining the proposed method, we explain the communication protocol for block transmission in Bitcoin. Blocks are sent and received as shown in Figure 1. First, the node that received or generated a block sends an inv message to its neighbor nodes if the block is valid, informing them of the hash value of the block. After that, if the node that receives the inv message has not received the block, it sends back a getdata message to request the block. The node that receives the getdata message sends the requested block.

Next, we explain the block sender switchover process, which is the method proposed in this paper. Figure 2 shows the process of block sender switchover when CBR is not used.

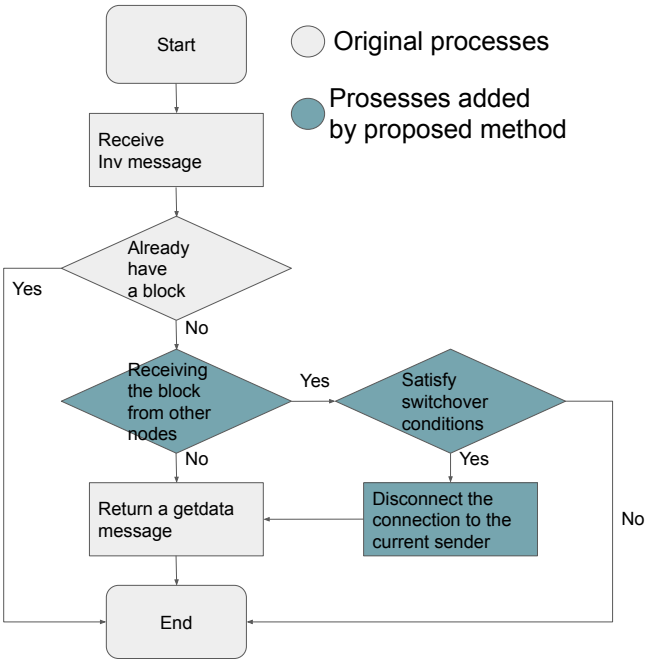

Fig. 2. Switchover of block sender.

If the conditions are met when the `inv` message is received, the node receiving the block switches the block sender. The node receiving the block disconnects from the node that sent the `inv` message in advance to avoid wasting bandwidth. Factors that may result in a switch of the block sender include the relevant geographic information, the number of times the block sender is switched, and the delay in the reception time of the block. When using CBR, in the case of switchover when a compact block has already arrived, the `getblocktxn` message is sent to the switched node without requesting the compact block from the switched node.

Nodes that send `inv` messages earlier are considered to be sending blocks to more neighboring nodes. Receiving blocks preferentially from nodes that send `inv` messages later distributes the transmission burden among nodes, resulting in a reduction in the block transmission time. Consequently, the block propagation time can be shortened by switchover.

When this method is applied, a node does not evaluate or select neighboring nodes, as in methods such as neighbor selection; therefore, it is resistant to eclipse attacks.

We consider several possible conditions for the selection of the block sender. In this paper, we propose a method of switchover mainly by considering the region to which each node belongs. In this method, it is necessary to know the area of the neighboring node to some extent based on the available node information. For example, we can obtain the regional information for the neighboring nodes from their IP addresses.The specific method is as follows. First, an `inv` message is sent. Next, the area of the node that sent the `inv` message and the area of the node that is currently sending the block are compared and switched based on the applied algorithm. Various switchover conditions can be considered at

this time. First, it can be estimated that the block transmission delay is large if the distance from the current block transmission node is large. Additionally, it is possible to examine the area of the node that newly sent the `inv` message and switchover to it if the distance is short.

A. Challenges and possible solutions

There are a few trade-offs to the proposed method. If an attacker frequently sends `inv` messages, the neighboring nodes will never receive blocks. One solution to this problem is to set a specific limit, such as a limit on the number of times the node can switchover. In addition, countermeasures against DoS attacks are effective.

In addition, switchover discards the block information that has already been sent, and by starting from scratch, extra communication is needed; bandwidth is consumed accordingly. Moreover, this process could increase the block transmission time. In this paper, we evaluate the increase in the communication volume due to switchover.

B. Adaptability

First, in switchover, the block transmission by the block sender before switchover must be interrupted. It is possible to break the connection in the transport layer or introduce a new message, such as a cancel message, to inform the switchover. The former is easy to implement in actual applications. In the simulation experiment conducted in this paper, the cancel message is introduced from the implementation point of view. Currently, Bitcoin and Ethereum do not provide such a message, so it is necessary to specify this step when applying the proposed method.

V. Evaluation

We conducted an experiment based on simulations. We used SimBlock, which can simulate the Bitcoin system in 2019 [15] [16], for the experiment. Each node had up to 8 outbound connections and a maximum of 125 inbound connections. In SimBlock, each node belongs to 6 different areas. Let the distribution of hash rates be a normal distribution with an average of $4 \cdot 10^5$(/sec) and a standard deviation of $1 \cdot 10^5$(/sec). And, we set the block generation interval to 10 minutes. In addition, SimBlock uses only low-bandwidth mode simulation for CBR, so experiments were conducted with low-bandwidth mode only. SimBlock is designed to consider the percentage of churn nodes and their impact on the network, as well as the percentage of nodes involved in CBR [16] [17]. All experiments were performed up to a block height of 5000. As mentioned above, from an implementation perspective, the experiment was conducted by introducing a cancel message instead of breaking the connection in the transport layer. In this experiment, the block sender before switchover and the block sender node after switchover may temporarily transmit blocks at the same time when switchover occurs, but the effect is so small that it does not affect the experimental results.

We compared the switchover results under six conditions. Table I, Table II and Figure 3 show the average block propagation time, the 90%ile block propagation time, and the average amount of traffic per block for each node. The top is the case without switchover. The second from the top shows the case of unconditional switchover. The third from the top switchover result shows the case of switchover from outside the same area to inside the same area. The fourth from the top involves the case of switchover from inside the same area to inside the same area in addition to the conditions in the top case. Similarly, the result second from the bottom includes the case of switchover from outside the same area to outside the same area in addition to the conditions for the top case. The bottom result shows except when switchover occurs from inside the same area to outside the same area. Table I shows that the block propagation time is the smallest for the result that is second from the bottom.

TABLE I
VARIOUS REGION-BASED SWITCHOVER (WITHOUT CBR).

	average (msec)	90%ile (msec)	traffic (byte)
without switchover	2807	3806	533371
unconditional	2588	3137	1037188
outside→ inside	2664	3367	598662
outside→ inside or inside→ inside	2627	3260	887696
outside→ inside or outside→ outside	2593	3165	683918
outside→ inside or inside→ inside or outside→ outside	**2585**	**3120**	964735

TABLE II
VARIOUS REGION-BASED SWITCHOVER (WITH CBR).

	average (msec)	90%ile (msec)	traffic (byte)
without switchover	762	1156	49853
unconditional	699	1093	222746
outside→ inside	726	1112	60749
outside→ inside or inside→ inside	698	**1087**	174656
outside→ inside or outside→ outside	716	1108	94609
outside→ inside or inside→ inside or outside→ outside	**693**	**1087**	210713

As seen in the tables and the figure, all the results with switching show an improvement in the average propagation time and the 90%ile propagation time, while the amount of traffic increases when switchover is applied. It can also be seen that the block propagation is fastest when conditions exclude switchover from within the same area to outside the same area. In the case without CBR, the average propagation time improves by a maximum of 7.9% and a minimum of 5.1% compared to the case of no switching. The 90%ile propagation time has improved by a maximum of 18% and a minimum of 12%. The increase in traffic is approximately 18% at the maximum and 12% at the minimum. The average propagation time with CBR is improved by a maximum of 9.1% and a minimum of 4.7% compared to the case without the switchover. The 90%ile propagation time has improved by a maximum of 15% and a minimum of 3.8%. The increase in traffic is approximately 347% at the maximum and 22% at the minimum. The propagation time tends to decrease with increasing traffic regardless of the application of CBR.

In the following part, based on the above results, region-

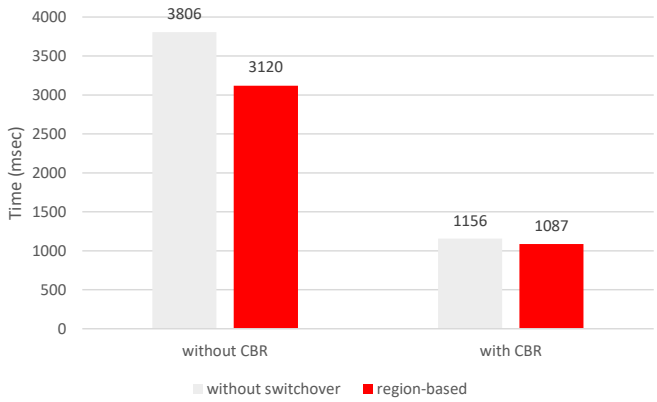

Fig. 3. 90%ile propagation time.

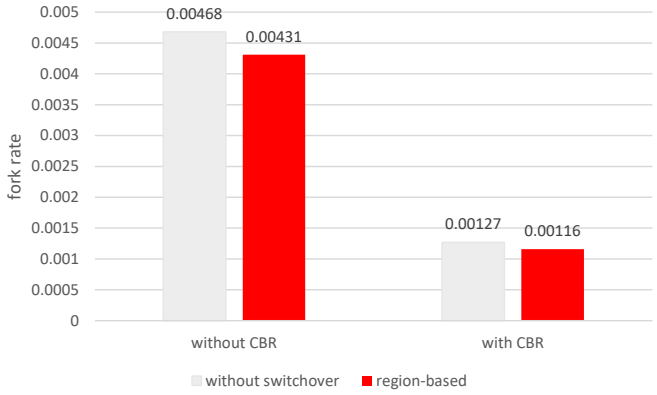

Fig. 4. fork rate.

based switchover refers to the case of switchover excluding switchover from within the same area to outside the same area.

We briefly supplement here the average block propagation time. When the mining success probability of a single hash calculation is sufficiently small, the average block propagation time weighted with hash rate is equal to the product of the fork rate and the average block generation interval. The distribution of hash rate used in this study can also be accurately approximated by the average propagation time instead of the average block propagation time weighted with hash rate [18]. In other words, we can calculate the fork rate by multiplying the average block propagation time by the block generation interval. Figure 4 shows the fork rates of the case without switchover and the case of region-based switchover. As with the average propagation time, applying region-based switchover improves the fork rate by 7.9% without CBR and 9.1% with CBR.

VI. CONCLUSION

In a blockchain network, the block propagation time is important from the perspectives of security, scalability, and fairness of mining. In this paper, we proposed a method to switch block-sending nodes based on conditions. This approach makes it possible to reduce the time it takes for blocks to spread throughout a network. We also experimented with this method using SimBlock. The application of the switchover improves the 90%ile block propagation time by up to 18% and the fork rate by up to 7.9%.

In the future, additional switchover conditions, such as limits on the number of steps and communication time, should be considered. Specifically, all possible conditions should be explored in detail with simulations. In addition, we will confirm the effectiveness of switchover applying the proposed approach with more existing block propagation time countermeasure methods, such as those relay networks.

REFERENCES

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.
[2] “Ethereum: A secure decentralised generalised transaction ledger,” https://ethereum.github.io/yellowpaper/paper.pdf.
[3] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, “On the security and performance of proof of work blockchains,” *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.
[4] C. Decker and R. Wattenhofer, “Information propagation in the bitcoin network,” in *IEEE P2P 2013 Proceedings*, 2013, pp. 1–10.
[5] R. Kanda and K. Shudo, “Block interval adjustment toward fair proof-of-work blockchains,” *2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW)*, pp. 1–6, 2020.
[6] E. Georgiadis, “How many transactions per second can bitcoin really handle ? theoretically,” *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 416, 2019.
[7] Y. Aoki and K. Shudo, “Proximity neighbor selection in blockchain networks,” in *Proc. 2nd IEEE Int’l Conf. on Blockchain (IEEE Blockchain 2019)*, 2019, pp. 52–58.
[8] “Fibre,” https://bitcoinfibre.org/.
[9] U. Klarman, S. S. Basu, A. Kuzmanovic, and E. G. Sirer, “bloxroute: A scalable trustless blockchain distribution network whitepaper,” 2018.
[10] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, “Eclipse attacks on Bitcoin’s Peer-to-Peer network,” in *Proc. 24th USENIX Security Symposium (USENIX Security ’15)*. Washington, D.C.: USENIX Association, Aug. 2015, pp. 129–144. [Online]. Available: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/heilman
[11] M. Walck, K. Wang, and H. S. Kim, “Tendrilstaller: Block delay attack in bitcoin,” in *Proc. 3rd IEEE International Conference on Blockchain*, 2019, pp. 1–9.
[12] “bip-0152.mediawiki,” https://github.com/bitcoin/bips/blob/master/bip-0152.mediawiki, 2022.
[13] H. S. Hiroshi Matsuura, Yoshinori Goto, “Region-based neighbor selection in blockchain networks,” 2021.
[14] K. Otsuki, Y. Aoki, R. Banno, and K. Shudo, “Effects of a simple relay network on the bitcoin network,” in *Proceedings of the Asian Internet Engineering Conference*, ser. AINTEC ’19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 41–46. [Online]. Available: https://doi.org/10.1145/3340422.3343640
[15] Y. Aoki, K. Otsuki, T. Kaneko, R. Banno, and K. Shudo, “Simblock: A blockchain network simulator,” in *Proc. IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM 2019 Workshops)*, 2019, pp. 325–329.
[16] R. Nagayama, R. Banno, and K. Shudo, “Identifying impacts of protocol and internet development on the bitcoin network,” in *Proc. 25th IEEE Symposium on Computers and Communications (IEEE ISCC 2020)*, 2020, pp. 1–6.
[17] M. A. Imtiaz, D. Starobinski, A. Trachtenberg, and N. Younis, “Churn in the bitcoin network: Characterization and impact,” in *Proc. 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC 2019)*, 2019, pp. 431–439.
[18] A. Sakurai and K. Shudo, “Impact of the hash rate on the theoretical fork rate of blockchain,” in *2023 IEEE International Conference on Consumer Electronics (ICCE)*, 2023, pp. 1–4.