Impact of the Hash Rate on the Theoretical Fork Rate of Blockchain

Akira Sakurai
Tokyo Institute of Technology
Tokyo, Japan

Kazuyuki Shudo
Kyoto University
Kyoto, Japan

***Abstract*—Forks in a blockchain make it less secure. So the fork rate can be used as a measure of security. Theoretical fork rates have been used in a number of studies, not all of which consider the distribution of hash rates. In this paper, we propose a theoretical fork rate that considers the distribution of hash rates. We compare this theoretical fork rate that considers the distribution of hash rates, a theoretical fork rate that does not consider the distribution of hash rates, and the real fork rate obtained by simulation experiments. Using a simulator, we show that the proposed theoretical fork rate is closer to the real value.**

***Index Terms*—blockchain, fork rate, block propagation time**

I. Introduction

Blockchain is a distributed ledger that is the underlying technology of cryptocurrencies such as Bitcoin [1].

A fork is a branching of the blockchain. The fork rate is the percentage of forked blocks out of the total number of blocks created. Forks are caused by malicious nodes and block propagation delays [2].For example, a fork occurs when a node successfully generates a block and another node that has not received the block generates another block. A fork divides the computational resources used for proof of work (PoW) in the network, increasing the risk of double spending attacks and selfish mining [3].

In [2], a theoretical fork rate that does not consider the distribution of hash rates is used. This is likely due to the difficulty of knowing the distribution of hash rates in a real blockchain network. However, such a theoretical fork rate may deviate from the real fork rate when the distribution of hash rate is biased. Based on the above, in this paper, we first show that $T_W = TF$ holds in order to define the theoretical fork rate considering the distribution of hash rates, where T_W is the average block propagation time weighted by the hash rate, T is the block generation interval, and F is the fork rate. This once again confirms the strong relationship between fork rate and block propagation time. We also define the theoretical fork rate from this relationship. This theoretical fork rate takes into consideration the distribution of hash rates. In this paper, we compare this theoretical fork rate with a theoretical fork rate that does not consider the distribution of hash rates and the real fork rate. Using a simulator, we show that the theoretical fork rate that takes into account the distribution of hash rates is closer to the real fork rate.

In section II, we show that $T_W = TF$ theoretically. In section III, we show the effectiveness of the theoretical fork rate defined from $T_W = TF$ compared to the theoretical fork rate that does not consider the distribution of hash rates. In section IV, we describe the application of the theoretical fork rate to real blockchain networks and simulation experiments. Section V gives a summary and describes future work. In the appendix, we show through experiments that $T_W = TF$ holds under various conditions, and that the theoretical fork rate defined from $T_W = TF$ is effective as an evaluation criterion for blockchain networks.

II. Theoretical verification

Here we show from a theoretical point of view that $T_W = TF$ holds.

Let p be the probability of generating a valid block in one hash calculation, V be the set of nodes participating in the blockchain network, and $M_i (i \in V)$ be the hash rate of each node i. At this time, the block generation success probability does not depend on the previous hash calculation. Therefore, the probability of successfully generating a block in N hash calculations is $(1-p)^{N-1}p$. Therefore, the average number of hash calculations required to successfully generate a block is $1/p$. From this, if the hash rate of the entire network is $M_{all} = \sum_{i \in V} M_i$, the expected value of the block generation interval of the entire network is $1/(p \cdot M_{all})$; let this be T.

Next, we show the fork rate when node i succeeds in creating a block. $u_i(t)$ is the sum of the hash rates of nodes that have not yet received a block after t units of time since node i successfully generated a block. This $u_i(t)$ takes a different value for each block even if the same i is used, but for the sake of simplicity, we assume that it does not depend on individual blocks. Then, the probability that node i succeeds in creating a block and the block is distributed throughout the network without forks is $(1-p)^{\int_0^\infty u_i(t)dt}$. Therefore, the probability F_i that a fork will occur before the block created by node i spreads throughout the network is as follows.

$$\begin{aligned} F_i &= 1-(1-p)^{\int_0^\infty u_i(t)dt} \\ &\approx p \cdot \int_0^\infty u_i(t)dt \quad (\because p \ll 1) \\ &= \frac{\int_0^\infty \frac{u_i(t)}{M_{all}}dt}{T} \end{aligned} \tag{1}$$

Then, the fork rate is $F = \sum_{i \in V} M_i \cdot F_i / M_{all}$.

Next, we show in more detail the relationship between the block propagation time, specifically the average block

TABLE I
F_R, F_W AND F_N WITH VARIOUS DISTRIBUTIONS OF HASH RATES.

	F_R	F_W	F_W/F_R	F_N	F_N/F_R
Distribution (1)	0.001308	0.001299	0.993	0.001299	0.993
Distribution (2)	0.001318	0.001277	0.969	0.001280	0.971
Distribution (3)	0.001148	0.001128	0.9826	0.001266	1.103
Distribution (4)	0.000990	0.001040	1.051	0.001270	1.283

propagation time weighted by the hash rate, and the fork rate. $T_{W,i}$, the average block propagation time weighted by the hash rate when node i successfully generates a block, is as follows.

$$T_{W,i} = \int_0^\infty -t \cdot \frac{u_i'(t)}{M_{all}} dt$$

$u_i'(t)$ in the equation in the integral on the right side indicates the derivative of $u_i(t)$. Actually, $u_i(t)$ is not differentiable, but here we treat $u_i(t)$ as differentiable for simplicity. Therefore, $-\frac{u_i'(t)}{M_{all}}$ indicates the amount of change in hash rate ratio for the entire network that received the block after t units of time have passed since the block was generated. We obtain the following result by further integration by parts.

$$\begin{aligned} T_{W,i} &= \int_0^\infty -t \cdot \frac{u_i'(t)}{M_{all}} dt \\ &= \frac{\{[-t \cdot u_i(t)]_0^\infty + \int_0^\infty u_i(t) dt\}}{M_{all}} \\ &= \int_0^\infty \frac{u_i(t)}{M_{all}} dt \\ &\approx TF_i \end{aligned} \quad (2)$$

We take the average based on the distribution of hash rates by averaging both sides and obtain the following result.

$$\begin{aligned} T_W &= \frac{\sum_{i \in V} M_i \cdot T_{W,i}}{M_{all}} \\ &\approx \frac{\sum_{i \in V} M_i \cdot T \cdot F_i}{M_{all}} \\ &= TF \end{aligned} \quad (3)$$

What this formula means is that the average block propagation time weighted by the hash rate is equal to the product of the block generation interval and the fork rate. In reality, formula (3) may not hold due to forks or other factors.

III. EXPERIMENT

We define the theoretical fork rate $F_W = T_W/T$ from the relationship shown in section II between the fork rate, the average block propagation time weighted by the hash rate, and the block generation interval. F_W takes into account the distribution of hash rates, as is clear from the definition. On the other hand, we define the theoretical fork rate F_N without considering the distribution of hash rates as follows.

$$F_N = \frac{\int_0^\infty 1 - f(t) dt}{T}$$

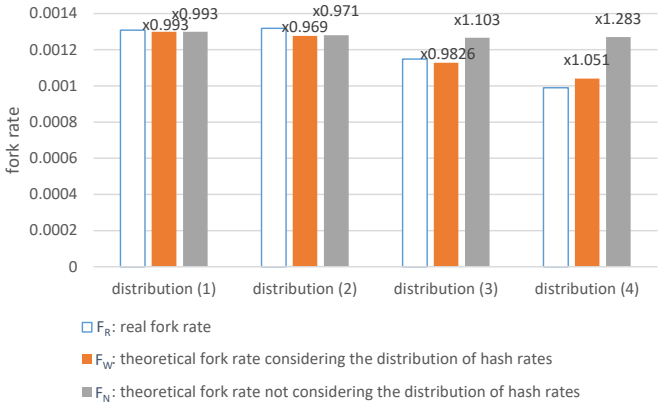

Fig. 1. F_R, F_W and F_N with various distributions of hash rates.

$f(t)$ indicates the percentage of nodes that have received the block t units of time after the block was created. Originally, this variable could take a different value for each block. Here, the mean of the values for each block is treated as $f(t)$. Such a theoretical fork rate that does not consider the distribution of hash rates is used in various studies [2] [4].In this section, we compare the theoretical fork rates F_N and F_W and the real fork rate F_R obtained by conducting simulation experiments with various distributions of hash rates. We show that F_W is closer to the value of F_R than F_N.

We used SimBlock [5] for the simulation. The number of nodes was 1000, the maximum number of inbound connections was 125, and the number of outbound connections was 8, reconnecting every 10 blocks. All experiments were performed with a maximum block height of 100000. The experiments were conducted on the following four distributions of hash rates.

Distribution (1):
Let the hash rate of all nodes be $4 \cdot 10^5$(/sec) uniformly.

Distribution (2):
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).

Distribution (3):
Let the distribution consider the current Bitcoin distribution of hash rates [6]. Assuming that there are mining pools, we simulate nodes with high hash rates. Specifically, we randomly select six nodes and let their hash rates be $1.50 \cdot 10^8$, $1.33 \cdot 10^8$, $9.33 \cdot 10^7$, $8.06 \cdot 10^7$, $5.06 \cdot 10^7$, and $1.26 \cdot 10^7$(/sec). Let the distribution of the hash rates of the other nodes be a normal distribution with an average of $4 \cdot 10^5$(/sec) and a standard deviation of $1 \cdot 10^5$(/sec).

Distribution (4):
Let the distribution consider the current Ethereum [7] distribution of hash rates [8]. As with distribution (3), we assume there are mining pools and simulate nodes with higher hash rates. Specifically, we randomly

select five nodes and let their hash rates be $3.31 \cdot 10^8$, $1.60 \cdot 10^8$, $1.14 \cdot 10^8$, $8.00 \cdot 10^7$, and $5.71 \cdot 10^7$(/sec). Let the distribution of the hash rates of the other nodes be a normal distribution with an average of $4 \cdot 10^5$(/sec) and a standard deviation of $1 \cdot 10^5$(/sec).

Table I and Figure 1 show the experimental results. It can be seen that F_W is close to F_R in all distributions (1) to (4). In the case of distribution (1), F_N is close to F_W because the hash rate follows a uniform distribution. In the case of distribution (2), the hash rate follows a normal distribution, but even in this case there is no significant difference between F_R and F_N. In distributions (3) and (4), the distributions of the hash rates are greatly biased by introducing nodes with high hash rates. In this case, the values of F_R and F_N diverge greatly.

IV. Application

Based on the experimental results in the appendix, we consider applying the theoretical fork rate F_W to real blockchain networks and simulation experiments.

To apply the theoretical fork rate F_W to a real blockchain network, it is necessary to know the distribution of hash rates. Therefore, we plan to estimate the distribution of hash rates from the block generation rate of each node.

From the experimental results in the appendix, when measuring the fork rate of a blockchain network in a simulation experiment, it is necessary to simulate many blocks. On the other hand, by using the theoretical fork rate F_W, it is possible to determine the fork rate of the network quickly and accurately.

V. Conclusion and Future Work

This paper shows that the average block propagation time weighted by the hash rate is equal to the product of the block generation interval and fork rate. Based on this relationship, we defined a new theoretical fork rate. Furthermore, we conducted simulations to show that our theoretical fork rate F_W is closer to the actual fork rate F_R than the theoretical fork rate F_N, which does not consider the hash rate distribution, under various distributions.

In the future, we plan to evaluate block propagation acceleration methods such as neighbor node selection [9] using the theoretical fork rate F_W presented in this work.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *IEEE P2P 2013 Proceedings*, 2013, pp. 1–10.

[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] 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.

[5] 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.

TABLE II
F_W, F_R AND T_W UNDER VARIOUS CONDITIONS.

	F_R	F_W	F_W/F_R	T_W(msec)
2019	0.001318	0.001277	0.968	769
2015	0.001916	0.001845	0.963	1112
Region-based neighbor selection	0.001228	0.001254	1.021	755
Without compact block relay	0.004588	0.004695	1.023	2814

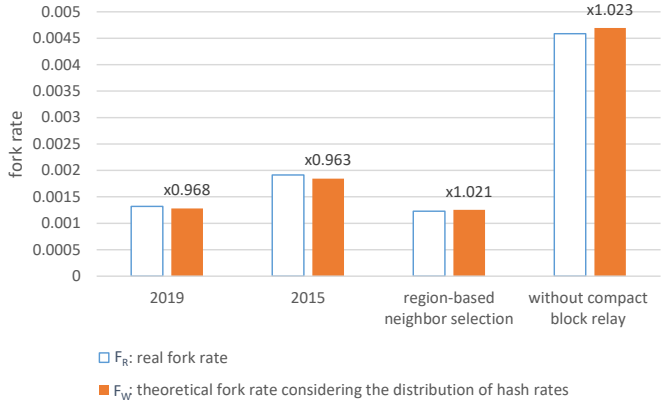

Fig. 2. F_R and F_W under various conditions.

[6] "Blockchain.com," https://www.blockchain.com/pools, 2022, accessed: Aug. 19. 2022.

[7] "Ethereum: A secure decentralised generalised transaction ledger," https://ethereum.github.io/yellowpaper/paper.pdf.

[8] "Blockchair," https://blockchair.com/ja/ethereum/charts/hashrate-distribution, 2022, accessed: Aug. 17. 2022.

[9] 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.

[10] 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.

[11] M. Corallo, "Compact block relay," https://github.com/bitcoin/bips/blob/master/bip-0152.mediawiki, 2016, accessed: Aug. 16. 2022.

[12] H. Matsuura, Y. Goto, and H. Sao, "Region-based neighbor selection in blockchain networks," in *Proc. 2021 IEEE International Conference on Blockchain (IEEE Blockchain 2021)*, 2021, pp. 21–28.

Appendix

Here, we show that the relationship described in II holds even in more complex situations. We also show the effectiveness of the theoretical fork rate F_W on the convergence speed.

As in the experiment above, we used SimBlock with 1000 nodes, 125 inbound connections, and 8 outbound connections, reconnecting every 10 blocks. All experiments were performed with a maximum block height of 100000.

We investigated F_W, F_R and T_W under various conditions. The details of each condition are as follows.

2019:

All nodes in SimBlock belong to one of six regions. We use the 2019 regional bandwidth and latency

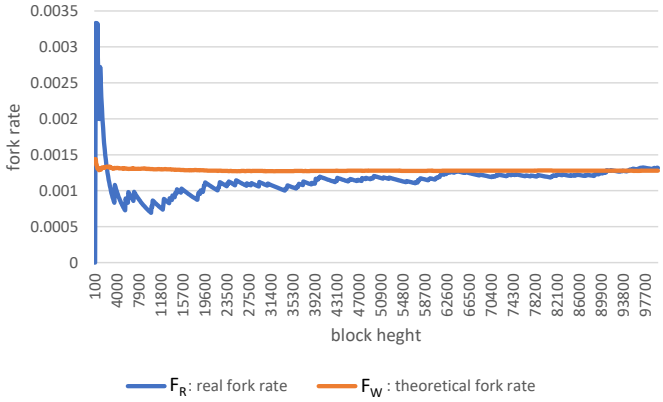

Fig. 3. T_R and T_W per block height.

available in SimBlock [10]. 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). We use compact block relay [11] and choose neighbor nodes randomly.

2015:

We use the 2015 regional bandwidth and latency available in SimBlock. The other conditions are the same as those of 2019.

Region-based neighbor selection:

We apply region-based neighbor selection [12]. With this method, the number of connections between the same regions among the outbound connections of each node is set to 6. The other conditions are the same as those of 2019.

Without compact block relay:

We do not apply a compact block relay. The other conditions are the same as those of 2019.

Table II and Figure 2 show the experimental results. As seen, F_W matches F_R with high accuracy under all conditions. This shows that formula 3 holds even in complicated situations.

The next experiment demonstrates the effectiveness of the proposed theoretical fork rate F_W from the viewpoint of convergence speed. As shown in Figure 3, F_W is faster and more stable than F_R. From this experimental result, we can see the effectiveness of the theoretical fork F_W when evaluating the blockchain network.