Block Interval Adjustment Toward Fair Proof-of-Work Blockchains

Reiki Kanda
Tokyo Institute of Technology
Tokyo, Japan
kanda.r.aa@m.titech.ac.jp

Kazuyuki Shudo
Tokyo Institute of Technology
Tokyo, Japan
shudo@is.titech.ac.jp

***Abstract*—We propose a finer measure of fairness than that found in the literature. We then propose a method of adjusting the block interval that maintains fairness and high transaction throughput. The existing fairness measure is so rough that its judgement is considered fair as long as a node can start mining by the time another node mines a block. However, the time left for mining differs between nodes; this can result in unfairness. Therefore, we define the proposed fairness as the mining success rate of miners depends only on computational power. This is a finer measure of fairness than existing one. We propose a method for adjusting the block interval to maintain a certain level of fairness. Most Proof of Work (PoW) blockchains fix the block intervals and do not consider fairness. In contrast, our proposed technique adjusts the block interval based on the fork incidence rate. Our method maintains fairness even if the network conditions change. Simulations confirm that our technique maintains fairness even when the Internet performance varies.**

***Index Terms*—Bitcoin, Blockchain, Proof of Work, Fairness**

I. Introduction

A number of blockchains have been proposed since Bitcoin appeared in 2008. These blockchains can be classified as permissionless or permissioned blockchains [1]. Separating permissionless from permissioned blockchains depends on whether the confirmation is performed by a limited number of validators. We refer to a permissionless blockchain in which each participant competes by computing to determine a block proposer as Proof of Work (PoW) blockchains.

A PoW blockchain is naturally open to more participants who cannot be trusted. However, participants may be effectively limited despite a permissionless blockchain. In the blockchains, the participants begin the competition for new block proposals while they synchronize the proposed blocks across the network. Therefore, when the synchronization time of the entire network becomes long, computation competition occurs only among the participants who arrive before the block has spread across the entire network. If the block interval is too short, this effectively limits the participants despite the blockchain being permissionless. Croman *et al* defined to the situation in which mining competition is completed before a block reaches all participants and whereby some participants are limited as unfair [2]. On the other hand, long latency and low throughput are suffered in the case of the relatively long block interval of PoW blockchains, such as Bitcoin; however, most participants have sufficient time to participate in the competition. Therefore, it is necessary to set the block interval such that the performance is as high as possible while also ensuring fairness.

We propose a new fairness measure that is more precise than the fairness of Croman *et al*. They noted that there are nodes that cannot receive blocks before the end of the mining competition. However, it is not enough to receive blocks before the end of the competition, and unfairness occurs due to the difference of propagation delay on each node. For example, if the block interval is 15 seconds and a node receives a block in 14 seconds when competing, the node becomes a fair node under the existing fairness definition but becomes an unfair node if the mining competition continues for a long time. We analyze the effects of unfairness due to differences in arrival times on the mining success rates in the long term. With this proposal, we can find the stricter bound of the block interval given by the fairness defined in existing research. A stricter lower bound is useful in setting the block interval appropriately.

In any fairness definition, the lower bound of the block interval varies with the propagation delay. Principal PoW blockchains, such as Bitcoin and Ethereum [3], [4], have fixed block intervals. Therefore, we propose a method to adjust the block interval appropriately according to the network condition. We experimented to adjust the block generation interval with the goal of satisfying the fairness of this proposal sufficiently. From this experiment, we confirm that the block interval adjustment method reaches the target when the propagation delay does not change significantly between block interval updates.

In the next section, we introduce the research background. We then reconsider the lower bound of the block interval in Section III and propose a method for block interval adjustment in Section IV. Section V provides a discussion of the proposed method, and finally, we provide a summary about and conclusions for this research.

II. Background

In this section, we first given a brief overview of PoW blockchains and then provide a discussion on existing research concerning the block interval.

Proc. 2nd Int'l Workshop on Blockchain and Data Management (BlockDM 2020), April 2020

A. PoW blockchains

A PoW blockchain is supported by multiple nodes that organize into a peer-to-peer network. Some of these nodes are miners who generate blocks. When a transaction is issued, it is propagated to the blockchain network and stored in each node's transaction pool. This unconfirmed transaction group waits for processing via PoW, which prevents tampering. PoW is generated by a large number of miners who attempt to create a collection of data called blocks based on certain rules that contain transactions. At this time, multiple miners attempt to generate the same-height block, and other nodes verify that the block generated by a miner is the correct block when relaying. Therefore, blocks that are generated incorrectly are not spread; only correct blocks that are generated earlier are spread. The block is propagated across the blockchain network and added to the blockchain of each node. This blockchain is a history of transactions that occur on the network and is a very large transaction ledger. When generating a new block again, the information obtained from the previous block is also used, and the consistency of the ledger is guaranteed.

B. Difficulty adjustment algorithm

In the case of the PoW blockchain, the block interval is adjusted to be constant. Let us explain how to keep the block interval constant with Bitcoin as an example. When generating a block, the miner repeats hash computations to find the nonce that satisfies the following inequality.

$$hash(\text{nonce, previous block header hash}) < \text{difficulty target}$$

The block interval is kept constant by updating the difficulty over a certain period. The equation for updating the difficulty of the Bitcoin blockchain is as follows [5]:

$$\text{new difficulty} = (\text{old difficulty})\frac{\text{Time for mining 2016 blocks}}{\text{20160 min}}$$

As seen from this equation, the Bitcoin blockchain is such that the block interval is approximately 10 minutes.

C. Block interval

Blocks are not generated unless a certain period of time has passed stochastically by PoW. This is an obstacle to blockchains with respect to processing a large number of transactions faster. To overcome this, it is necessary to shorten the block interval; however, some studies have shown problems with this solution [6], [7]. Gervais *et al* utilized a simulator that increased the fork (orphan) rate on the blockchain by shortening the block interval [7]. It is better to avoid forks since multiple blocks of the same height are generated, dispersing the miner's computational power and making the blockchain vulnerable to 51% attack [6]. In addition, Decker *et al* showed that if the block interval is fixed and if the propagation time of a block is reduced, the fork rate decreases [8]. In other words, it is important to reduce the block propagation delay to shorten the block interval while also avoiding forks. For the purpose of reducing the block propagation delay, there have been various proposals for Graphene [9] to reduce the block size substantially and a mechanism for high-speed relay [10]–[12]. As mentioned above, the block interval and the propagation time are closely related from the viewpoint of security.

On the other hand, the study by Croman *et al* reported that if the propagation delay is determined from the viewpoint of fairness, a lower bound of the block interval is given [2]. The lower bound is based on the idea that it is unfair that the next block is generated before the block reaches X% of the network. This fairness focuses on whether the block reception time of each node exceeds the block interval. However, in PoW, the mining success rate is affected by the time at which each node receives a block; thus, it is necessary to focus on such a difference in reception time. We consider it fair that in the long term, the mining success rate does not depend on the time required to receive a block but rather only depends on the hash rate. Our fairness is a natural extension of the fairness of Croman *et al*.

D. Propagation delay

It has been reported that the propagation delay has decreased in the last three years [13], [14]. It has also been noted that this is because of the effects of Compact Block Relay protocol [15] and relay networks [16], [17]. By improving the propagation delay, the block interval can be shortened, and the performance of the blockchain can be improved. However, the performance of the blockchain has not been improved because the block interval is constant.

Approximately 10 years have passed since the first block of Bitcoin was issued, and miners continue to generate blocks. Considering that people continue to generate blocks for incentive fees, it is realistic that the blockchain will continue to grow over the next 10 or 20 years. Therefore, when someone proposes a new PoW blockchain, it is necessary to design on the assumption that the blockchain will be used for several decades. It is not realistic to assume that the propagation delay will be constant in the long term. Therefore, it is necessary to adjust the block interval flexibly to account for changes in propagation delay.

III. The Lower Bound of Block Interval

In this section, we define fairness from propagation delay and block interval perspectives and compare it with the lower bound of the block interval proposed in existing research. From III-B, we calculate fairness based on Bitcoin's propagation delay.

A. Definition of fairness measurement

We define fair as when each node has a block mining success rate that is based on the hash rate and unfair as when the mining success rate changes depending on the difference in block arrival times. When the block interval is shortened, it is more likely that the difference in the block arrival time will affect the mining success rate. The hash rate considering the difference in arrival times is defined as the effective hash rate. Assuming that the time from which a block is generated until

TABLE I
COMPARISON OF PROPAGATION MEASURED AND ESTIMATED ON THE BITCOIN NETWORK.

propagation delay (msec)	5sec	10sec	15sec
estimated 50% propagation delay	450	697	911
measured 50% propagation delay	528	806	1482
estimated 90% propagation delay	5055	10079	14946
measured 90% propagation delay	4994	9989	14986

TABLE II
FORK RATE AND FAIRNESS.

	$\frac{\text{block interval}}{\text{90\% propagation delay}} = 2$	$\frac{\text{block interval}}{\text{90\% propagation delay}} = 3$
5sec	14.6%	9.8%
10sec	14.2%	9.6%
15sec	14.1%	7.2%

it arrives at a certain node is t_a and that the block interval is t_b, the effective hash rate and hash rate have the following relationship.

$$\text{effective hash rate} = \left(1 - \frac{\text{arrival time}}{\text{block interval}}\right)(\text{hash rate})$$

Let r_h be the hash rate distribution, t_b be the block interval, $f(t)$ be the block arrival distribution at each node, and r_{eff} be the effective hash rate distribution. The effective hash rate distribution is

$$r_{\text{eff}} = \frac{r_h \int_0^{t_b} f(t)\left(1 - \frac{t}{t_b}\right) dt}{\sum_{i=1}^{n} r_{h_i} \int_0^{t_b} f_i(t)\left(1 - \frac{t}{t_b}\right) dt} \quad (1)$$

r_{eff} means the hash rate distribution when there is no difference between block arrival times. For a sufficiently long blockchain, the mining success rate r_m satisfies

$$r_{\text{eff}} \simeq r_m \quad (2)$$

To consider propagation delays, the block interval, and fairness, we evaluate fairness by the difference between the hash rate and effective hash rate. Then, we formulate the fairness as follows. For a sufficiently long blockchain, let $V = \{v_1, v_2, ...v_n\}$ be the set of nodes, and let X be the ratio of fair nodes. X is

$$X = \frac{|\{v_i \in V \mid |r_{m_i} - r_{h_i}| \leq \delta_i\}|}{|V|} \quad (3)$$

$$\simeq \frac{|\{v_i \in V \mid |r_{\text{eff}_i} - r_{h_i}| \leq \delta_i\}|}{|V|} \quad (4)$$

However, δ satisfies $Pr\{|r_m(i) - r_h(i)| \geqq \delta\} \leq \epsilon$. In a fair case, we define the error of r_m and r_h to converge to r_m in probability by the central limit theorem. In addition, we define (X, ϵ) fairness as the case when specifying X and ϵ. In this study, we calculate these values as $X = 0.9, \epsilon = 0.01$. This means that 90% of nodes are unfair nodes with probability 1% or less for a sufficiently long blockchain.

B. Computing fairness in the case of Bitcoin

The time from when a block is mined to when it is received with distribution $f_i(t)$ for each node i is required to calculate the effective hash rate. Therefore, we show how to generate $f_i(t)$ using data from Bitcoin Monitoring [18] for the period from June 2018 to November 2019. $f_i(t)$ follows a gamma distribution, and its parameter is obtained by maximum likelihood estimation. The actual hash rate is calculated from the estimated $f_i(t)$. We calculated the 90 % propagation delay and 50 % propagation delay from the estimated $f_i(t)$ and show them in Table I to compare them with the measured values. X % propagation delay means the time that it takes for a block to reach a node with X % probability.

The current 90% block propagation delay of Bitcoin is approximately 5 sec [18]. To consider cases where the propagation delay varies, we show how to calculate $f_i(t)$ when the 90% propagation delay is 10 seconds and 15 seconds. The data are divided into three parts, and the parameter of each $f_i(t)$ is estimated. The three estimated parameters are linearly regressed, and the values on the straight line are candidates for the parameters of $f_i(t)$. We select values from these candidates that result in 90% propagation delays of 10 seconds and 15 seconds. We show in Table I the correspondence between the 90 % propagation delay and the 50 % propagation delay measured between January 2018 and November 2019 to confirm that the estimated $f_i(t)$ is valid [18]. For example, if the 90 % propagation delay is 10 seconds, this is the average of the data when the 90 % propagation delay is approximately 10 seconds in the above period.

Figure 1 shows the ratio of nodes with a fair mining success rate from the calculation results from equation 4 assuming that the block height is 100000 and that the number of nodes is 10000. When the block interval is set to two to three times the 90% propagation delay, the ratio of fair nodes exceeds 90%. Even when the block interval is further increased, there is no significant change in the ratio of nodes with fair conditions.

C. Fork rate and fairness

It is known that when the block interval is fixed, a change in propagation delay appears in the fork rate [8]. It is necessary to determine the relationship between the fork rate and fairness when setting the target fork rate for the block interval adjustment described later. The following equation is used to approximate the fork rate.

$$Pr[F \geqq 1] \simeq 1 - \left(1 - \frac{1}{t_b}\right)^{\int_0^\infty (1-f(t))dt} \quad (5)$$

$$\simeq 1 - \left(1 - \frac{\int_0^\infty (1 - f(t))\, dt}{t_b}\right)$$

$$= \frac{\int_0^\infty (1 - f(t))\, dt}{t_b} \quad (6)$$

We calculate the fork rate from this equation when the block interval is set to two to three times the 90% propagation delay. Table II shows that we can obtain a guide to determine the fork rate to achieve (0.9, 0.01) fairness.

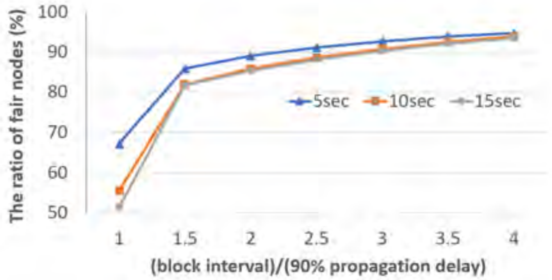

Fig. 1. Nodes with fair mining success rates.

D. Discussion on block interval

Croman *et al* provided the following lower bound on the block interval [2]. We adopt $X = 90$ based on that study.

$$\frac{\text{block size}}{\text{X\% effective throughput}} < \text{block interval}$$

$$\text{X\% effective throughput} := \frac{\text{block size}}{\text{X\% propagation delay}}$$

$$\iff 1 < \frac{\text{block interval}}{\text{X\% propagation delay}} \tag{7}$$

Applying equation 7 whereby the current 90% propagation delay is approximately 5 seconds, the block interval can be set to 5 seconds or longer. If the block interval is set to 5 seconds and we apply the (0.9, 0.01) fairness proposed in this study, Figure 1 shows that 67.2% of the nodes are fair nodes. In the current network situation, when achieving (0.9, 0.01) fairness, the block interval satisfies the following:

$$k < \frac{\text{block interval}}{\text{90\% propagation delay}} \qquad (2 < k < 3) \tag{8}$$

In addition, a similar result is obtained from Fig. 1 when the 90% propagation delay is 10 seconds and 15 seconds. Therefore, in the case of the Bitcoin network, the lower bound of the existing block interval can be made stricter by the proposed fairness technique.

IV. Block Interval Adjustment Toward Fair Blockchains

It is necessary to adjust the block interval in response to changes in propagation delays to make the blockchain achieve fairness and maintain high performance over a long period of time. A change in the propagation delay in this study means a long-term change and not a short-term change. It is assumed that the block interval is also adjusted over several months to years. This is because it is important for the block interval to be stable in the short term [19]. In this section, we introduce the proposed method and apply it to the blockchain network simulator SimBlock [20]. We prepare several scenarios of propagation delay changes and confirm that the results approach the target fork rate when applying the block interval adjustment method. Finally, we show that the ratio of unfair nodes can be reduced by setting the target fork rate of this method appropriately.

TABLE III
Settings in experiment.

# of nodes	8000
Block interval	Adjusted
Block size	534KB
# of connections	Distribution according to Miiler *et al.* [21]
Geographical Distribution	Distribution according to Nagayama *et al.* [22].
Bandwidth	Changes in each scenario.
Delay	6 regional delays according to Nagayama *et al.* [22]

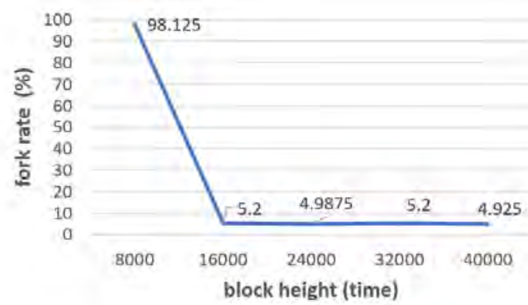

Fig. 2. Fork rate in Scenario 1: bandwidths are constant.

A. Proposal method

We can word toward the target fork rate by considering the above-mentioned discussion on fairness, performance and security. The block interval adjustment method is realized by updating this target fork rate. The update method of the block interval is shown below.

$$t_b^{(k+1)} \leftarrow \frac{r_{\text{fork}}}{Target\,(r_{\text{fork}})} \quad t_b^{(k)} \tag{9}$$

It can be seen that equation 9 is updated such that the fork rate is constant compared with equation 6.

B. Experiment

Let us check the settings of SimBlock used in the experiment. We explain how to set the bandwidth and block interval for each scenario. Table III shows the common settings. It is necessary to relay blocks other than the main chain to share the fork rate. Bitcoin's protocol is designed to relay orphan blocks (not main chain blocks) in certain situations; thus, it is difficult to apply as is. Therefore, each node can share the fork rate by changing the relay protocol of SimBlock. This change is simple and does not significantly affect the actual blockchain simulations.

1) Scenario 1: We assume a situation whereby the bandwidth hardly changes over time. To reproduce the relatively low propagation delay and high fork rate, the bandwidth is set to 1.5 times the value used by Gervais *et al* [7]. The fork rate before updating is set high to clarify the fork rate after adjusting the block interval. Figure 2 shows the change in the fork rate. In this experiment, the target fork rate is set to 5%, and the update interval is set to 8000 blocks in each scenario such that the fork rate after 8000 blocks is kept close to 5%.

2) Scenario 2: We assume that the bandwidth will increase over time. We set the bandwidth to increase by 1.4 times every 8000 blocks, and we set the initial bandwidth to be the same

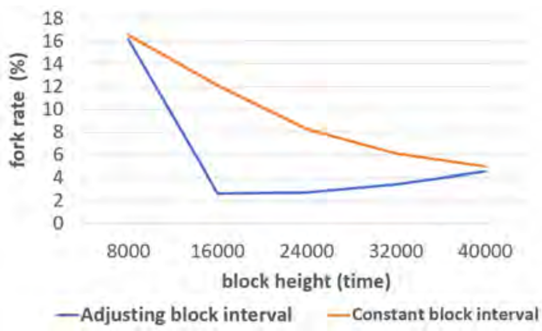

Fig. 3. Fork rate in Scenario 2: bandwidths are increasing.

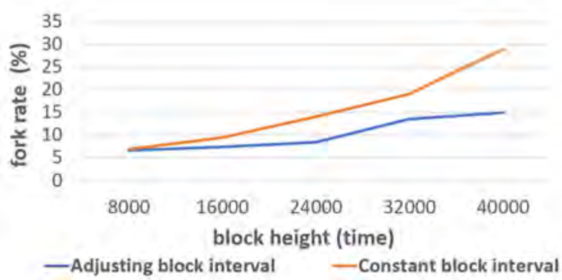

Fig. 4. Fork rate in Scenario 3: bandwidths are decreasing.

as in scenario 1, which uses data from 2015. Therefore, in scenario 2, the propagation delay decreases every 8000 blocks, and the fork rate decreases. Figure 3 shows that the fork rate approaches the target as the propagation delay decreases.

3) Scenario 3: We assume that the bandwidth will decrease over time and set the bandwidth to increase by a factor of 0.7 approximately every 8000 blocks. The initial bandwidth is the measured value [22]. Since the bandwidth in 2019 is greater than the bandwidth in 2015, it is suitable for creating a situation where the propagation delay increases over time. Therefore, in scenario 3, the propagation delay increases every 8000 blocks, and the fork rate increases. Figure 4 shows that the increase in the fork rate can be suppressed when the propagation delay increases under our method.

C. Effects on fairness

We confirm at the end of this section that the proposed method improves fairness. We examine the ratio of fair nodes by using 800 nodes, a block height of 80000, 0.7 times the bandwidth and a target fork rate of 5% every 16000 blocks. Figure 5 shows the change in the fork rate. The variation of the fork rate after applying the method is also shown. We confirmed on a simulator that 85% of nodes are fair nodes when this method is not applied and 98% when it is applied.

V. Discussion

Scenario 1 shows that the fork rate approaches the target without any change in the propagation delay. A survey of Bitcoin after 2018 shows that the change in the 90% propagation delay over several months is less than 5 sec [18]. Therefore, when applying this proposal to the actual Bitcoin network, we

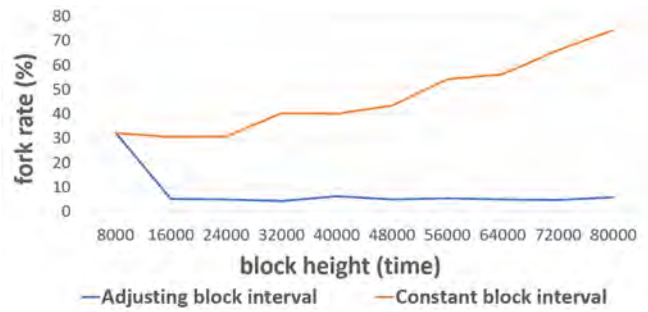

Fig. 5. Fairness and block interval adjustment

can expect to be closest to Scenario 1 when the update interval of the block interval adjustment is set to several months. In Scenario 2 and Scenario 3, the updates of the bandwidth and interval are performed simultaneously. Updating the interval only corresponds to the prior increase in propagation delay; it does not correspond to subsequent changes. For this reason, there are some values that are far from the target fork rate even after the block interval is adjusted. You can take values as close as possible. In addition, it was confirmed that the fork rate did not degrade when adjusting the block interval. At the end of the above section, we experimented with the block interval adjustment and fairness. After applying this method, the ratio of fair nodes is 98%. Compared to Figure1 and Table II, the ratio of fair nodes is higher on the simulator. This is because the target fork rate is set to 5% and because the simulator does not include a relay network.

VI. Related works

In this study, we define fairness such that the mining success rate of each miner matches the computational power for a sufficiently long blockchain. We introduce other references to fairness for blockchains. One reference is the fairness proposed by Croman, which defines unfair if the block interval is shorter than the time it takes for the block to spread across the entire network [2]. This gives an upper bound on the performance of PoW blockchains. The other definition was proposed by Gencer *et al*, which states unfair as whereby miners are mining a relatively large number of pruned blocks with respect to their computational power [23]. This gives a measure of decentralization under this fairness.

We proposed a block interval adjustment method for the purpose of keeping the blockchain safe and fair in the long term. However, if the block propagation delay decreases, we can shorten the interval accordingly by our proposal. Therefore, this proposal can be used to improve performance. For example, Bitcoin-NG, a theoretically improved version of Bitcoin, shows that the delay resulting from the block interval is limited by the diameter of the network [24]. Since the size of the network changes under this condition, we expect that it will be further improved by incorporating the method proposed herein.

VII. Conclusion

Block interval adjustment works by sharing the fork rate among participants. Existing studies on the trade-off between performance and safety provide important knowledge considering an appropriate fork rate. However, to consider the lower bound of the fork rate and block interval, a perspective considering is necessary. In this research, we can give bounds stricter than those given by existing research on fairness. It is necessary to give the target fork rate for block interval adjustment. This is also important for engineers and scientists who will develop PoW blockchains.

This work cannot be applied directly to current blockchains, such as Bitcoin or Ethereum, but can be implemented as a simple change in relay protocols. Considering that the propagation delay gradually decreases over periods of several months or years, a blockchain integrating a block interval adjustment leads to improved performance. Furthermore, in the far future, even when the propagation delay increases, the block interval adjustment works flexibly and ensures both security and fairness.

Acknowledgment

This research was supported by a SECOM Science and Technology Foundation General Research Grant.

References

[1] G. Hileman and M. Rauchs, *2017 Global Blockchain Benchmarking Study*, 2017.

[2] K. Croman, C. Decker, I. Eyal, A. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. ShiEmin, G. Sirer, D. Song, and R. Wattenhofer, "On scaling decentralized blockchains," in *Financial Cryptography and Data Security FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers*, 2016, pp. 106–125.

[3] G. Wood, *Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper*, 2014.

[4] E. Community., "next generation smart contract and decentralized application platform." Accessed: 2019. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[5] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*, 2014.

[6] Y. Sompolinsky and A. Zohar, *Secure High-Rate Transaction Processing in Bitcoin*, Jan 2015.

[7] A. Gervais, G. Karame, K. Wust, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *in Proceedings of the ACM Conference on Computer and Communication Security (CCS), 2016*, 2016, pp. 3–16.

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

[9] A. Ozisik, G. Andresen, B. Levine, D. Tapp, G. Bissias, and S. Katkuri, "Graphene: efficient interactive set reconciliation applied to blockchain propagation," in *In SIGCOMM '19: Proceedings of the 2019 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications.*, 2019, pp. 303–317.

[10] E. Rohrer and F. Tschorsch, *Kadcast: A Structured Approach to Broadcast in Blockchain Networks*, 2019.

[11] Y. Aoki and K. Shudo, *Proximity neighbor selection in blockchain networks*, 2019.

[12] U. Klarman, S. Basu, A. Kuzmanovic, and E. Sirer, "bloxroute: A scalable trustless blockchain distribution network whitepaper," Accessed: 2019. [Online]. Available: https://bloxroute.com/wp-content/uploads/2018/03/bloXroute-whitepaper.pdf

[13] T. Neudecker, *Security and anonymity aspects of the network layer of permissionless blockchains*. [Online]. Available: https://publikationen.bibliothek.kit.edu/1000089033/21409038

[14] "21.co. bitnodes." Accessed: 2019. [Online]. Available: https://bitnodes.21.co/

[15] M. Corallo, "Bip152: Compact block relay," Accessed: 2019. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0152.mediawiki

[16] FIBRE, "Fibre fast internet bitcoin relay engine," Accessed: 2019. [Online]. Available: http://Bitcoinfibre.org/public-network.html

[17] Falcon, "Falcon - a fast bitcoin backbone," Accessed: 2019. [Online]. Available: https://www.falcon-net.org/

[18] D. Systems and N. S. R. Group, "Dsn bitcoin monitoring," Accessed: 2019.

[19] S. Noda, K. Okumura, and Y. Hashimoto, *A Lucas Critique to the Difficulty Adjustment Algorithm of the Bitcoin System*, 2019.

[20] Y. Aoki, K. Otsuki, T. Kaneko, R. Banno, and K. Shudo, "Simblock: A blockchain network simulator," in *CryBlock 2019 (in conj. with INFOCOM 2019)*, 2019, pp. 199–213.

[21] A. Miller, J. Litton, A. Pachulski, N. Gupta, D. Levin, N. Spring, and B. Bhattacharjee, "Discovering bitcoin's network topology and influential nodes," in *Coinscope: Discovering Bitcoin's Network Topology and Influential Nodes (Website)*, May 2015, pp. 1–17, https://cs.umd.edu/projects/coinscope/coinscope.pdf.

[22] R. Nagayama, K. Shudo, and R. Banno, *Simulation of the Bitcoin Network Considering Compact Block Relay and Internet Improvements*, 2019.

[23] A. Gencer and S. Basu, "Decentralization in bitcoin and ethereum networks," in *International Conference on Financial Cryptography and Data Security FC 2018: Financial Cryptography and Data Security*, 2018, pp. 439–457.

[24] I. Eyal, A. Gencer, E. Sirer, and R. Renesse, "Bitcoin- NG: A scalable blockchain protocol," in *Proceeding NSDI'16 Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation*, 2016, pp. 45–59.