Hindawi
Security and Communication Networks
Volume 2019, Article ID 3674274, 13 pages
https://doi.org/10.1155/2019/3674274

Research Article

MWPoW: Multiple Winners Proof of Work Protocol, a Decentralisation Strengthened Fast-Confirm Blockchain Protocol

Yibin Xu[1] **and Yangyu Huang**[2]

[1]*School of Computer Science and Informatics, Cardiff University, Cardiff, CF24 3AA, UK*
[2]*School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China*

Correspondence should be addressed to Yibin Xu; work@xuyibin.top

Received 11 July 2019; Accepted 8 October 2019; Published 18 November 2019

Guest Editor: Veljko Milutinovic

Blockchain mining should not be a game among power oligarchs. In this paper, we present the Multiple Winners Proof of Work Protocol (MWPoW), a mining-pool-like decentralised blockchain consensus protocol. MWPoW enables disadvantaged nodes which post only a small amount of calculation resource in the mining game to create blocks together and compete with power oligarchs without centralised representatives. A precise Support Rate of blocks can be determined through the mining process; the mechanism of the mainchain determination is therefore changed and has become faster and more straightforward. A method that periodically adjusts the block size and the block interval is introduced into MWPoW, which increases the system flexibility in the changes of network conditions and data flow. Experiments suggest, without lifting calculation and bandwidth requirements, MWPoW is more attractive to disadvantaged nodes due to its mostly increased reward expectation for disadvantaged nodes. The transaction pending time is shortened chiefly, and either the block interval or the block size can be adapted amid the changes of overall network conditions.

1. Introduction

Bitcoin [1], Ethereum [2], and other blockchain systems take the research direction of acquiring a consistent view among mutually distrusting participants to public attention. Blockchain's transparency and irreversibility make it a perfect mechanism for maintaining the integrity and public confidence in applications like crowd-funding and gambling, where trust is the essence. Blockchain also brings hope to change our financial system and other fields that are vulnerable to the catastrophe caused by a failure in a single central component. Blockchain achieved the named properties with the premise that more than half of the participated resources are always honest. Every decision in the blockchain is a decentralised consensus decision among the participants instead of a centralised order from the superior governor. A blockchain is entirely decentralised when it operates in a permitless mode where it opens the membership and sustains anonymous participants to join and leave the system freely. Nevertheless, there are problems with the design of permitless blockchain. With people seeking to extend the usage of permitless blockchain to various fields, they have found that the majority implemental barrier is the dilemma between the increment of throughput and the maintenance of decentralisation.

On the one hand, the decentralisation and security of blockchain are in jeopardy. Permitless blockchain usually uses cryptocurrency to reward its participants to encourage participation; however, the less competitive nodes are expecting less or zero rewards. In Bitcoin (Nakamoto blockchain), with a standard computer having a hash rate of 1 million H/s, the computer has to mine for an average of 62,000 years to find a block [3]. The design that only the block creator receives remuneration in every iteration of the mining game has discouraged the disadvantaged nodes from participating in this tensed competition, causing a tendency to partly-centralisation. We see the same problem in Proof of Stake (PoS) [4] byzantine consensus protocol, the people

with more tokens become oligarchs who are likely to win the game. The decentralisation is further damaged when nodes join in the mining pools for a higher reward expectation. A mining pool is a centralised node that gathers the resources from individual nodes, mine as one, and distribute the funding gained based on the resources individual nodes posted. The participated nodes generally have no sense of how their resources are used; thus, honest mining pool participants can also be an accomplice of byzantines. Merged mining [5] is an example of this, where the mining pool participants' power can be used in multiple mining games of different blockchains without their acknowledgements. The same problem goes to DPoS, where the stakes of minorities are gathered by several prominent representatives, the system is partly centralised, and then the security concerns are aroused.

Blockchain needs to enable as many devices as possible to participate in maintaining the decentralisation. It must give disadvantaged devices enough time (block interval) to digest information, verify blocks, and reach a consensus with others. Every block interval consists of two subintervals: the time for creating a block and the time for broadcasting the block. The time allocation inside every block interval changes with the come and go of participants of diverse calculation ability and network situation. It is difficult to conclude a convincing reason for setting a specific block interval and whether a block interval adjustment improves the performance or damages the security. A previous research [6] shows that a significant propagation latency in blockchain network may cause miners to mine on dated blocks. If the block interval is too short, the impact of network advantage may surpass that of calculation advantage in the chance to win the mining game, causing a weakened security.

On the other hand, the throughput (transactions per second) of blockchain, in reality, is relatively low to power its potential applications. For the network latency and security concerns, the block size cannot be significant. If the block size is large, with the same block interval, the subinterval for creating a block is shorter compared to that with smaller block size. This setting may cause an unbalance of the time allocation between the two subintervals, further damaging the security of the blockchain.

Most blockchains [1, 4, 7] require three later block confirmations to confirm a transaction embedded to block. This rule brings a pending time of at least 30 minutes to accept the transaction in bitcoin [1], without counting the pending time before a block embeds this transaction. This rule also contributes to the low throughput problem because only the confirmed transactions can be used in further transactions. However, this rule is essential for the blockchain security. People only know if a branch of blocks is the longest one to their knowledge. They do not know the exact percentage of the overall calculation power that has agreed on that branch and if there is an unknown branch of blocks of more support. People must wait for a predefined time window to accept a block to the mainchain and prevent a wrong decision in a significant propagation latency. There are alternatives to this mainchain determine rule in protocols like Directed Acyclic Graph [8–10] which has uncertain transaction confirmation time and is more vulnerable to attacks. GHOST [11] shortens the time for mainchain determination by allowing people mine on the branch that has the most blocks instead of the highest branch. However, it still requires later block confirmations to accept a block finally. There is no clue of how much percentage of the overall calculation power in exact an attempt would take to overwrite the mainchain.

This problem also exists in other blockchain byzantine consensus protocols, e.g., Proof of Stake (PoS) [4] and Delegated Proof of Stake (DPoS) [7]. They take a long pending time to decide on a branch of blocks as the mainchain confidently. There are many byzantine consensus mechanisms in pre-blockchain eras; they often make security/performance trade-offs which make them no longer a trustworthy decentralised system. Byzantine consensus mechanisms such as those in [12–14] are specified working in a closed-membership setting, the participants are fixed, and they all know each other's identities through authenticated third parties. These mechanisms are incredibly vulnerable to Sybil attacks [15], where the byzantine repeatedly creates different identities to acquire a significant influence on the system. More protocols such as those in [16–18] either scale poorly with the number of participants or are inefficient.

In this paper, we show Multiple Winners Proof of Work protocol (MWPoW), a consensus protocol that attempts to increase the scalability of blockchain while maintaining the decentralisation by (1) introducing a method to dynamically adjust the block interval and make it fit into the network situation, (2) changing the way of mainchain determination and shortening the transaction pending time, and (3) strengthening the decentralisation by increasing the chance for disadvantaged participants to receive remuneration.

In MWPoW, we encourage people to join in the blockchain directly by eliminating the reward-expectation gap between mining on the blockchain and mining in a mining pool. A block is not published by one miner but by a group of miners. Miners receive remuneration based on their contribution directly if they collectively mined a block. We improve the scalability of blockchain by shortening the time to accept a block finally: any node can calculate a precise Support Rate of a block without waiting later block confirmations. An exact percentage for time allocation in block interval can be derived, either the block size or the block interval can be dynamically adjusted safely and reasonably to fit into the data flow.

Compared to the previous MWPoW paper [19], this paper makes the following improvements; parts of the MWPoW are redesigned:

(i) The mandatory grouping requirements are removed, and miners are not assigned into three predefined groups.

(ii) The block Support Rate is systematically defined and analyzed.

(iii) The branch choosing and the mainchain determination method are thoroughly redefined.

(iv) The new approach to adjust block interval and block size is introduced.

(v) The performance is compared with other fast-confirm protocols.

2. Multiple Winners Proof of Work Protocol

2.1. MWPoW Outline

2.1.1. Definition

(i) *Calculation Power Claim*. A miner's calculation power is defined as the hash difficulty one can achieve in a fixed time window. Calculation Power Claim is the hash difficulty that a miner intends to reach in every round of the mining game:

$$\mathrm{CP} = \mathrm{CP}_0 + \mathrm{CP}_1 + \cdots + \mathrm{CP}_{N-1}, \tag{1}$$

where CP is the overall power claimed by registered participants, N is the number of registered participants in the network, and CP_{N-1} is the Calculation Power Claim of registered participant N.

(ii) *New Join*. New Join is a data set, which records the Calculation Power Claim of a participant and a wallet address of this participant (the wallet address is used for receiving remuneration). There is a Nonce field in New Join, which is used for adjusting the hash of New Join. For a New Join to be valid, the hash of this New Join must meet at least the Calculation Power Claim of this New Join. Table 1 shows the structure of New Join.

(iii) *Try Range*. Try Range (TR) is a number interval of the Nonce in block header:

$$\mathrm{TR}_i = \left[\sum_{k=0}^{i-1} Tt_k, \sum_{k=0}^{i} Tt_k\right), Tt_{i\in N} = \frac{\mathrm{CP}_i}{\mathrm{CP}} * 2^{256}, \tag{2}$$

where N is the number of registered participants in the network. Miner $i \in N$ mines on TR_i.

(iv) *Acceptance Difficulty*. The first block that reaches the Acceptance Difficulty in a round of mining should be placed in the mainchain. Acceptance Difficulty is adjusted based on how much time is consumed for the winner block to achieve the Acceptance Difficulty:

$$\mathrm{AD}_x = \frac{\mathrm{BI} * \mathrm{AD}_{x-1}}{\mathrm{Timestamp}_{x-1} - \mathrm{Timestamp}_{x-2}}, \tag{3}$$

where AD_x is the Acceptance Difficulty at the block height X, BI is the predefined block interval, and $\mathrm{Timestamp}_x$ is the time when block X is created.

(v) *Entrance Difficulty*. A block is broadcasted to the network when this block reached Entrance Difficulty. Entrance Difficulty of a new round is adjusted base on how many blocks reached Entrance Difficulty in the previous round of the game:

$$\mathrm{ED}_x = \min\left(\frac{\mathrm{NE}_{X-1}}{\mathrm{DN}} * \mathrm{ED}_{X-1}, \frac{\mathrm{AD}_x}{2}\right), \tag{4}$$

where ED_x is the Entrance Difficulty at the block height X; NE_{x-1} is the number of blocks reached Entrance Difficulty at block height $X - 1$; DN is the ideal number of NE, and we set DN = 1.

(vi) *Share*. Share is a container of Nonce when broadcasting. The Nonce inside a Share which is sent by a miner must make the hash of the block fulfill at least 25% of this miner's Calculation Power Claim. Table 2 shows the structure of Share.

(vii) *Countable Share*. If a miner has sent at least two valid Shares for a block, the difficulties of these Shares are counted toward the Support Rate of this block and the miner will be able to receive remuneration for announcing this block if this block wins the game later.

(viii) *Share Difficulty Cap*. The maximum sum of difficulties of Countable Shares sent by a miner X in a round of game is CP_X (its Calculation Power Claim). If it sends more, the sum is capped at CP_X.

(ix) *Support Rate*. The Support Rate of a block is defined as the ratio between the sum of the difficulties of the Countable Shares for the branches stem from this block and the sum of difficulties of all Countable Shares of all the branches in the blockchain since the block height of this block:

$$\mathrm{SR}_X = \frac{\sum_{i=X}^{\mathrm{XL}} \mathrm{SD}_{\{i\}}}{\sum_{i=0}^{k} \sum_{j=i}^{\mathrm{iL}} \mathrm{SD}_{\{j\}}}. \tag{5}$$

where SR_X is the Support Rate of block X; XL is the latest block on top of the blockchain branch stem from block X; k is the number of all the branches; iL refers to the latest block on top of the specific branch; $\mathrm{SD}_{\{i\}}$ is the total difficulty of the Countable Shares for block i.

Figure 1 shows an example where red and blue dotted boxes include the branches stem from block 1 and block 2, respectively. The Support Rate of block 1 is 28.57% while which of block 2 is 71.43%.

(x) *Reward*. The reward is given as follows:

$$R_{i\in N^R} = \frac{\mathrm{SD}_i}{\mathrm{SD}_{\{X\}}} * R_{\{X\}}. \tag{6}$$

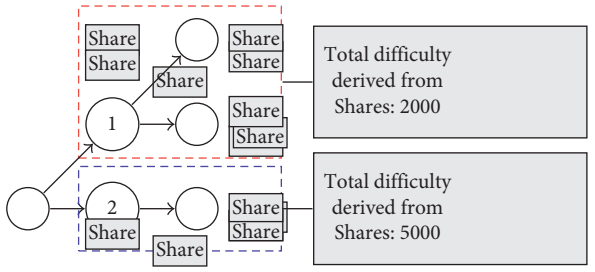
where N^R is the set of miners who contributed the Countable Shares for announcing block X; $R_{\{X\}}$ is the overall reward assigned from the system for the block at the block height X; Shares of block X are embedded in block $X + 1$; $\mathrm{SD}_{\{X\}}$ is the total difficulty of the Countable Shares embedded in block $X + 1$; SD_i is the difficulty of the Countable Shares miner i contributed to; and $R_{i\in N^R}$ is the amount of remuneration given to miner i as a Coinbase transaction in block $X + 1$.

Table 1: Structure of New Join.

Filed	Purpose	Bytes
HashPrevBlock	A 256-bit hash of the latest block in the mainchain	32
Intended_difficulty	The power (difficulty) which the miner intended to place into the mining game	4
Wallet address	For receiving compensation	34
Nonce	Hash tried (A 256-bit number starts from 0)	32

Table 2: Structure of Share.

Filed	Purpose	Size (bytes)
L_4_D_IH	Last 4 bytes of the block header hash of a block candidate	4
Nonce	Hash tried (A 256-bit number)	32



$SR_1 = 2000/(2000 + 5000) = 28.57\%$ $SR_2 = 5000/(2000 + 5000) = 71.43\%$

Figure 1: The Support Rate of block 1 and block 2.

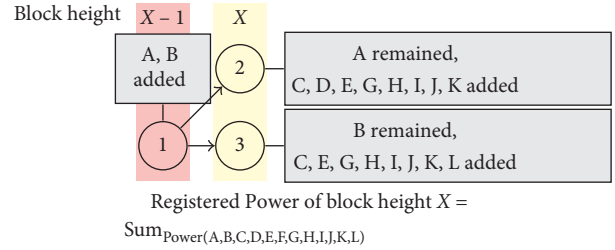


Figure 2: Explanation for Registered Power.

(xi) *Valid Block*. A miner determines a block as a valid one when the transactions, New Joins and Shares in this block are correct; the Shares and New Joins must be more than 90% previously known to the miner.

(xii) *Registered Power*. The Registered Power of a block height is defined as the sum of all the new calculation power shown in the New Joins embedded in the block sat this block height plus the sum of all the remaining calculation power after expelling unqualified miners who failed to show 50% of their powers when announcing the preceding block. Figure 2 shows an example, where the New Joins of miner A and miner B are embedded in block 1, Miner A worked on block 2, and miner B worked on block 3 (that is why they are expelled from the other branch). The New Joins of miners C, D, E, F, G, H, I, J, K, and L show up on the network and should be embedded into a block at the block height X; block 2 does not include L, and block 3 does not include D; however, both blocks can be accepted as the discrepancy does not excel 10% of the contents in the network. But, when calculating the Registered Power, all the new participants as well as the remained miners should count.

(xiii) *Restriction for Branch Choosing*. If a miner has sent two Shares for a block, this miner is not allowed to change branches in this round of the game. If a miner broadcasted two or more valid Shares for blocks of different branches at the same block height, the miner would be expelled from the game and its contribution will not be counted toward the Support Rate.

(xiv) *Statement Rate*. $STR_X = SRP_X/Rp_X$, where $SRP_X = \text{sum}(CP_{i \in k})$, k is the set of miners who have sent two valid Shares for a block at the block height X and are not violating the restriction for branch choosing, and Rp_X is the Registered Power at the block height X.

(xv) *Noncommittal Power*. If a miner has not yet sent two Shares for a block, its power is counted toward the Noncommittal Power. If a miner breaks the restriction for branch choosing, its Shares will be eliminated from the Support Rate of all blocks at this block height and will not be counted toward Noncommittal Power.

(xvi) *Block Size*. The number of transaction allowances per block.

2.1.2. Game Overview. Miners need to claim the calculation power they intended to put into the mining game before participating the game. Each miner is given a unique TR based on the calculation power it claimed. When a miner creates a block and finds a Nonce that fulfill Entrance Difficulty in its TR, it will broadcast the block as well as the Share. Then other miners will attempt to find a Nonce in their TR to make this block fulfill the Acceptance Difficulty

if they acknowledge this block as a valid one. Ideally, miners should announce a block collectively by doing PoW in their TR in parallel. When a Share of a block is broadcasted and the Nonce in it made the block reached the Acceptance Difficulty, this block is announced. The first block that reaches Acceptance Difficulty is the winner block, and miners who contribute Shares to this block will divide the remuneration of mining. During the announcement, miners should send Shares which do not fulfill the Acceptance Difficulty but fulfill at least 25% of the power they claimed previously as the proof of contribution. A miner can only send up to four Shares to the network per round of the game. If more than one block is successfully announced in a round of game, miners should mine on top of the one first reached Acceptance Difficulty from their perspectives (miners may have different views due to the network delay). Assume that this winner block is the block X, the blocks at the next block height (block $X + 1$) will embed Shares of block X. According to the Shares embedded, if a miner failed to find the Shares which together weighted more than 50% of the power it previously claimed, this miner will be expelled from the game (its TR will be cancelled since BH $X + 1$). The remuneration for the miners of block X is given at the block height BH $X + 1$ through the Coinbase transaction. All the valid miners of block X divide the reward based on the difficulty of Shares they sent. As every miner oversees different Try Ranges, it is easy to determine who should receive remuneration and what amount of remuneration.

2.1.3. Game Procedure

(i) Register power: A new miner creates and submits a New Join to the system.

(ii) Get a Try Range: Miners whose New Joins are embedded into a block will be assigned with Try Ranges.

(iii) Mining: The miners try to create a block and find a Nonce that fulfills Entrance Difficulty in theirTR; if a block created by a miner has reached Entrance Difficulty and other miners approve this block as a Valid Block, they will try to find a Nonce that fulfills Acceptance Difficulty in their Try Ranges.

(iv) Getting reward: If the miner submitted adequate number of valid Shares for the winner block, the amount of reward would be given at the next block height.

(v) Rearrange Try Range and start over: After every round of the game, the invalid miners will be globally expelled (miners who failed to send Shares which stand for at least 50% of the power they claimed, their Try Ranges will be cancelled). Then, new miners are added, and the Try Ranges for all the valid miners be rearranged. After that, a new round of the game starts. Miners who submitted New Join before and were not expelled do not need to register power again to participate the new round of the game.

2.2. Block Simplification and Bandwidth Demand. Because New Joins and Shares are embedded to the block, the block size is increased tremendously. We use a block simplification algorithm Graphene [20] to simplify the block as to lower the bandwidth demand. The structure of MWPoW block is given in Figure 3. Graphene is a block simplification method which uses Bloom filter (BF) [21] and IBLT [22]. It can encode 2000 transactions into 2.6 kbytes, and the encoded blocks can be decoded by nodes using the previously received transactions.

Though the block is simplified, nodes still need to promptly hear all the New Joins, Shares, and transactions in the system to decode the simplified blocks. A New Join sized 102 bytes while a Share sized 36 bytes. Figure 4 shows the amount of data needed for hearing New Joins and Shares with the different number of valid miners in the network. Every miner sends four Shares to the network per round of the game, and in every round of the game, 200 new miners are added into the network until there are 10000 miners.

Bitcoin nowadays has a steady number of around 8000 miners worldwide in the network. Let us assume MWPoW also has this user scale, then the minimum bandwidth for a node will be 1.12/Interval Mbytes, where Interval is the predetermined block interval (in minutes). The minimum bandwidth with different block interval is shown in Figure 5. It only requires an additional 2 kbytes/s to function MWPoW with the Bitcoin setting (block interval of 10 minutes).

2.3. Distributed Remuneration. According to the Shares embedded in the block, the remuneration will be given to the miners of the winner group directly through a Coinbase transaction. The amount of remuneration for miners is calculated based on the total reward amount of the last block height multiplied the sum of the miner's valid Share proportions to the sum of all the Shares in winner group. Figure 6 shows an example of this, where the sum of the difficulty of the Shares sent by the Miner A and Miner B is 212 and 49, respectively, and the sum of the difficulty of all the valid Shares of the group is 1000. The total reward amount of the last block height is 100. Miner A and Miner B receive 21.2 coins and 4.9 coins, respectively. It should be noticed that if the difficulty of the valid Shares of a miner excelled the difficulty it claimed in the New Join, the system should use the difficulty in the New Join to calculate remuneration.

2.4. Fast Confirmation. It is predefined that the miners should mine on top of the block which, to their knowledge, first reached the Acceptance Difficulty. However, a miner may shift to mine on top of another block if this miner is allowed to change branches (when the miner has not yet sent two Shares for a specific block), and there is another block of Acceptance Difficulty with more Support Rate.

A block is finally accepted when the following finally acceptance criteria are satisfied:

(1) This block is inside the highest branch of the mainchain.

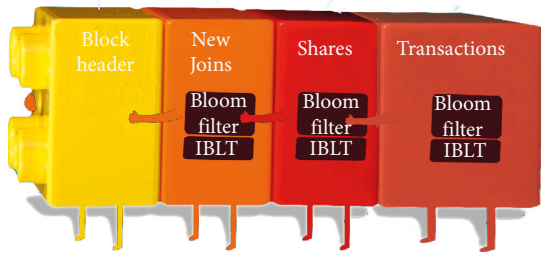


Figure 3: Block overlook.

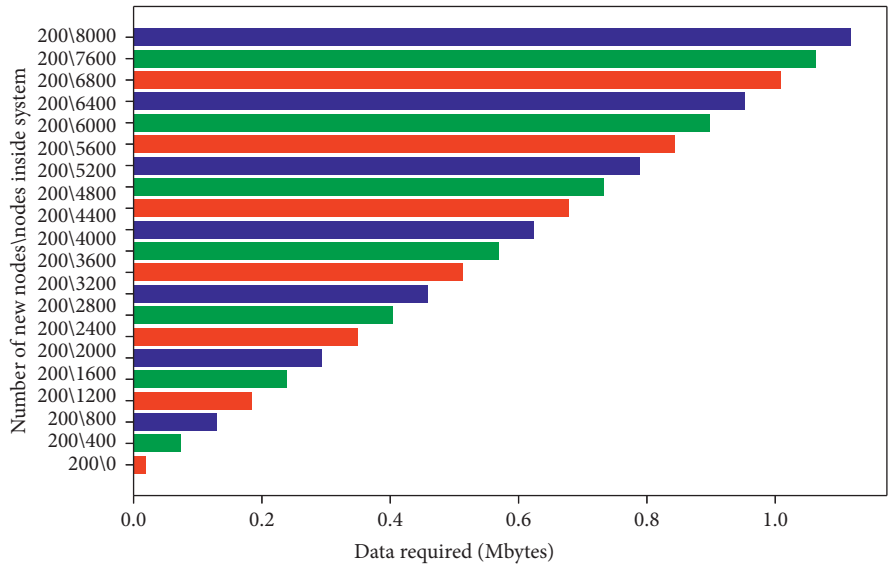


Figure 4: The amount of data for New Join and Shares VS different miner number.

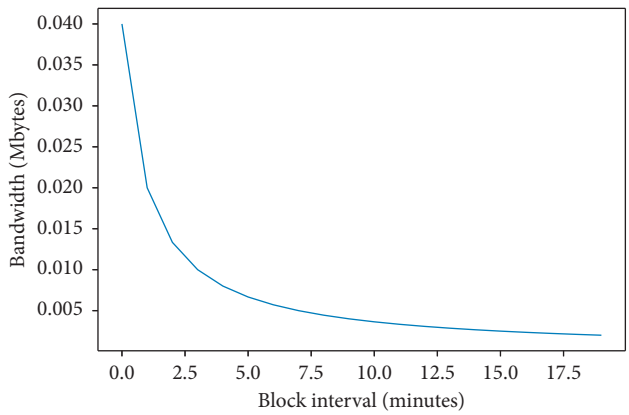


Figure 5: The minimum bandwidth for New Join, Shares VS block interval.

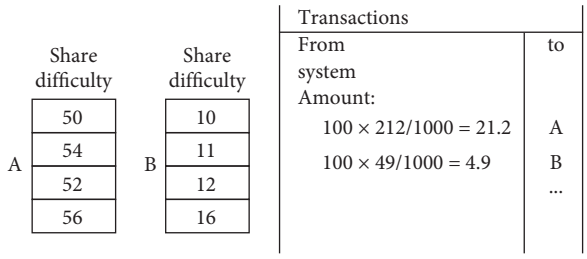


Figure 6: Reward assignment.

(2) The Statement Rate of the latest block height is larger than 50%

(3) The calculation power that reflected by the difference between the Support Rate of this block and the Support Rate of the second largest block at the same block height is larger than 25% of the Registered Power of the latest block height.

Figure 7 shows an example of the branch choosing. In (*a*), final acceptance criteria (3) is not met so that we cannot yet determine which block is the final accepted one. In (*b*), where there are succession blocks of blocks *A*, *B*, and *C*, the Support Rate of blocks *A*, *B*, and *C* are changed. Blocks *C* and *D* are finally accepted because the differences of difficulty between this branch and other branches are larger than 25% of the latest Registered Power. Block *C* and the branch stems from it have Shares that altogether stand for 6110 difficulties while the second popular block—block *B* and its branch altogether have 3172 difficulties; the difference is 2938, that is, about 41% of the Registered Power is exceeded the final acceptance criteria.

2.5. Adjustment of the Block Interval and Block Size. Let T_x be the timestamp for the time of creating block x, TT_x is the timestamp that indicates the time when the creator of block x heard first out of the four Shares of block $x-1$ from

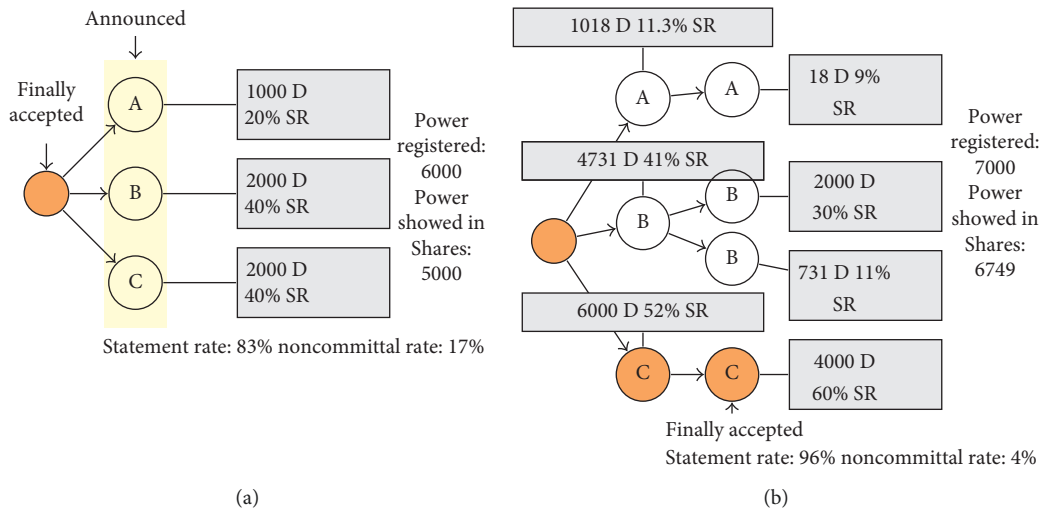


Figure 7: Finally accept a block.

miners whose Calculation Power Claim together stand for at least 50% of the overall registered power at the block height $x-1$; both T_x and TT_x are embedded in the block header of block x. The block interval is defined as interval $= S + M$, where S is the time for synchronization and M is the time for mining. Real time S and M can be derived from (1) $\mathrm{TT}_x - T_{x-1} = S + 0.25 * M$ and (2) $T_x - T_{x-1} = S + M$. If a desire ratio $R = S/M$ is predefined, the block interval and block size can be adjusted accordingly.

2.5.1. Block Interval Adjustment. Block interval is adjusted as follows:

$$\frac{I-M}{M} : R = \mathrm{Interval}_x : \mathrm{Interval}_{x-1}, \tag{7}$$

where $I = T_x - T_{x-1}$, $M = I - (TT_x - T_{x-1})/0.75$ and $\mathrm{Interval}_x$ is the setting value of the block interval between the block height x and the block height $x+1$.

2.5.2. Block Size Adjustment. The block size is adjusted as follows:

$$\frac{R * M}{I-M} = \frac{\mathrm{BS}_{x+1}}{\mathrm{BS}_x}, \tag{8}$$

where BS_x is the block size of the block height x.

In the experiment section, we will examine how block interval, block size, and acceptance difficulties change when the number of nodes and the bandwidth of nodes fluctuates.

3. Security Analysis

3.1. Restriction of Branch Choosing. If a miner has sent two valid Shares for a block at a block height, this miner is not allowed to change branches in this iteration. This rule sets a bottom line for speculators; if the rule is broken, their power is not considered Registered Power and they are expelled since the next round of the game.

3.2. Attack. Because the Support Rate of a block may be reduced when a miner breaks the restriction of branch choosing, this brings the room for the attack. Figure 8 shows an example of the attack, where the rectangles in red stand for the attacker's power. In (a), block Alice is finally accepted because its Support Rate is more significant than Bob (the second largest) for more than 25% of the Registered Power and the Statement Rate is more significant than 50%. The attacker has placed at least two Shares for Alice. Otherwise, its Shares are not Countable Shares for Alice. Because the maximum sum of difficulties of Shares one can contribute to a block height is its Calculation Power Claim (the difficulty is capped if one contributed more), it takes at least 50% the attacker's Calculation Power Claim to make two valid Shares for Bob (otherwise the Shares are not countable); after that, we entered situation (b). As condition (b) violates the restriction of branch choosing, the attacker's power is eliminated from both Alice and Bob and then (c) is reached, where Bob is the finally accepted block instead of Alice.

Let us assume block Alice reached final acceptance criteria and had a F Support Rate; at that time, block Bob had $F - D$ Support Rate in the network. The attacker must have placed at least D amount of Registered Power into block Alice's branch through at least two Shares submitted at the latest block height of Alice's branches. Then, a similar D amount of power must be used to create two more valid Shares for another branch which does not stem from Alice to make the attacker's contribution to Alice invalid.

The attacker must have claimed a power that is around two times of D when joining in the game because each Share stands for at least 25% of the power and the maximum difficulty of the sum of the attacker's countable Shares is its power claim. For block Alice to reach acceptance criteria at first, D must be equal to or larger than 25% of the overall Registered Power (see (3) of *Fast confirmation section*). So, an attack costs $2 * D$ power (at least 50% of the overall Registered Power.)

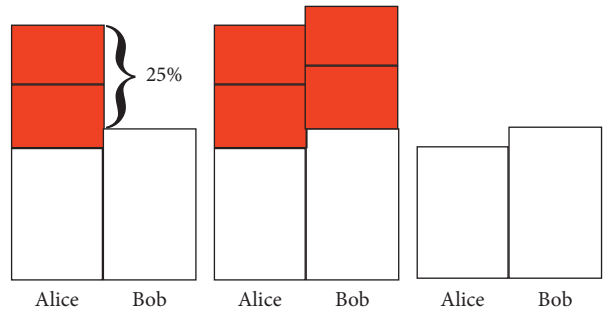


Figure 8: The attack simulation.

4. Related Work and Contributions

In 2013, Sompolinsky and Zohar [11] proposed the GHOST protocol. GHOST handles more transactions per second than Nakamoto blockchain because it can determine the mainchain faster. Not only the earliest block at each round of the game is accepted, other blocks, e.g., orphaned blocks, which are found later could also be accepted. GHOST accelerates the mainchain determination by allowing block parallelism in the network; it prevents the computational power to scale up the network. Some ideas of MWPoW are similar to GHOST: Shares in MWPoW function like fork blocks in GHOST for mainchain determination except Shares are much smaller and generated faster. Also, every Share brings an explicit increase or decrease of the Support Rate for every pending block.

Eyal et al. [23] improved the transaction per second in their blockchain protocol named Bitcoin-NG. Bitcoin-NG brings more blocks within one round of mining game. Key blocks and macroblocks are two different block types in Bitcoin-NG. The key blocks carry no transactions; they are used for leader elections solely. In the meantime, the macroblocks are employed to deliver transactions proposed by the creator of the key block. The Share structure of MWPoW is also similar to macroblocks in Bitcoin-NG, with no transaction attached. We do not add transactions into the Share structure, because the macroblocks in Bitcoin-NG are linearly linked to each other. They are sent by the key block creator only so that it guarantees the transactions are not duplicated. However, the Share structure does not link with other, and Shares are used to determine the Support Rate and reward distribution, which features the macroblocks cannot provide.

Lightning blocks [24] proposed by Poon et al. uses an off-chain micropayment channel between two parties to improve the transactions per second in the Bitcoin network. Micropayment channel allows both parties to send large transactions in an instant between the parties after only a few transactions been included into the blockchain. However, as it is only an application running on top of Bitcoin, it still underlies Nakamoto blockchain regardless of its contribution to enhance the transaction throughput. It does not solve the core problem of the Nakamoto protocol.

Buterin et al. [25] explore another approach that aims to ease the scalability problem. This approach involves sampling and challenging techniques. In their model, participants are split into several subcommittees in the network to distribute computation and verification cost. Random sample verifier verifies the correctness of others' updates and challenges some others' verification results. However, some invalid updates would never be detected under this mechanism.

Regarding block size/block interval adjustment or a clear view of the time allocation for data synchronisation and mining, we did not find any previous research that addresses these issues.

Besides the contribution toward a shorter transaction pending time and the methods of adjusting block size and block interval, MWPoW contributes a clear security threshold for the blockchain. Other consensus protocols would not achieve this without registering power before the mining game using New Join and counting the Support Rate using Shares.

5. Experiment

Three experiments were conducted; the first one compared the reward distribution for nodes which participated in Nakamoto blockchain and MWPoW. The second one tested the change of block interval and the block size in different network schemes with a fixed R. The third one tested and compared the performance of MWPoW and GHOST regarding the speed of mainchain determination in the different network schemes.

5.1. Experimental Setup. We simulated three networks A, B, and C of 1000, 2000, and 5000 nodes, respectively, using a communication protocol specialised for blockchain systems [26]. We gave every connection a random delay time ranging from 10 MS to 200 MS; the distribution of the connection delay time is shown in Figure 9. During the experiment, we assigned three bandwidth schemes of 1 Mbytes/s, 5 Mbytes/s, and 10 Mbytes/s in average for every network; the bandwidth situation is shown in Figure 10. The calculation ability for every node was 5 MH/S fixed; we used the fixed setting because the differences in calculation ability only affect the reward distribution, and it does not change the speed of Share generation; if a miner registered more, this miner needs to calculate more in the same time window.

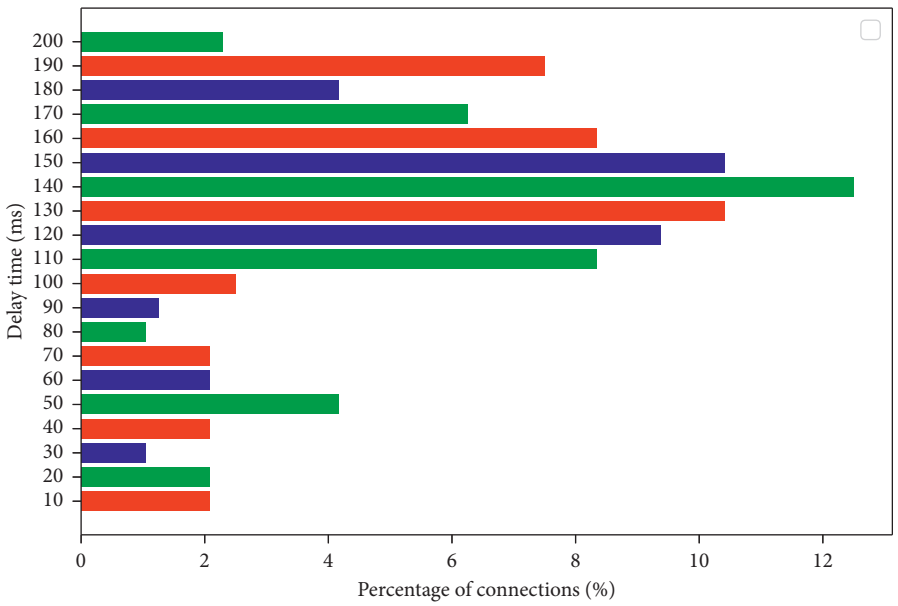
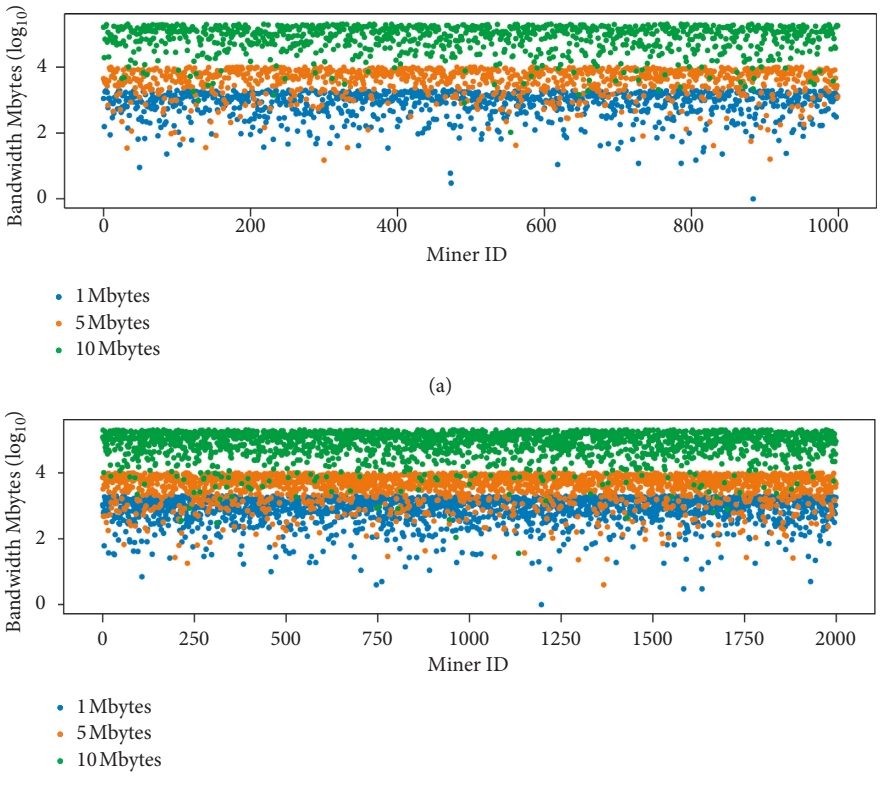


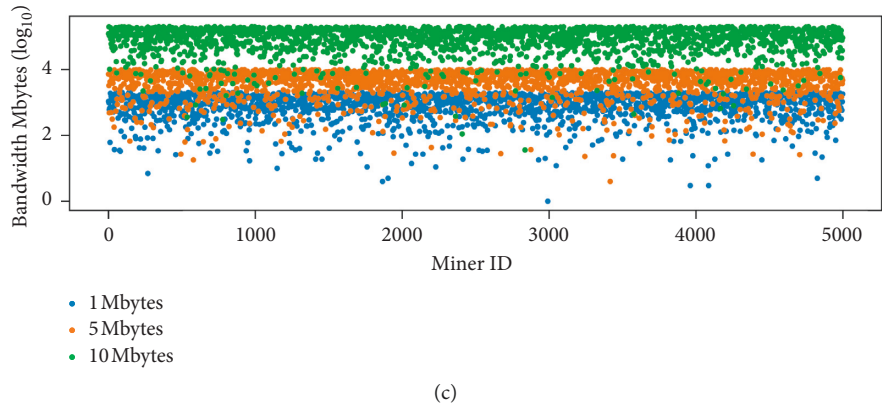
Figure 9: Delay time distribution.



(a)

(b)

Figure 10: Continued.



(c)

Figure 10: Bandwidth distribution (a) Network A. (b) Network B. (c) Network C.

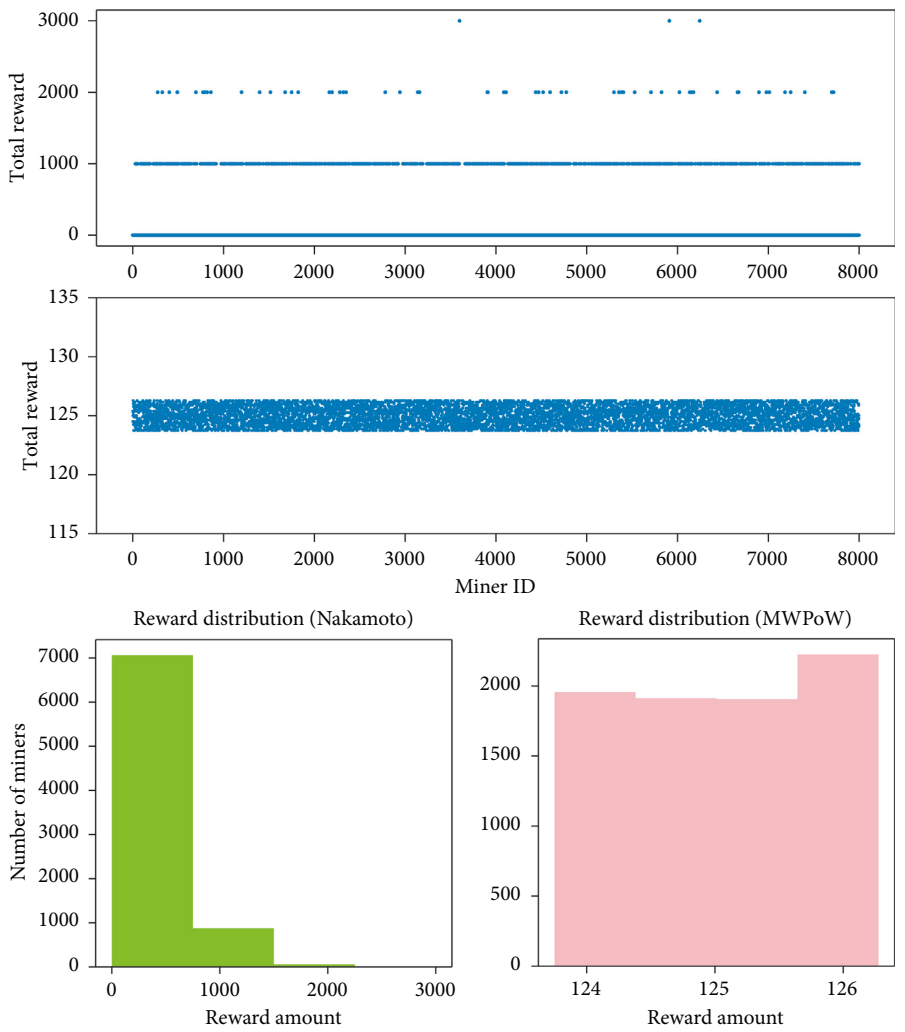


Figure 11: Reward experiment.

5.2. Reward Distribution. We ran a Nakamoto blockchain network and an MWPoW blockchain network as a comparison in network *B* with 5 Mbytes/s bandwidth in average. The block interval was set to be 5 minutes, the hash rate for every node was a fixed 5 MH/s, meaning every node gets the same chance to publish a block or finding a valid Share. The mining reward was set to be a fixed 1000 coins per block. Figure 11 presents the distribution of funding after the experiment ran for 1000 iterations. The experiment shows most miners in Nakamoto blockchain did not receive a

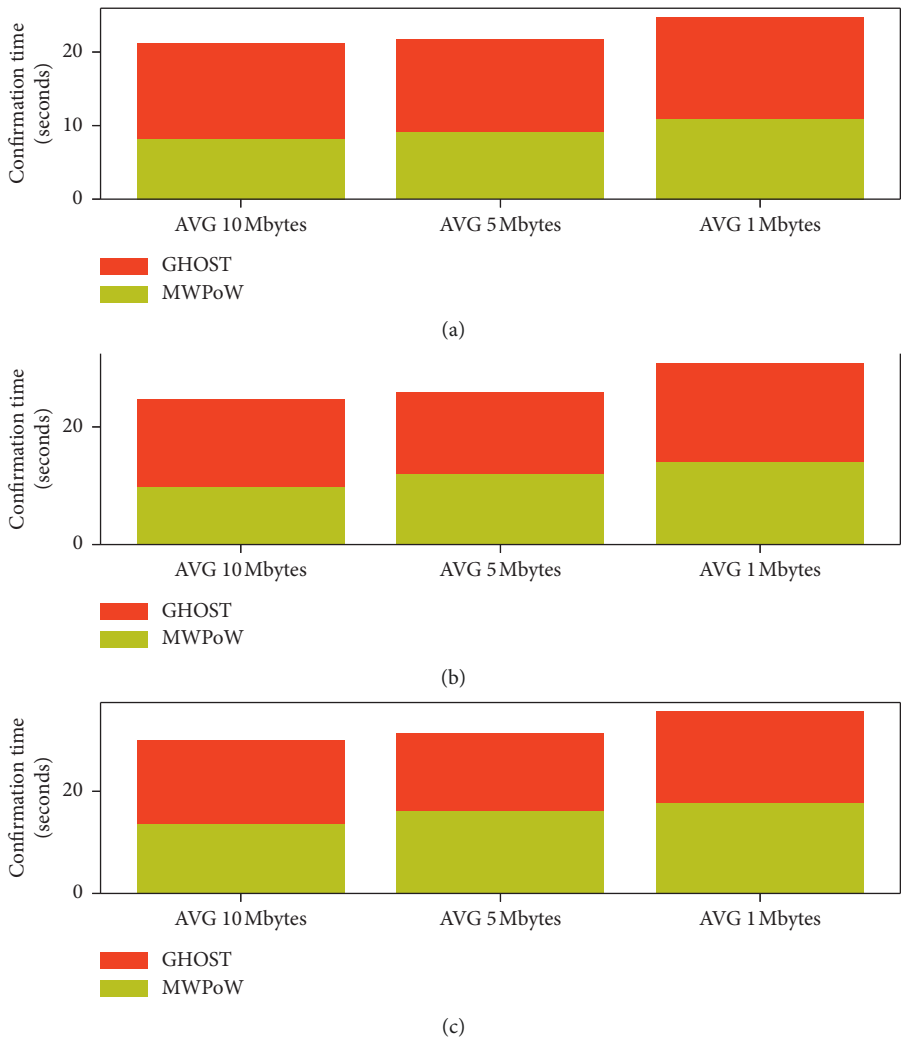


Figure 12: Block confirmation speed (a) Network A. (b) Network B. (c) Network C.

reward, only the minority of them received tremendous rewards while miners in MWPoW all received remunerations, and the currency was mostly distributed among miners.

5.3. Block Confirmation Speed Test and Comparison. We compared the performance regarding the mainchain determination speed between GHOST protocol and MWPoW in networks A, B, and C with different bandwidth schemes. Both MWPoW and GHOST used the same network with the same nodes of the same connections. To make the experiment fair, the block size and the block interval of MWPoW were not adjusted, and the block size and the block interval were 1000 transactions per block and 30 seconds, respectively, for both MWPoW and GHOST. Every subexperiment lasted for 500 block heights and was repeated 50 times. 200 transactions were sent to the network per second. The result shown in Figure 12 is the average time for accepting a block finally. As can be seen from the result, MWPoW outperformed the GHOST protocol and the average accepting time was around (1/4) of the block interval regardless of the experiment settings. Usually, MWPoW can finally accept a block when that block is announced because most miners have exchanged four views through four Shares. The cost of time for accepting this block is (0/4) of the block interval at the next block height. If a block cannot be finally accepted when announced, then which block is the final accepted block can usually be decided at around (1/2) of the block interval at the next block height. At that period of the next block height, the Shares of most miners become countable Shares and that brings a change to the Support Rate of the preceding block. Thus, on average, the accepting time is around (1/4) of the block interval. The power register mechanism and the Support Rate in MWPoW accelerated the speed of block confirmation. This result is achieved likely because GHOST needs to count the number of blocks stemmed from a block to determine if this block should be finally accepted. However, MWPoW counts the difficulty of Shares sent for the blocks stemmed from a block. It is much easier to generate a Share than a block.

5.4. Adjustment of Block Interval/Block Size. The change of block interval and block size was performed alternatively

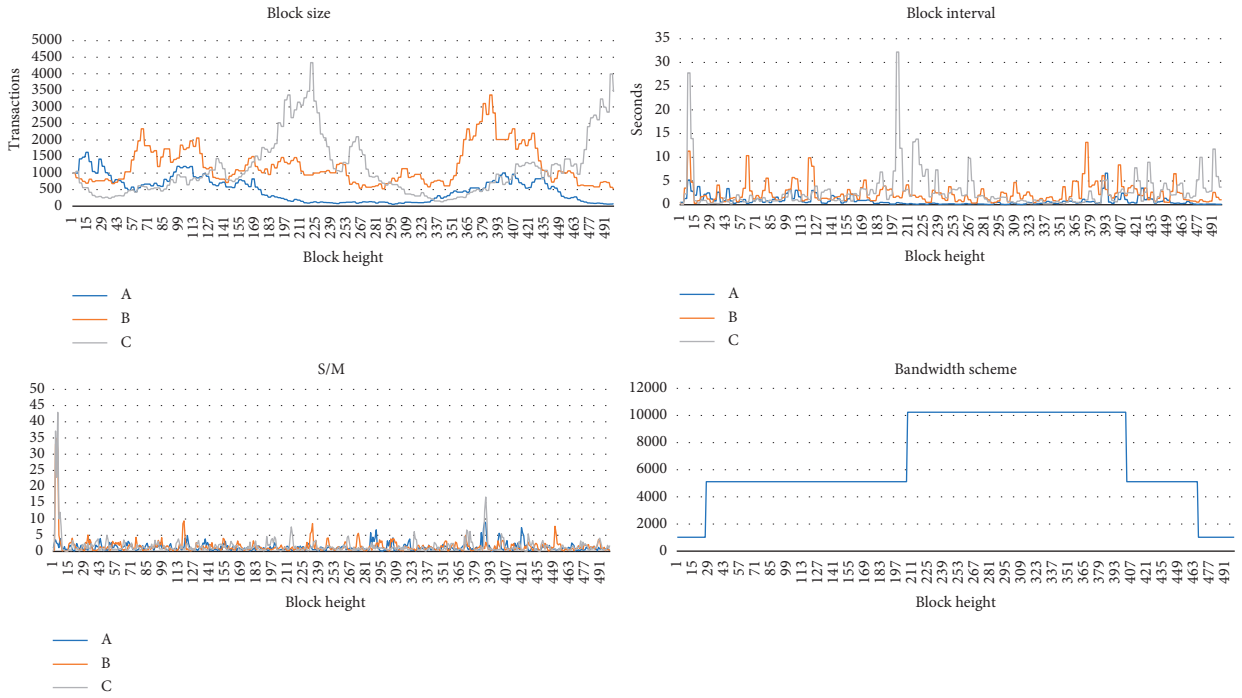


Figure 13: Block interval/block size adjustment. $R = 1$

with the change of difficulties. In this experiment, if the block height mod three equals zero, the network difficulties were adjusted; if the block height mod three equals two, the block size was adjusted; otherwise, the block interval was adjusted. We tried $R = 1$ in this experiment. A transaction sized 300 bytes was fixed, and 20 transactions were sent per second from the random nodes in the network. The default block interval at the beginning was 5 seconds; the default block size was 1000 transactions. The experiment lasted for 500 block heights; during the experiment, the bandwidth scheme was shifted from 1 Mbytes/s to 5 Mbytes/s and 10 Mbytes/s and went back to 5 Mbytes and 1 Mbytes. Figure 13 shows the result. We can see from Figure 13 that both the block interval and block size were adjusted dynamically during the experiment.

6. Conclusions

In this paper, we introduced a new version of Multiple Winner Proof of Work protocol (MWPoW) with detailed definitions of the Support Rate, branch choosing, and the unique feature of block interval/block size adjustment. We attempted to increase the scalability of blockchain by shortening the pending time to accept a block finally and strengthen the decentralisation of blockchain by increasing the chance for miners to profit from the game. The experiment result shows that MWPoW is much faster than GHOST in mainchain determination. Meanwhile, the block interval and block size can be changed with the change of network situation as like adjusting network difficulties to make the protocol more adaptable. Also, it is much easier to receive remuneration in MWPoW than in Nakamoto blockchain.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors appreciate the help from Zhan Tong Zhang from the Ecole polytechnique fédérale de Lausanne (EPFL).

References

[1] S. Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, October 2019, https://bitcoin.org/bitcoin.pdf.

[2] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 1–32, p. 151, 2014.

[3] L. Luu, R. Saha, I. Parameshwaran, P. Saxena, and A. Hobor, "On power splitting games in distributed computation: the case of bitcoin pooled mining," in *Proceedings of the 2015 IEEE 28th Computer Security Foundations Symposium*, pp. 397–411, IEEE, Verona, Italy, July 2015.

[4] A. Kiayias, A. Russell, B. David, and O. Roman, "Ouroboros: a provably secure proof-of-stake blockchain protocol," in *Proceedings of the 39th Annual International Cryptology Conference*, pp. 357–388, Springer, Santa Barbara, CA, USA, August 2017.

[5] A. Judmayer, A. Zamyatin, N. Stifter, A. G. Voyiatzis, and E. Weippl, "Merged mining: curse or cure?," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pp. 316–333, Springer, Berlin, Germany, 2017.

[6] C. Perez-Sola, J. A. D. Donet, and J. Herrera-Joancomart, "The bitcoin p2p network," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 87–102, Springer, Berlin, Germany, 2018.

[7] D. Larimer, "Delegated proof-of-stake (DPOS)," Bitshare Whitepaper, Deventer, Netherlands, 2014, https://bitshares.org/technology/delegated-proof-of-stake-consensus/.

[8] S. Popov, *The Tangle*, vol. 131, 2016, http://tanglereport.com/wp-content/uploads/2018/01/IOTA_Whitepaper.pdf.

[9] F. M. Benčić and I. V. Žarko, "Distributed ledger technology: blockchain compared to directed acyclic graph," in *Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1569-1570, IEEE, Vienna, Austria, July 2018.

[10] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "Spectre: a fast and scalable cryptocurrency protocol," *IACR Cryptology ePrint Archive*, vol. 1159, 2016.

[11] Y. Sompolinsky and A. Zohar, "Accelerating bitcoin's transaction processing. fast money grows on trees, not chains," *IACR Cryptology ePrint Archive*, no. 881, 2013, https://eprint.iacr.org/2013/881.

[12] L. Lamport, "The part-time parliament," *ACM Transactions on Computer Systems*, vol. 16, no. 2, pp. 133–169, 1998.

[13] C. Cachin, K. Kursawe, and V. Shoup, "Random oracles in constantinople: practical asynchronous byzantine agreement using cryptography," *Journal of Cryptology*, vol. 18, no. 3, pp. 219–246, 2005.

[14] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, vol. 99, pp. 173–186, New Orleans, LA, USA, February 1999.

[15] J. R. Douceur, "The sybil attack," in *Proceedings of the International Workshop on Peer-To-Peer Systems*, pp. 251–260, Springer, Cambridge, MA, USA, March 2002.

[16] B. Gabriel, "An asynchronous (n−1)/3-resilient consensus protocol," in *Proceedings of the Third Annual ACM Symposium on Principles of Distributed Computing*, pp. 154–162, ACM, Vancouver, Canada, August 1984.

[17] R. Canetti and T. Rabin, "Fast asynchronous byzantine agreement with optimal resilience," in *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing—STOC '93*, vol. 93, pp. 42–51, Citeseer, San Diego, CA, USA, May 1993.

[18] V. King and J. Saia, "Breaking the O(n2) bit barrier: scalable byzantine agreement with an adaptive adversary," *Journal of the ACM*, vol. 58, no. 4, p. 18, 2011.

[19] Y. Xu and Y. Huang, "Mwpow-multi-winner proof of work consensus protocol: an immediate block-confirm solution and an incentive for common devices to join blockchain," in *Proceedings of the 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, pp. 964–971, IEEE, Melbourne, Australia, December 2018.

[20] A. P. Ozisik, G. Andresen, B. George, A. Houmansadr, and B. Levine, "Graphene: a new protocol for block propagation using set reconciliation," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pp. 420–428, Springer, Berlin, Germany, 2017.

[21] J. K. Mullin, "A second look at bloom filters," *Communications of the ACM*, vol. 26, no. 8, pp. 570-571, 1983.

[22] M. T. Goodrich and M. Mitzenmacher, "Invertible bloom lookup tables," in *Proceedings of the 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 792–799, IEEE, Monticello, IL, USA, September 2011.

[23] I. Eyal, A. E. Gencer, E. Gün Sirer, and R. Van Renesse, "Bitcoin-NG: a scalable blockchain protocol," in *Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI '16)*, pp. 45–59, Santa Clara, CA, USA, March 2016.

[24] J. Poon and T. Dryja, *The bitcoin lightning network: scalable off-chain instant payments*, 2016, https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf.

[25] B. Vitalik, C. Jeff, and W.-D. Matthew, *Notes on scalable blockchain protocols*, October 2019, https://pdfs.semanticscholar.org/ae5b/c3aaf0e02a42f4cd41916072c87db0e04ac6.pdf.

[26] Y. Xu and Y. Huang, "Contract-network protocol: an efficient communication protocol for distributed ledger technology," in *Proceedings of the 38th IEEE International Performance Computing and Communications Conference*, IEEE, London, UK, October 2019, http://gofun.online/document/communication-protocol.pdf.