Proof-of-Play: A Novel Consensus Model for Blockchain-based Peer-to-Peer Gaming System

Ho Yin Yuen
Hong Kong Polytechnic University
Hong Kong, China
andy.aa.yuen@connect.polyu.hk

Feijie Wu
Hong Kong Polytechnic University
Hong Kong, China
harli.wu@connect.polyu.hk

Wei Cai*
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
caiwei@cuhk.edu.cn

Henry C.B. Chan
Hong Kong Polytechnic University
Hong Kong, China
henry.chan.comp@polyu.edu.hk

Qiao Yan
Shenzhen University
Shenzhen, China
yanq@szu.edu.cn

Victor C.M. Leung
Shenzhen University
Shenzhen, China
vleung@ece.ubc.ca

ABSTRACT

Data storage in peer-to-peer (P2P) games in a perfect applications scenario for blockchain. However, suffering from high transaction cost and latency, proof-of-work (PoW) becomes the bottleneck of blockchain games. Many attempts have been made to overcome various limitations of blockchain for P2P games, but many of them require modifying the game itself to be compatible with a blockchain solution. These overheads often bring new undesirable results to deal with. In this paper, we propose Proof-of-Play, a novel consensus model, to address these issues with a blockchain naturally integrated with P2P games, with minimum intervene to the game. The ultimate goal is to create a secure and fully decentralized architecture to transform a game being community-sustainable.

KEYWORDS

Blockchain; Security; Consensus Model; P2P; Games

1 INTRODUCTION

Peer-to-peer (P2P) games are among the most popular categories of multiplayer games, especially when multiplayer online battle arena (MOBA) games, such as Dota[1] and League of Legends[2], dominate recent video gaming market. P2P gaming architecture decentralizes a game network by having every player acts as the client and the server at the same time, such that all the game server hosting effort is distributed among players. P2P architecture receives a great deal of research attention [1][16] due to its high scalability, especially in a network intensive game genre like Massively Multiplayer Online game. However, pure P2P gaming architecture with distributed data storage is rare, due to the vulnerability to cheating behaviors in decentralized P2P data storage, in which every player is in control of some piece of game objects. Therefore, a centralized server is still required to save the data for the participating players, including account balance, battle records, etc.

On the other hand, the blockchain system [12] has introduced a decentralized, transparent and trustworthy platform, which is resistant to data modifications. Apparently, it is a natural fit for P2P games. The immutability of blockchain data makes it a perfect solution to the distributed data storage issue in P2P gaming, such to avoid various tampering issues like distorting player's combat historical records. In fact, the adoption of blockchain for data storage can also remove the single point of failure problem in P2P games, which means the whole gaming ecosystem can be sustained by the players' community rather than the game operator. In addition, by leveraging cryptocurrency driven by the blockchain, the participating players are able to use a unified, fine-granularity, and transparent token to stimulate the gaming ecosystem, including the incentives for data storage and in-game economics.

Nevertheless, the blockchain integration model for the P2P gaming system is yet to be investigated. A straight forward idea is to adopt a conventional public blockchain system, e.g. Ethereum[3] [4], as an external data storage. These blockchains are commercial platforms enabling immutable data writing and reading services. For example, CryptoKitties[4] [3], a web-based kitty collection game, utilizes Ethereum to store its gaming data. In particular, the virtual kitties can be purchased and traded through smart contracts [10] by the players, while all gaming data are synchronized in the blockchain after each operation performed by the players. However, the bottleneck of system performance is the cost and delay overhead for the data synchronization to the blockchain, which is introduced by the proof of work (PoW) [2] consensus model proposed in Satoshi Nakamoto's classic Bitcoin whitepaper[11].

The consensus model [14] is the key technique to keep independent parties in a blockchain to agree on the data that should be stored in the network. The purpose of the consensus model is to solve one of the major problem in a decentralized system, the Byzantine Generals' Problem [8]. The Byzantine Generals' Problem stated that a decentralized system must require a certain number of honest users, otherwise certain type of algorithm has to be implemented to guarantee a majority consensus on the decisions of the decentralized system. Satoshi's PoW approach requires participating nodes to compete for the privilege of writing blocks with each other in solving a puzzle, which is a mathematical calculation to scan for a numeric value whose hash value is smaller than a specific threshold. The computational difficulty of PoW reduces the collision of puzzle solver, thus, enforces a public consensus over the PoW winner to secure the majority consensus. Apparently, PoW is the primary cause of monetary cost and delay in a blockchain. To

*corresponding author
[1]http://www.dota2.com/
[2]https://na.leagueoflegends.com/
[3]https://www.ethereum.org/
[4]https://www.cryptokitties.co/

this end, many attempts on building a consensus model have been made.

In this work, we explore the similarity of the nature of P2P gaming system and blockchain, and investigate the possibility to leverage the gaming behavior as part of the consensus model in a blockchain. We dive deep into this idea to proposes Proof-of-Play (PoP), a consensus model for the blockchain-based P2P gaming system and evaluate its ability in keeping data integrity as a consensus model in comparison to other major consensus models.

The rest of this paper is organized as follows. We reviewed the related work of the gaming system with blockchain in Section 2 and presented the overview of the proposed PoP consensus model in Section 3. We then present the technical design and test-bed implementation in Section 4, and Section 5.1, respectively. Based on our development, the experiments are conducted to validate our system in Sections 5.2. A short case study of PoP is conducted against other major consensus models in Section 6. Finally, Section 8 concludes this paper.

2 RELATED WORK

2.1 Blockchain Systems

A blockchain system consists of blockchain data structure, consensus model and P2P network. The blockchain data structure, by definition, is a continuously growing chain of blocks, each of which contains a cryptographic hash of the previous block, a time-stamp, and its conveyed data [12]. The blockchain data structure is designed to resist modifications. With the help of P2P system and proof-of-work (PoW) [2] consensus model proposed in Bitcoin [11], the blockchain system can be utilized to support decentralized data synchronization, which becomes the foundation of decentralized ledgers. In order to add more values to the blockchain ecosystem, Ethereum [4] was implemented to facilitate decentralized smart contracts, which are immutable and transparent executable programs hosted by the blockchain. Nowadays, the blockchain-based decentralized applications (dApps) [13] have been extended to various areas, including initial coin offerings (ICO), social networks, networked games, and IoT.

2.2 General Consensus Models

As discussed in Section 1, PoW requires participating nodes to do useless mathematical works for the privilege of writing blocks, which brings the energy and time inefficiency issue to the blockchain systems. Therefore, a number of novel consensus models have been proposed as alternatives for general purpose blockchains.

Proof-of-Stake (PoS) [15] chooses the producer of the new block based on their stake on the network. For example, coin age is defined as the time of the coin left unspent, the higher the coin age of an individual, the more likely the individual will mine a new block. In other words, the richer an individual is, the more blocks the individual will mine in the blockchain. However, since holding tokens in different forks introduce no extra overhead for the stakeholders, PoS blockchains will spawn a large number of forks that reduce the value of the network. This is known as the nothing-at-stake problem.

Proof-of-Excellence [15] is a conceptual model mentioned in the PoS whitepaper. It stated that "a tournament is held periodically to mint coins based on the performance of the tournament participants, mimicking the prizes of real-life tournaments". Essentially, the node for the blockchain to hold consensus with is chosen via a game. However, in this model, good players will be more likely to win a game, this creates an unfair situation where good players will be able to write blocks repeatedly. So, the blockchain will become a partially centralized platform controlled by elite players.

Delegated Proof of Stake (DPoS) consensus model[5] solves the PoW overhead issue from another aspect: network participants delegate their rights of producing blocks to a small group of supernodes, which write blocks in turns for all users in the blockchain network. High throughput and low latency have been achieved in such a model. However, the public is still criticizing that DPoS being a partially centralized platform since it is impossible to prevent the supernodes from colluding with each other. A similar idea has been adopted by Proof of Vote (PoV) [9], which is coordinated by the distributed nodes controlled by consortium partners who come to a decentralized arbitration by voting. The key idea is to establish different security identity for network participants so that the submission and verification of the blocks are decided by the agencies' voting in the league without the depending on a third-party intermediary or uncontrollable public awareness.

2.3 Game-Specific Consensus Model

Since novel consensus models for general purpose blockchains are not yet accepted by the public, consensus models for specific verticals may leverage application features to improve the blockchain data synchronization. In this section, we summarize the approaches in the gaming domain.

Huntercoin[6] claims that around 80% of their coins are obtainable by collecting coins in a virtual universe which resides inside the blockchain. The platform provides a multiplayer game for the players to combat with each other in the map to collect coins. Huntercoin proposes the concept of Human (or AI) mining, and they can adjust the mining speed by increasing/decreasing the game difficulty over time. Similarly, Motocoin[7] has players to play a coin-collecting (the motocoin) game by driving a virtual motorbike. If a player finished the game before a targeted time, the player can write a new block on the blockchain along with the coin collection. The targeted time will be adjusted dynamically to maintain a consistent mining rate. These two consensus models rely on the players' effort in playing the games. However, the gaming progress is lack of entertainment but incentive driven only.

BUFF[8] proposed a Proof-of-Play consensus with another approach, where players earn token from playing games. The mining process is performed in the background, it does not interfere with the gameplay nor requires any expenses. From their whitepaper, the BUFF PoP has the player-base elect 21 players to vote for the consensus of the next block. The elected players are motivated to produce block since there are rewards. Also, their best interest is not to collude, since that harms the reputation of the blockchain and thus their stake on the blockchain (e.g. the value of their rewards).

[5]https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper
[6]https://huntercoin.org/
[7]https://motocoin-dev.github.io/motocoin-site/
[8]https://buff.game

This type of consensus model is intrinsically a DPoS consensus, and so their design is to have the consensus drove mainly by some specific players.

Motivated by these approaches, we propose another Proof-of-Play (PoP) consensus model. The PoP model acts as a data storage solution in P2P gaming and aims to interfere with the gameplay at a minimum. By simply playing in a P2P game, the blockchain runs and form consensus automatically, and players will naturally receive incentives to participate in the distributed data storage service of the game.

2.4 Blockchain Security

Decentralization security is of great importance in blockchain systems. This section features attacks that consensus models aim to solve.

2.4.1 Byzantine Generals Problem. In a decentralized system, every node has its version of the system. This raises the problem of forming consensus over the system, as every stakeholder may tell a different version of the system, cheating or not. This is the Byzantine Generals Problem [8]. The statement of the problem is rephrased as follows: There is a Byzantine army with generals that requires consensus on whether to attack or retreat. Within the generals, there may be traitors. How may an algorithm be designed to assure consensus can be made with traitors in the army? Several solutions have been made in the paper[8]. The solution, in general, either uses a signature on consensus, or relies on majority consensus. In blockchain, the solution is to implement a consensus model.

2.4.2 Sybil Attack / Eclipse Attack. Similar to the Byzantine Generals Problem, the Sybil Attack [5] is an attack related to dishonest nodes in a decentralized network. The statement is rephrased as follows: a single faulty entity can present multiple identities, thus is able to control a substantial fraction of the system and undermine the integrity of the system. The nature of the decentralized system is the anonymity of the users, thus it is hard to reveal the entity of identity in a decentralized network. However, the premise of the attack is the malicious party has the time and energy to disguise as multiple identities in the network. Techniques like PoW is known as economically secure since it requires a node to have the tremendous computational power to become a blockchain writer. Other technical techniques are also introduced, such as in Bitcoin, the number of outbound connection per IP address is regulated[9]. Eclipse Attack [6] is a variation of Sybil Attack, it states that the malicious party presents multiple identities to a single node to provide misinformation of the blockchain network.

3 SYSTEM OVERVIEW

The following section discusses the properties of the Proof-of-Play(PoP) consensus model. Then, a list of rules will be defined as a summary of the discussion above. Finally, the design of the system will be proposed to realize the rules.

[9]Bitcoin community on Sybil Attack: https://en.bitcoin.it/wiki/Weaknesses#Sybil_attack

3.1 The Need of the PoP

As mentioned in Section 1, blockchain is a perfect solution for data storage issues in P2P gaming. So, by having a game database powered by blockchain, data integrity in the P2P gaming system is enforced. Yet, many designs of the game may need to be changed to integrate into a blockchain naturally. For example, adding cryptocurrency to the game such that players are economically motivated to write blocks. However, any modification upon the game content is not an optimal solution. For example, the act of play in both Huntercoin and Motocoin (Section 2.3) becomes incentive-driven due to the blockchain, making the game progress lack entertainment. The Proof-of-Play aims to avoid these problems by having the consensus comes from playing naturally.

3.2 The Properties of the PoP

The PoP consensus model is proposed to integrate a P2P gaming system with a blockchain seamlessly. PoP by definition, is the act of play enables players to write blocks (data) to the blockchain. Then, by simply playing, new blocks will be written into the blockchain.

Also, the concept of PoP fit nicely with the trustless aspect of a blockchain design. The following properties can be secured by PoP:

(1) Users tend to be honest (Since gaming is time-consuming)
(2) Users are motivated to keep the blockchain reputable (Since users are the stakeholder of the blockchain)

With this design, blockchain can be integrated into a P2P gaming system seamlessly without modifying the game for the use of a blockchain. Also, without the separation of miners and users, the intention of the users of the system remains simple: they play the game and run the blockchain because they want to, not because of external motivation e.g. cryptocurrency.

3.3 Breakdown of PoP System Design

There are two system components to realize the properties of PoP:

(1) The integrity of the data representing the act of play
(2) The block writing process is an act of play

The first property corresponds to the data integrity of the P2P gaming architecture, and the second property corresponds to the PoP blockchain. As such, the flow of the system is designed as in figure 1.

To elaborate figure 1, assume a game of chess for player group A and player group B. The players finished their game and form a consensus on their game result as game data. Then one of the players of each group broadcasts the game data to the blockchain. The PoP will validate the game data integrity and rate the game for each game. The rating process then determines if any player can be a candidate block's writer. In this figure, a player in Group B will be a candidate block's writer, then the player successfully write the next block to the blockchain.

Note that the game data need not be a game result, but could be a state of a game at any point of time.

4 SYSTEM DESIGN

The following section is to propose techniques to realize the design in Section 3.3.

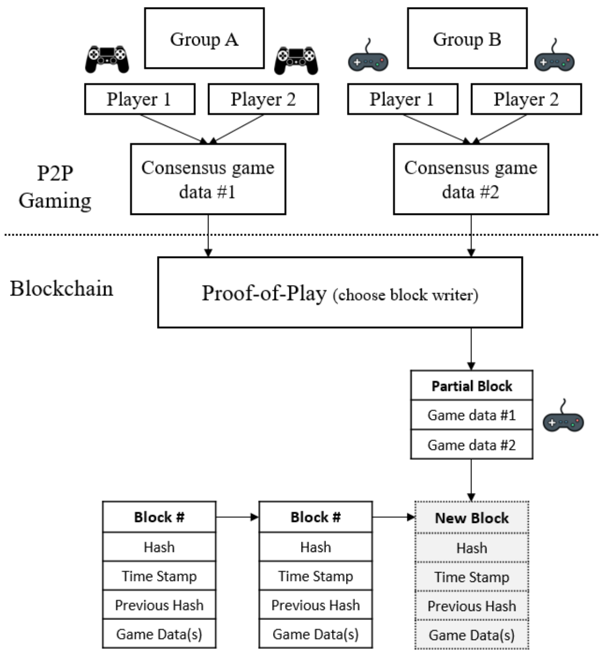

Figure 1: Proof-of-Play overview

4.1 Game Data Integrity

By keeping the integrity of the game data, the act of play is then a valid representation as PoP (section 3.2). In this section, we assume mechanism to discourage cheating behavior is implemented in the game (e.g. anti-cheat, idle detection etc), so we will focus on assuring the data integrity from players' collusion.

As shown in figure 1, the consensus game data is formed based on an agreement of the players in a game. However, If there is a malicious party forges a game result that is beneficial for everyone to agree with, then the game result will simply be an outcome of players' collusion instead of an act of play. Thus, the game data integrity is compromised and will disrupt the reputation of the blockchain. Although this is not the best interest of a node in a PoP blockchain (section 3.2), the following technique is proposed to reduce the risk of players' collusion in such way.

The idea of Shared Turns[7] has been discussed to have two players reveal their move simultaneously, so that a player cannot know another player's move in advance. This is a Commitment scheme and can be used to avoid players' collusion by having players to reveal their version of the game result simultaneously. The goal is to enable consensus of the majority players under Commitment scheme, demonstrated in figure 2. This process corresponds to the consensus game data shown in figure 1. The following describes the flow of the Commitment scheme:

(1) obtain everyone's public key
(2) players hash the game result (becomes a game hash)
(3) players create a signature with the copy of game hash using their private key
(4) players broadcast the game hash and the signed game hash

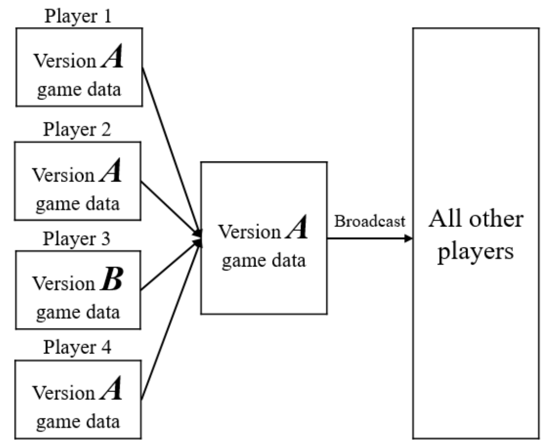

Figure 2: Proof-of-Play Shared Turns

(5) after players has received everyone's broadcast, broadcast own's game result
(6) players verify the game result by the corresponding public key and game hash
(7) determine the most competitive player by the consensus game result
(8) the most competitive player is responsible to broadcast the game result along with the signatures

After the process, the final game result is agreed by the majority of players. The process is further elaborated as below:

- In step 3, the game hash is signed, so the received game hash in step 5 is authenticated by the signature's owner.
- In step 4, players broadcast their game result. Malicious players may broadcast their game result based on others' game result. This opens up an opportunity to collude. However, since the game result is hashed, raw game result from other players is not known until step 6, so to avoid the aforementioned cheating behavior. If a player still decides to broadcast a raw game result different than the hashed game result after the first broadcast, other players will know immediately that the player is telling lie by comparing the hash value of both received game result.

As such, players are most likely broadcast a truthful game result, the act of play.

At step 7 and 8, the most competitive player will broadcast the game. Competitive can be defined as, for example, the highest rated player of a game. This step is to facilitate the design of the next section 4.2. The design in the next section ensures the PoP process is an act of play. Only the competitive player will be a block writer, this enforces the PoP since players compete for block writing.

4.2 Block Writing Procedure

With the process above, the integrity of the game data is secured and the act of the play is presented. Then, the last step is to ensure the block writing procedure is an act of play (section 3.3), so that the blockchain integrates into a P2P gaming system naturally (section 3.2).

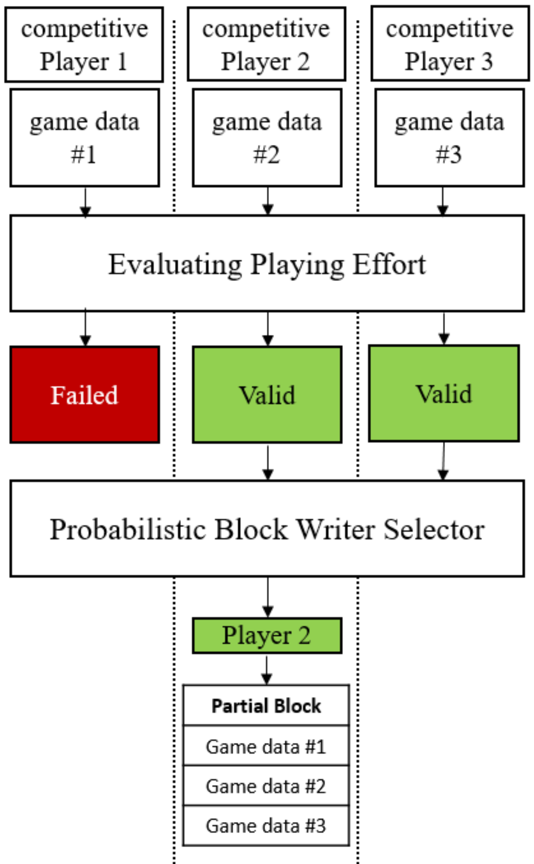

Figure 3: Block Writing Procedure in Proof-of-Play

The design of the block writing procedure is shown in figure 3, and is corresponding to the "Proof-of-Play" block in figure 1. Firstly, to realize the block writing process as an act of the play, one has to evaluate if the player of the game is paying enough effort. So, for a player to become a candidate block's writer, the player must have paid enough effort in the game data the player is presented in. Then, after the list of candidate blocks' writer are established, one candidate block's writer has to be selected as the final block writer for the next block. (Note that the term "competitive player" in figure 3 indicates the player is the representative of the game. This is to enforce the rule "the act of play" being the consensus of the blockchain as mentioned in section 4.1)

The final block writing process is designed to be probabilistic to reduce collision in the blockchain network, known as a fork. Fork stated that some of the nodes of the blockchain recognize a node as the next block writer, while other nodes recognize another node as the next block writer. If the block writing process is deterministic, it will function the same as "Evaluating Playing Effort" in figure 3 due to the nature of PoP. Then:

(1) if average (majority of) players can normally pass the evaluation, there will be too many valid block writers
(2) if only good players can normally pass the evaluation, this is an unfair advantage to lower-skilled players

Even if the evaluation is dynamically adjusted according to the individual's ability, the result of the adjustment will be classified between one of the two cases above (too easy / hard to pass), thus a deterministic approach is not feasible.

Note that the probabilistic selection process is not part of the mining. The mining has already happened at the process of "Evaluating Playing Effort", and the probabilistic selection works similarly to a random access protocol in the Data Link Layer of the OSI model. It is to avoid a burst in a number of candidate blocks' writer that massively forking the blockchain network. So, the act of play of the players is still where the blockchain nodes agree to form a consensus with.

In the current PoP design, the evaluation is adjusted according to the player's ability. This is to provide a fair chance for everyone to pass through the evaluation stage with enough effort of their ability.

With the proposed PoP process, a block writing procedure with minimum intervene to the P2P gaming system is designed.

4.3 Security Concerns

The main idea of the PoP system is to have human cognitive work in the block mining process. Once the presence of the cognitive work is assured, then many of the attacks rely on multiple identities will become harder, because users are miners (section 3.2), and mining needs an actual human. Then, multiple-identities-related attacks require multiple humans to perform, so attacking the blockchain is harder. To elaborate this idea, the following section discusses some major security problems in the blockchain, and evaluate PoP based on the problems.

4.3.1 Byzantine Generals Problem and PoP. The PoP solves Byzantine Generals Problem in a way similar to PoW. In PoW, Byzantine Generals Problem is solved[10] by having generals agree that the PoW winner holds the decision to form a consensus with. Then, since solving the puzzle in PoW is difficult (section 1), collision in solving the puzzle at the same time is rare. Thus, every general can follow the order of the single PoW winner. Even if there is an unfortunate collision, the generals simply have to perform PoW multiple times to the point that collision is almost impossible to happen. Then, a final PoW winner can be chosen to form a consensus with for the blockchain.

In PoP, Byzantine Generals Problem is solved by having generals agree that the PoP winner is the node to form a consensus with for the blockchain. Suppose there are n generals that love to play chess, they decided that whoever wins a game of chess against one of the generals has a chance to be a PoP winner. For each winner, PoP will rate the chess game according to the general's ability, to determine if enough effort has put into winning the game. Then, the winners will throw a dice to see if they can be the final PoP winner. If no winner or multiple winners have presented, everyone starts a game of chess against one of the generals again. Note that the expected value of the dice getting a final PoP winner should be $\frac{1}{n}$, so it is expected that every general just have to finish one game of chess to successful form consensus for the blockchain.

[10]the original Bitcoin mailing list on Byzantine general's problem: http://www.metzdowd.com/pipermail/cryptography/2008-November/014849.html

4.3.2 *Sybil Attack / Eclipse Attack and PoP.*

Sybil Attack states that a single faulty entity can present multiple identities to control a substantial fraction of the system [5]. In a PoP system, players have to be competitive to be able to write a block. If the game is complex enough to represent a human cognitive work, then creating multiple identities will be difficult, since one has to assure the quality of the multiple identities (i.e. competitiveness) to control a portion of the system.

Eclipse Attack in a PoP system is still easier to perform than a Sybil Attack, since the attack only targets a single node. However, PoP does not make a system more vulnerable toward the attack. Since it is still a multiple-identities-related attack, the requirement of having a better quality of multiple identities still a barrier for the attacker.

4.3.3 *Other Security Concerns.*

There are many other different types of attacks that are not covered specifically in the above section. In the below sections, the security properties of PoP is summarized to provide an overview of the secured concern.

4.3.4 *Other Security Concerns: Anonymity.*

One of the properties of a blockchain is the anonymity of the users. However, as stated in the paper[5], one can claim as many electronic personas as one has the time and energy to. In blockchain, the anonymity properties make a human easier to disguise itself with multiple identities. The PoW consensus model solves this problem by having a node to invest in huge computational resources to become a miner, making a PoW blockchain "economically secured".

In PoP, rather than punish or reward the users, an identity of a node is formed with the anonymity property of a blockchain intact. By assuming playing is a human-exclusive cognitive work, the mining process of a node must involve an actual human. So, PoP is intended to maintain a one-to-one relationship of a miner and a human, limiting the ability for a human to disguise itself with multiple identities (i.e. multiple miners).

4.3.5 *Other Security Concern: Data Integrity.*

Another property of a blockchain is data integrity. The blockchain is designed to resist against modification[12]. To maintain data integrity, the source of the data has to be correct. Then, changes made to the blockchain has to be assured to be irreversible.

In PoP, not all possible threats of the integrity of the source of data are covered. In section 4.1, the shared turn process is designed against data integrity problem due to collude. However, if the game allows players to connect to other players directly, a colluding party can be formed to broadcast game data without the effort of playing. Although this is not the best interest of a PoP user (section 3.2), the possibility of this compromises the PoP system. As such, an anti-cheating mechanism has to be agreed by all players as the genesis of the PoP blockchain. This will solve the colluding problem above and many other data integrity problem. PoP only assures a correct source of data holds its integrity under the P2P system.

For the property of irreversible modification, same as other consensus models (e.g. the PoW), the mechanism to counteract attackers trying to modify the data is to make rewriting the data very costly. For example, in PoW, attackers need to possess 51% of the network computational resources to outrun other computers in solving the puzzle to write a new block. This is economically very difficult to compete against the computational resources of all other users combined.

In PoP, the irreversible property is assured by assuming playing requires effort, similar to solving a puzzle. Then, the rest of the operation is same as section 4.3.4: an actual human is behind the mining, so gathering a group of competent human to compete against all honest human using the blockchain is very difficult.

5 SYSTEM IMPLEMENTATION

The following section introduces the implemented system and the experiment on it. There are 2 separate implementations for this research, a demo for the PoP architecture flow (figure 1), and the PoP mining simulation.

The objective of the architecture flow demo is to demonstrate the actual process to implement the system, and the objective of the experiment is to test the system stability.

5.1 Architecture Flow Demo

The flow demo consists of the Shared Turns implementation (figure 2) and the PoP implementation (figure 3). For the PoP implementation, the "Probabilistic Block Writer Selector" part is not implemented, but it will be tested in the next section (section 5.2) with a simulation program.

The repository contains the complete library with a folder of example scripts of Shared Turns demo individually, PoP demo individually, and both of the demos combined together. The demo is to let interested developers understand roughly what to and how to implement the blockchain and the shared turns. The repository has a comprehensive README file to explain the structure of the repository.

5.2 PoP Mining Experiment

In section 4.2, a probabilistic approach to select the block writer is described to ensure the robustness of the PoP system. The following section will discuss the importance of the mining process, perform experiments for the system stability test, and present the experiments' result.

5.2.1 *The details of probabilistic mining.*

As explained in section 4.2, the probabilistic approach for PoP mining is to reduce the probability of forking in the blockchain. However, several parameters need to be defined for a complete probabilistic approach.

To begin with, the implementation of the guessing puzzle in Bitcoin PoW[11] is explained as an idea of probabilistic technique in a decentralized system.

Bitcoin PoW has users to guess a number fulfills a criteria: a number below the "Target value". For example, if the "Target value" is 10, then miners have to guess a number lower than 10. With this concept, "Difficulty" can be defined. There is a "maximum target value" for Bitcoin (i.e. 2^{224}). So, the "Difficulty" is defined as:

$$\text{Difficulty} = \frac{\text{Maximum target}}{\text{Target value}} \tag{1}$$

where Difficulty = 1 is the easiest Difficulty, the higher the Difficulty value, the harder the number to guess.

[11] Bitcoin Difficulty: https://en.bitcoin.it/wiki/Difficulty

Then, to generate random number, Bitcoin's PoW use hashing (i.e. SHA-256 algorithm). Since a good hashing function is a good random function, by having miners input values into the SHA-256 algorithm, a hash value (hexadecimal number) will be generated, the hashed value can be used to guess a valid "Target value". In Bitcoin PoW, the possible combination of hexadecimal number output is 2^{256} using SHA-256. So, the probability of finding a valid number (below the "Target value") is as following:

$$P[\text{Getting a valid number}] = \frac{\text{Target value}}{\text{Range of hash value}} \tag{2}$$

where "Range of hash value" is 2^{256}. Then, by substituting the Equation (1) to Equation (2), the P[Getting a valid number] in Bitcoin PoW can be alternatively expressed as:

$$P[\text{Getting a valid number}] = \frac{\text{Maximum target}}{\text{Difficulty} \cdot \text{Range of hash value}}$$

Since "Maximum target value" is 2^{224}, and "Range of hash value" is 2^{256}, it can be rewritten as:

$$P[\text{Getting a valid number}] = \frac{1}{\text{Difficulty} \cdot 2^{32}}$$

Then, by definition, the "Expected number of hashing to get a valid number" is:

$$\text{Difficulty} \cdot 2^{32} \tag{3}$$

Since the "Expected time between mining each block" (Confirmation Time) in Bitcoin is 10 minutes, suppose we know the "History number of hashing per block" and the "History confirmation time", By Equation (4)

$$\text{Hash rate} = \frac{\text{Number of hashing per block}}{\text{Confirmation time}} \tag{4}$$

where the term "History confirmation time" means "the average of n previous actual confirmation time", the same applies for "History number of hashing per block". It can be written in such way to calculate the "Expected number of hashing of a block":

$$\frac{\text{Expected number of hashing}}{\text{Expected confirmation time}} = \frac{\text{History number of hashing}}{\text{History confirmation time}} \tag{5}$$

thus, the "Expected number of hashing" of a block can calculated using equations 5, and a mathematically sensible "Target value" can be derived by backtracking from equation 3.

5.2.2 PoP Probabilistic Mining. In PoP, the same mathematical approach has implemented as a probabilistic selector (figure 3). Since the hashing operation happens much less frequently than PoW, the confirmation time is also different than the one in Bitcoin (i.e. 10 minutes). So, the following experiment is conducted to observe how changes in different variables in the calculation brings impact to the stability of the system.

A simulation program has been written to simulate the mining process. A few adjustments have been made to the calculation in section 5.2.1, customized for the mining process in PoP. Below shows the adjustment made to the calculation:

We want the Proof-to-Play to have an average confirmation time dependent on the time length of a game match (For a non-match-based game, the time can be arbitrarily defined). For example, when a player has played n matches, a block will be mined. so the "Play Effort" in the "Evaluating Playing Effort" block (3) is the average score of a player winning n matches. Assume all players win a match at the same time, the desired probability of getting a valid number of any time is:

$$P[\text{Getting a valid number}] = \frac{1}{\text{Number of players} \cdot n} \tag{6}$$

Since the number of players is defined for a simulation, using equation (2), the initial target can be calculated as follows:

$$\frac{1}{\text{Number of players} \cdot n} = \frac{\text{Maximum target}}{\text{Difficulty} \cdot \text{Range of hash value}}$$

$$\frac{1}{\text{Number of players} \cdot n} = \frac{\text{Target value}}{\text{Maximum target}} \cdot \frac{\text{Maximum target}}{\text{Range of hash value}}$$

$$\text{Target value} = \frac{\text{Range of hash value}}{\text{Number of players} \cdot n} \tag{7}$$

For the actual implementation, n is $\frac{\text{Expected confirmation time}}{\text{Average match time}}$. n is not fixed since n is dependent to the experiments i.e. experiment on change of confirmation time will be conducted.

After the initial target, the first block will be mined. Then, the hash rate of the first block can be known. So, after the first block, the adjustment of the target is made by m previous hash rate using equation (5). The mean of m previous hash rate is taken to represent the blockchain network history hash rate.

For the maximum target, we assume that it is dependent on the confirmation time, since the function of target value is to govern a stable confirmation time when the blockchain scales. As a lower confirmation time means a lower difficulty, we assume a negative linear relationship between the target and the confirmation time (i.e. the lower the confirmation time, the higher the target value). Then, by using Bitcoin average confirmation time (600 seconds) and the base-2 exponent of its maximum target (224), the maximum target of any confirmation time is defined as:

$$\log_2(\text{Maximum target}) = 256 - \frac{\text{Confirmation time} \cdot (256 - 224)}{600} \tag{8}$$

5.2.3 Experiment Implementation. The experiment flow is shown as below:

(1) create a manager process with n players' process
(2) manager process defines and calculates the parameters of the blockchain. The parameters include:
  (a) initial block index = 0
  (b) number of players
  (c) expected confirmation time
  (d) maximum target value
  (e) initial target value
  (f) number of blocks (mining simulation loops)
(3) manager process tells players' process to start mining

(4) players wait 5 seconds on average to simulate playing a match.
   (a) 100 samples of match time are generated with a normal distribution of 5 seconds mean and 1 second standard deviation
(5) if no longer chain was received from the manager process, the player will continue the mining process by hashing values, including the string of his game score, a random number, and a nonce
(6) if the hashed value is lower than the target, broadcast to the manager, otherwise do nothing, then go to step 4

throughout the steps above, the manager will keep waiting for a valid new block, calculate new target, store the history blockchain states, and broadcast the new blockchain.

The concept of manager process is used in the experiment, to simplify the workload of player's process, so that the experiment mainly test the mining speed and stability of the player's process over adjusting targets. Inter-process communication is handled by the manager process entirely.

5.2.4 Experiment Variables. The variables below are the experiment's subject. By changing the value, we would like to observe its impact on the stability of the PoP system. The experiment's subject is shown as below:

(1) expected confirmation time
(2) the number of miners (players)
(3) number of history blocks as reference of history hash rate

For the third variable, given number m, the mean of m previous hash rate will be obtained as the history hash rate for the calculation of the current block.

The 3 variables are given as the experiment input. These three variables determine the adjustment of the "target value". The first and the third variable directly involves in calculating the target of each block, and the second variable affects the calculation of the target value via the history hash rate. Stability of the blockchain network can be observed via the changes in the variable correspond to actual confirmation time in the experiment data output. We can then conclude the practicality of the methodology and assumptions made on deriving the calculation for the PoP mining.

5.3 Experiment Results

The following three experiments have been conducted. The experiment subjects are intended to adjust the target value accordingly, and the side effects of the adjustments are to be observed, interpreted and to make conclusions.

The mining in the experiments are conducted using a single AMD Ryzen 1400 CPU, GPU is not used for the experiments. So, the values to be experimented are chosen based on the CPU performance. The details will be explained prior discussing the experiment results.

5.3.1 Expected Confirmation Time. This experiment is set with the following parameters:

- number of blocks = 100
- number of players = 10
- number of reference history blocks = 10
- confirmation time: experiment subject

The number of adjustments is chosen to be 7 according to the number of processes (from python multiprocessing library) the CPU can handle simultaneously under maximum usage.

The intended evaluation of this experiment is the accuracy in target value by the actual confirmation time, so a range of short and realistic confirmation time is picked. In this experiment, the experiment values are defined as 10, 30, 60, 90, 150, 180 seconds. It is expected that other experiment values scale the same way the experiment values do.

The result of the experiment shows roughly an exponential distribution, where the "percentage of a block being mined" decrease exponentially as the "confirmation time" increases.

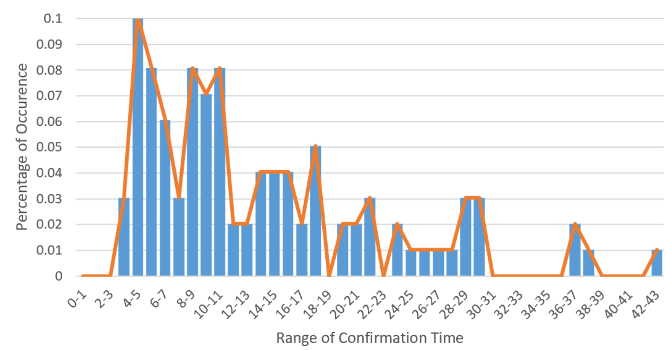

Figure 4: Occurrence of actual confirmation time under 10 seconds expected confirmation time

Figure 4 depicts the experiment result with a parameter of 10 seconds expected confirmation time. Similar to Bitcoin[12], it roughly shows an exponential distribution. It is expected that the variance of the confirmation time will be normalized as the number of trials increase (as shown in later experiments), so the exponential distribution will be more apparent.

So, the hashing operations in the mining process are properly implemented.

Parameter	μ	Range	$\tilde{\mu}$	σ
10	13.85591	40.92175	10.60783	9.272552
30	29.24059	131.9419	22.5276	23.24391
60	57.19258	222.9035	49.98521	45.61157
90	113.7163	532.5268	85.62569	107.2782
120	123.6353	765.8719	78.50864	134.1295
150	146.9611	784.2096	98.08061	147.5669
180	164.2757	958.4286	102.3475	164.5644

Table 1: Section 5.3.1 Experiment Result

Table 1 shows the experiment result of the 7 different expected confirmation time parameters with figure 5 showing the distribution of actual confirmation time in the experiment.

The notation of the table is defined as follows: Parameter is the expected confirmation time, Range is the maximum confirmation time from the sample minus the minimum confirmation time from

[12]Confirmation: https://en.bitcoin.it/wiki/Confirmation

the sample, σ is the standard deviation of the sample, the μ is the Mean and the $\tilde{\mu}$ is the Median, both calculated from the result of actual confirmation times.

The intended effect of an increase of the parameter is the increase of the Mean of the sample confirmation time. The Mean is close to the parameter, indicates the calculation for target adjustment described in section 5.2.2 is efficient. The approximation of the Mean and the σ also indicates the efficiency of the calculation, since an exponential distribution has the same value for both mean and σ.

The effect of the increasing parameter is the difference between the mean and median increases, where median < mean. The interpretation here is the large divergence between the data range and the mean. Data Range increases since the target value is lower as the parameter increases. As the target value is lower, the difficulty is higher, making for some blocks it takes much longer to hash a valid target. These small cases of very large confirmation time affect the mean, so the mean moves further away from the median as the parameter increases. In other words, the λ (rate parameter) in the exponential distribution increases as the parameter increases.

Developers have to make choices of the parameter based on the design of the blockchain. In general, the higher the parameter, the better the blockchain stability, since more percentage of the blocks will have an actual confirmation time before the Mean. This can be observed by examining the Peak Frequency of each expected confirmation time at Figure 5: both 10 and 30 seconds of expected confirmation time has its peak frequency at sample portion of 10% ($0.1<n<0.2$), the rest of the expected confirmation time has its peak frequency at sample portion of 0-10% ($0.0<n<0.1$).

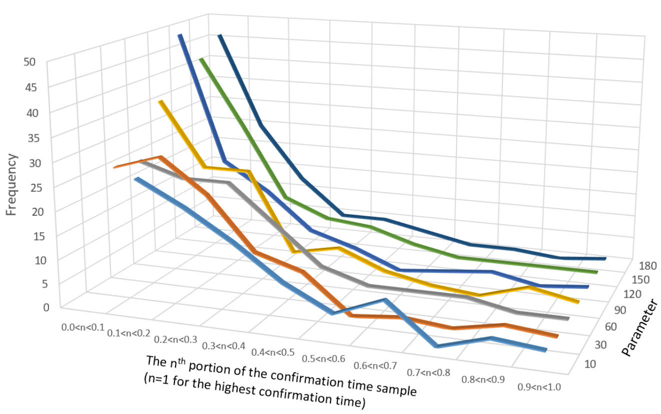

Figure 5: Frequency distribution of the sample confirmation time (Experiment Section 5.3.1)

5.3.2 Number of reference history blocks. This experiment is set with the following parameters:

- number of blocks = 1000
- number of players = 10
- number of reference history blocks = experiment subject
- confirmation time: 10

The intended evaluation of this experiment is the variance of the target value. A sudden change in history hash rate will easily affect the target value by making the blockchain very difficult or very easy all the sudden. Increasing the number of reference blocks for the history hash rate neutralize the effect of the sudden change in history hash rate to the target value (i.e. concept of moving average).

The number of blocks has been set to 1000 to allow an adjustment to a larger value of the experiment subjects. The number of 1000 blocks is picked since even with a low confirmation time 10 seconds, the distribution appears to be more of an exponential distribution than figure 4 (as the variance of the confirmation time is normalized). The exponential distribution makes sure the target value is effective first. Then the improvement in target value can be evaluated. The adjustment is defined as the following:

$$n = \begin{cases} 2(n-1), & \text{if } x \geq 1 \\ 10, & \text{otherwise} \end{cases}$$

The number of adjustments is chosen to be 7 according to the number of processes (from python multiprocessing library) the CPU can handle simultaneously under maximum usage. The adjustment is defined as above function to experiment with a range of a low number of reference blocks, to the maximum reference blocks the blockchain is allowed, while having a reasonable experiment duration length to introduce enough variance of the data: average 2.7 hours per process.

Similar to section 5.3.1, it is expected that other experiment values scale the same way the experiment values do.

Parameter	μ	Range	$\tilde{\mu}$	$\tilde{\mu}_{loc}$	σ	α
10	14.55	114.42	11.56	0.10	10.45	166
20	14.74	93.52	11.44	0.12	10.61	-66
40	14.77	63.50	11.77	0.18	10.24	-142
80	14.14	80.14	11.38	0.14	9.75	-154
160	14.14	80.14	11.38	0.14	9.75	-154
320	13.92	64.75	11.17	0.17	9.21	-178
640	14.34	54.33	11.35	0.20	9.32	-175
1000	14.22	71.24	11.49	0.16	9.66	-193

Table 2: Section 5.3.2 Experiment Result

Part of the table 2 evaluates the same notation of the experiment result as table 2 does, the notation of the table has already been explained in the section 5.3.2 when table 2 is presented. In this table, the "Parameter" means "Number of reference history blocks" instead, $\tilde{\mu_{loc}}$ means the location n of the Median of the Parameter (shown in figure 6), and α means the difference of y (Y-axis) between $n = 0.0<n<0.1$ and $n = 0.1<n<0.2$ (X-axis) in figure 6.

Figure 6 in this section has the same X-axis and Y-axis as figure 5. So, both figures can be interpreted similarly.

The intended result of an increase of the parameter is the decrease of the Data Range, since the history hash rate used in Equation (5) is now based on a larger portion of the full history hash rate. The experiment result shows a decrease in the data range, so the experiment is effective. The consistency of the Mean also indicates the calculation in section 5.2.2 is effective.

From figure 6, a shift of the distribution (by observing the x with peak y) with the increase of parameter is showed. This shift

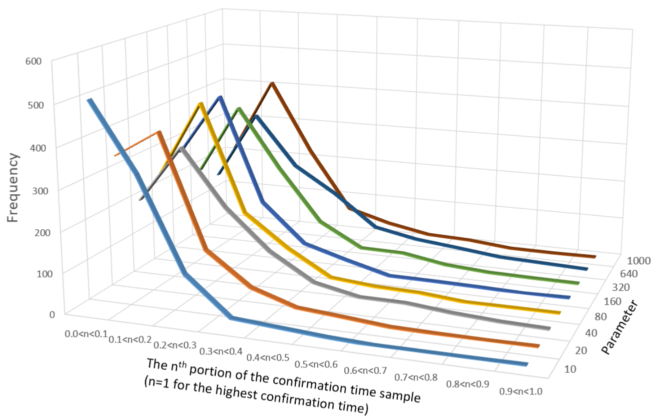

Figure 6: Frequency distribution of the sample confirmation time (Experiment Section 5.3.2)

is expressed in α in table 2. To recap, α means the difference of y (Y-axis) between $n = 0.0<n<0.1$ and $n = 0.1<n<0.2$ (X-axis). To interpret, since the data range decreases, outliers like extremely large values are not present anymore. That makes the Median shifts towards the right. Thus the increasing value of $\tilde{\mu}_{loc}$.

Thus, the higher the number of reference history blocks, the more reliable the target value is. This makes the blockchain algorithm resists to a sudden burst or cuts of hash rate, increasing the stability of the blockchain network.

5.3.3 Number of Players. This experiment is set with the following parameters:

- number of blocks = 100
- number of players = experiment subject
- number of reference history blocks = 10
- confirmation time: 10

The intended evaluation is the relationship between the number of players with the target value. So, the number of adjustments to the experiment subject, and the values of the experiment subject are defined below.

$$n = \begin{cases} 2(n-1), & \text{if } x \geq 1 \\ 10, & \text{otherwise} \end{cases}$$

There are 6 adjustments in total. Both the number of adjustment and the adjustment itself are picked according to the CPU capability. Running 320 python multiprocessing process needs a lot of processing power from the CPU, so a lesser number of adjustments is chosen compare to previous sections. Also, similar to section 5.3.1, it is expected that other experiment values scale the same way the experiment values do. So, a lesser number of adjustments for the experiment will still be an effective experiment.

Part of the table 3 evaluates the same notation of the experiment result as table 1, the notation of the table has already been explained in the section 5.3.1 when presenting the table 1. In this table, the "Parameter" means "Number of Players" instead.

Figure 7 in this section has the same X-axis and Y-axis as figure 5 and figure 6. So, all three figures can be interpreted similarly.

Parameter	μ	Range	$\tilde{\mu}$	σ
10	13.85591	40.92175	10.60783	9.272552
20	15.39057	57.12786	11.3413	11.19832
40	16.12811	65.08809	13.89211	10.4721
80	16.1453	68.46416	12.51086	12.20591
160	15.1577	58.9716	11.52297	9.714505
320	17.565	79.87559	12.82916	14.37079

Table 3: Section 5.3.3 Experiment Result

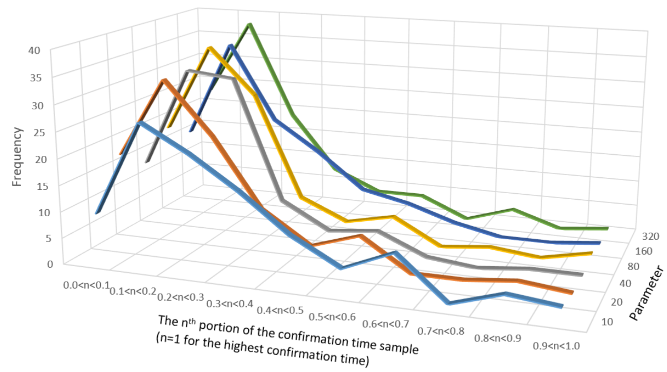

Figure 7: Frequency distribution of the sample confirmation time (Experiment Section 5.3.3)

The intended effect of an increase in "Number of Players" is the decrease in target value. Same as section 5.3.1, Data Range increases as the parameter increases due to a lower target value. However, the degree of the increase in Data range is lesser than the one in table 1. This concludes the indirect relationship of the "number of players" affecting the target value via the history hash rate in a smaller degree.

The Mean has a slight increase as the parameter increases, creating a divergence with the Median. In section 5.3.1, it has been concluded that it is due to the increase of λ parameters in the exponential distribution. However, in figure 7, we can observe that the central tendency does not shift.

The slight increase in Mean is possibly contributed by the block propagation time. The block propagation time means the delay in receiving the latest data of the blockchain. In this case, since there are a large number of players, and the broadcast is managed by a manager process, some of the processes have a slight delay in receiving the new block. Assume node x is the current block producer. Since the history confirmation time comes from the mining time of node x, then if node x has a huge delay in receiving the last block, before node x started mining, some of the nodes have already mined for a while. Then, by Equation (4), the "confirmation time" is inaccurate (node x did not account of the nodes with a headstart in mining), therefore overestimating the hash rate. In our experiment, the history hash rate is also overestimated, the degree of overestimating the Mean for the parameters is ranked as 320, 80, 40, 20, 160, 10.

When the blockchain scale, it is inevitable for an increasing block propagation time. Bitcoin has defined a confirmation time of 10 minutes, and it has been generally accepted by the community that 10 minutes is a countermeasure for the problem of block propagation[13]. When the confirmation time is sufficiently long, even if a node has a 1-minute delay in receiving the latest block, the wasted effort of mining for the node is still acceptable.

In our experiment, although the block propagation time is artificial and can be fixed, in reality the situation is similar. It is difficult for every nodes to stay connected with all other nodes as the network scale, and there will be a chain of broadcast before every node is synchronized to the up-to-date blockchain. Developers have to consider the ideal expected confirmation time according to the expected scale of the blockchain network.

5.3.4 Experiment Conclusion. This experiment concludes that the adjustments of the three variables: "Expected Confirmation Time", "Number of reference history blocks", and "Number of players" can efficiently change the target value of the PoP mining algorithm. Both "Expected Confirmation Time" and "Number of players" have side effects on the system along with the increase of its value.

The increase of the "Expected Confirmation Time" increases the range of the sample confirmation time, and the increase of the "Number of players" affects the system stability due to the block propagation time.

"Number of reference history blocks" efficiently stabilize the target value, resists to sudden burst or cut in blockchain hash rate, this may act as a countermeasure of the side effects in the increase of the "Expected Confirmation Time".

However, for the side effect on the increase of "Number of players", it is up to the developers to compensate between an ideal expected confirmation time, or the scalability of the blockchain network.

The experiment can generally apply to any probabilistic mining approach similar to PoW, developers can make decisions on a blockchain system design based on the effect of adjusting the parameters above.

6 DISCUSSION

In this section, we discuss the nature of this consensus model, with the advantage and disadvantage of this model in comparison of other major consensus models mentioned in Section 2.2.

6.1 Nature of the PoP

The design of the PoP aims to decentralize a P2P gaming system without data storage issues (section 1). This model is not limited to the act of gaming, as the act can be anything as long as it is the use of the blockchain. So, the general rule of a PoP consensus model can be expressed as follows:

- The act of using the blockchain fulfills the consensus
- The representation of this act is not exploitable

Both of this rule is fulfilled by the block writing process (section 4.2) and the shared turn process (section 4.1) respectively. To apply this rule into other application, a metaphor of a cryptocurrency based on PoP is given:

- The money spent/received of an individual is the rating
- By spending / receiving money, the rating of an individual goes up, grants the individual the ability to try writing a block
- The money spent/received by an individual must be productive e.g. Gross Domestic Product (GDP) factor

6.2 PoP and other consensus model

This consensus model is similar to the conceptual model Proof-of-Excellence. However, the player in PoP need not be excellent, the player simply has to present its act of play to grant the chance of mining a new block. This avoids the issues of better players having an unfair advantage in the blockchain system.

While this system has adopted some of the ideas in PoS and PoW, some of the disadvantages of the system has been considered and has been eliminated under the PoP system. A summary of the comparison of Proof-of-Play to other models is shown in table 4.

6.2.1 PoW. The main cost of the PoW is the energy and time inefficiency(Section 2.2). Also, nodes with better computational resources will hash the valid number faster to other average nodes. This is especially true when a blockchain uses a basic Proof-of-Work algorithm, which the hash rate of a node is completely dependent on the node's CPU performance. So unfairness still exists in PoW.

In PoP, although probabilistic mining still exists, it functions similarly to a random access protocol (section 4.2). Also, it has an overhead of n game matches before the mining occurs and scale with the player population, depending on the developer's choice of expected confirmation time (equation (8)), the target can be defined such that the expected hash rate of the blockchain network is low. So, PoP is more power-efficient.

Also, the evaluation of the playing effort of a player 4.2 is adjusted dynamically according to the player's ability. A top player would not have a better chance in mining a block compared to an average player.

6.2.2 PoS. The nothing-at-stake problem of PoS in section 2.2 is elaborated as following: assume a fork in the network happens, the rational decision of a miner is to mine on both fork, since mining does not need any computational resources and miner will benefit no matter which fork is rejected by the majority of the blockchain in the end.

The nothing-at-stake problem opens up opportunities to launch security attacks. PoW does not have the nothing-at-stake problem, since the intrinsic cost of mining on multiple chains is the decrease in the chance of mining successfully. So, In a PoW system, miners are encouraged to mine on the same chain.

Also, PoS is not fair, since the more stake a node holds, the more likely the node will mine a block. A new node joins the network will never have a hash rate higher than older nodes in the network.

In PoP, the rule is the act of using the blockchain fulfills the consensus, there is no reason a community wants multiple version of game data. However, this is a weak assumption. To counteract this problem, similar to PoW, in-game rewards on successful mining can be implemented, such that the intrinsic cost of mining on multiple chains is the decrease in value of the player's in-game rewards (some chain do not acknowledge the player's rewards).

[13] https://medium.facilelogin.com/the-mystery-behind-block-time-63351e35603a

Model	Source of Consensus	Time and Energy Efficiency	Fairness in mining
Proof-of-Work	Computational Power	Low	Medium (algorithm dependent)
Proof-of-Stake	Stake	High	Low
Proof-of-Play	Using the blockchain	High	High

Table 4: Section 6.2 Summary of Proof-of-Play comparison

Also, as mentioned in Section 6.2.1, the chance of mining is designed to be more fair compare to PoW/PoS due to the dynamic adjustment of the "evaluation of the playing effort".

7 OPEN ISSUES

This paper presents an idea of the existence of a blockchain being the principle of fulfilling the properties of blockchain, that is, the act of play activates the consensus on a state of a peer-to-peer game. However, current work has several limitations to be a backbone system of a peer-to-peer game. Both Section 4.1 and Section 4.3 assumes the game can detect cheating behaviors. While there are existing works on topic of P2P anti-cheating, the questions still stand:

(1) can a PoP blockchain itself be cheat-proof on games (i.e. no assumption on the game design has to be made)?
(2) What quantitative ways can be used to show the strength of the security aspect if point (1) is possible?
(3) Do engineering techniques, like regulating connection of IP address in Bitcoin[14], is needed if point (1) not possible?

These investigations can be made to complete the Proof-of-Play as to be called an architecture of "peer-to-peer game driven by blockchain", or in other words, to realize a full blockchain-game.

8 CONCLUSION

This paper introduces a consensus model P2P gaming system using blockchain as a solution to data storage issues. The consensus model aims to create a blockchain system that forms a consensus by the use of the blockchain itself, while not compromising the general properties of a blockchain. Then, the system is implemented to demonstrate the flow of the PoP, experiments have conducted to show how different parameters affect the stability of the PoP system or probabilistic mining system in general. Finally, this paper generalizes the consensus model and discuss the differences between PoP and other major consensus models.

We believe this design would bring more attention on blockchain system related to the P2P gaming system. This also acts as a design reference on blockchain in interactive system, eventually decentralize any interactive system reliably with a simple design nature like PoP: the use of a blockchain form consensus for the blockchain.

REFERENCES

[1] BETTINA Kemme AMIR Yahyavi. 2013. Peer-to-Peer Architectures for Massively Multiplayer Online Games: A Survey. *Comput. Surveys* 46, 1 (October 2013). https://doi.org/10.1145/2522968.2522977
[2] Adam Back. 2002. Hashcash - A Denial of Service Counter-Measure. (09 2002).
[3] Blockchain Technology 2018. BLOCKCHAIN GAMES: A SURPRISING NEW PLAYER IN THE INDUSTRY. (March 2018). http://bitcoinist.com/blockchain-games-a-surprising-new-player-in-the-industry/.

[14]Bitcoin community on Sybil Attack: https://en.bitcoin.it/wiki/Weaknesses#Sybil_attack

[4] Vitalik Buterin. 2013. Ethereum white paper: a next generation smart contract & decentralized application platform. *https://github.com/ethereum/wiki/wiki/White-Paper* (2013).
[5] John R. Douceur. 2002. The Sybil Attack. In *Peer-to-Peer Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 251–260.
[6] Aviv Zohar Sharon Goldberg Ethan Heilman, Alison Kendler. 2015. Eclipse Attacks on BitcoinâĂŹs Peer-to-Peer Network. In *Proceedings of the 24th USENIX Conference on Security Symposium*. USENIX Association Berkeley, CA, USA, 129–144.
[7] Daniel Kraft. 2016. Game Channels for Trustless Off-Chain Interactions in Decentralized Virtual Worlds. *http://ledger.pitt.edu/ojs/index.php/ledger/article/view/15* (2016).
[8] Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine Generals Problem. *ACM Trans. Program. Lang. Syst.* 4, 3 (July 1982), 382–401. https://doi.org/10.1145/357172.357176
[9] K. Li, H. Li, H. Hou, K. Li, and Y. Chen. 2017. Proof of Vote: A High-Performance Consensus Protocol Based on Vote Mechanism amp;amp; Consortium Blockchain. In *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. 466–473. https://doi.org/10.1109/HPCC-SmartCity-DSS.2017.61
[10] B. K. Mohanta, S. S. Panda, and D. Jena. 2018. An Overview of Smart Contract and Use Cases in Blockchain Technology. In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. 1–4. https://doi.org/10.1109/ICCCNT.2018.8494045
[11] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. *White Paper: https://bitcoin.org/bitcoin.pdf* (2008).
[12] Michael Nofer, Peter Gomber, Oliver Hinz, and Dirk Schiereck. 2017. Blockchain. *Business & Information Systems Engineering* 59, 3 (01 Jun 2017), 183–187. https://doi.org/10.1007/s12599-017-0467-3
[13] Siraj Raval. 2016. *Decentralized applications : harnessing Bitcoin's blockchain technology* (1 ed.). O'Reilly Media.
[14] L. S. Sankar, M. Sindhu, and M. Sethumadhavan. 2017. Survey of consensus protocols on blockchain applications. In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. 1–5. https://doi.org/10.1109/ICACCS.2017.8014672
[15] Scott Nadal Sunny King. 2012. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. (August 2012).
[16] Youki Kadobayashi Takuji Iimura, Hiroaki Hazeyama. 2004. Zoned Federation of Game Servers: a Peer-to-peer Approach to Scalable Multi-player Online Games. In *3rd ACM SIGCOMM workshop on network and system support for games*. 116–120. https://dl-acm-org.ezproxy.lb.polyu.edu.hk/citation.cfm?doid=1016540.1016549