A Game-theoretic Approach to Storage Offloading in PoC-based Mobile Blockchain Mining

Suhan Jiang
Temple University
Suhan.Jiang@temple.edu

Jie Wu
Temple University
jiewu@temple.edu

ABSTRACT

Proof of Capacity (PoC) is an eco-friendly alternative to Proof of Work for consensus in blockchains since it determines mining rights based on miners' storage rather than computation. In PoC, for every block, a miner executes hashing on part of his dedicated storage. The miner that comes up with the smallest hash value among all miners will win the block. PoC has yet to be applied to mobile applications, due to the storage limitation of mobile devices. *Storage offloading* can be a viable solution that allows miners to offload mining all files to a cloud storage. In each mining round, a miner can decide whether to mine on his local device or by a cloud virtual machine (VM). *Self-mining* requires no extra cost but it incurs download delay, which will reduce the chance of winning. *Cloud-mining* experiences no delay but it brings cost on VMs. This delay-cost tradeoff challenges each miner to determine a ratio between self-mining and cloud-mining to maximize his utility. We model interactions among miners as a non-cooperative game and formulate a Nash equilibrium problem to investigate the effects of offloading on miners' utilities. We analyze the existence and uniqueness of equilibrium and propose a distributed algorithm to achieve the equilibrium in a uniform-delay setting. Further, we extend our results to non-uniform delays since miners may choose different network settings, *e.g.* 5G, 4G, or 3G. Both numerical evaluation and testbed experiments on Burstcoin are conducted to show the feasibility of storage offloading and to validate the proposed models and theoretical results.

CCS CONCEPTS

• **Networks** → *Mobile networks*; **Network economics**; • **Mathematics of computing** → Distribution functions.

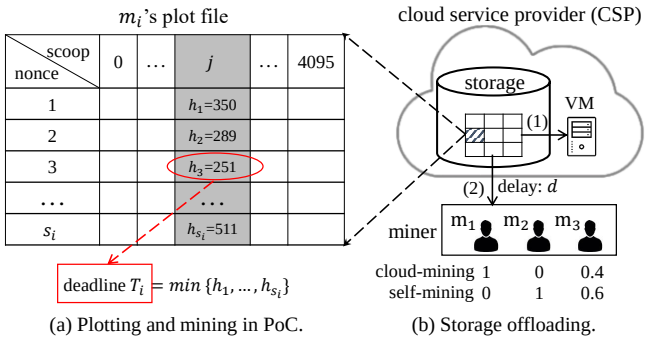
KEYWORDS

cloud storage, game theory, mobile blockchain mining, offloading, Proof of Capacity (PoC)

ACM Reference Format:
Suhan Jiang and Jie Wu. 2020. A Game-theoretic Approach to Storage Offloading in PoC-based Mobile Blockchain Mining. In *The Twenty-first ACM International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (Mobihoc '20), October 11–14, 2020, Boston, MA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397166.3409136

Mobihoc '20, October 11–14, 2020, Boston, MA, USA

ACM ISBN 978-1-4503-8015-7/20/10…$15.00
https://doi.org/10.1145/3397166.3409136



(a) Plotting and mining in PoC. (b) Storage offloading.

Figure 1: Miners offload plot files to a cloud storage and mine blocks (1) via cloud-mining using VMs and/or (2) via self-mining using mobile devices.

1 INTRODUCTION

There has been widespread adoption of blockchain [21] in various fields ranging from cryptocurrency, finance, IoT to public and social services. As a distributed ledger, a blockchain records data in the form of linked blocks secured by cryptography. Consensus mechanisms are a crucial element for every blockchain network as they are responsible for maintaining the security and reliability of these distributed systems. Currently, most blockchain applications are on top of a Proof of Work (PoW) [11] mechanism or a Proof of Stake (PoS) [24] mechanism to determine the block winning probability of each miner. The former relies on miners' computation power to solve cryptographical problems to win a block and hence monetary rewards. PoW is known for its simplicity and attack resistance, while it consumes significant electricity consumption. The latter virtualizes the power-hungry PoW mining by attributing mining power to the proportion of coins held by a miner. PoS can be a viable solution for energy inefficiencies but gives rise to a trend of system centralization since rewards go more towards miners that already have more properties.

Recently, a new mechanism called *Proof of Capacity* (PoC) [1] has emerged as a promising solution to the previously-mentioned problems of energy efficiency and fairness. As its name implies, PoC mining relies on idle storage capacity, *i.e.*, miners invest disk space, as opposed to computation in PoW mining, and the amount of space dedicated to mining determines the chances of winning a block. The mining process consists of two steps: one-time plotting and repeated mining in rounds. In the plotting step, miners configure their available storage with mining-related plot files. Fig. 1 (a) shows miner m_i's plot file, which is arranged into s_i units. Each unit is called a *nonce* and each nonce is divided into 4096 *scoop*s. In each mining round, m_i executes hashing over a scoop column (which

is randomly selected by the system to avoid miner-side cheating) individually to get the smallest value among all cells of the scoop column, which is called m_i's deadline. A miner cannot publish a block until his deadline comes. The one that finds the smallest deadline among all miners wins. Since more storage space gives a higher chance of finding the smallest deadline, a miner's probability of winning a block is related to the ratio between his own space to the total space invested by all miners.

The PoC mechanism has been applied to several blockchain applications, *e.g.* Burstcoin [1], Chia [4], *etc.* However, its storage requirement poses a challenge on mobile devices, thus hindering its applications to mobile services. To facilitate PoC-based blockchain application in future mobile IoT systems, storage offloading appears to be a viable solution. Miners with mobile devices can overcome capacity limitations by offloading all plot files to an external cloud storage. Given that a small amount of computation is required in each PoC mining round, a miner can execute hashing in his device by downloading corresponding scoops (*self-mining*) and/or in the cloud by employing virtual machines (VMs) (*cloud-mining*) provided by the cloud service provider (CSP). Fig. 1 (b) gives an example where miners apply different mining strategies. Self-mining requires no extra cost, but miners cannot start mining until a selected scoop column (one in 4096 rather than all of his plot file) is downloaded. The incurred download delay will reduce a miner's winning probability. Cloud-mining can avoid such a disadvantage, however, it also adds miners' cost on VM employment. Thus, a miner has to determine a suitable mining strategy, *i.e.*, the optimal ratio between self-mining and cloud-mining to maximize his utility, based on his storage offloading decision. In Fig. 1 (b), user m_3 deploys 40% for cloud-mining and the rest for self-mining.

In this paper, we study the interactions among multiple mobile miners, who aim to maximize their own utilities. The individual *utility* is defined as the difference between a miner's expected payoff and his cost, where the expected payoff is a product of the mining reward and his winning probability. Each miner maximizes his utility by deciding his strategies of both storage offloading and mining, while the individual utility is influenced by all miners' decisions. To solve the miners' resource management problem, we exploit game theory and propose a non-cooperative complete-information game to capture this complicated interplay among miners. The corresponding Nash equilibrium problem is then formulated. We prove the existence of unique equilibrium through our theoretical analysis and then propose a distributed algorithm to achieve the equilibrium. In practice, miners can use different network settings, *e.g.* 3G, 4G, or even 5G in the future, which will incur different download delays. Thus, we extend our approach to model non-uniform delays on the miners' strategies. Our primary contributions are as follows:

- As the first work analyzing the PoC mining mechanism in mobile applications, we derive expressions for miner winning probability with delays.
- We define a non-cooperative game to capture interactions among mobile miners and formulate a Nash equilibrium problem to optimize resource allocation among miners.
- We prove and design an algorithm to obtain a unique Nash equilibrium (NE) of the proposed game in the uniform-delay setting, and a close-form strategy is presented for homogeneous miners with identical budgets.
- We show the existence of NE in the non-uniform-delay setting, where the uniform-delay algorithm still can be applied to achieve an NE point.
- We perform numerical evaluation and conduct testbed experiments on Burstcoin and Google Cloud. The equilibria obtained are consistent with all the theoretical results.

2 CHALLENGES AND MAIN RESULTS

This paper aims to solve a resource allocation problem for all miners by maximizing their individual utilities. We assume that miner m_i's files are all offloaded in the cloud and define his utility U_i as the difference between his expected payoff and his cost, denoted C_i, where the expected payoff is a product of the mining reward R and his winning probability, denoted P_i, *i.e.*, $U_i = R \cdot P_i - C_i$. Our goal is to find a strategy that leads to the maximal utility for m_i. However, it is non-trivial to obtain optimal strategies for individual miners, given that each miner's strategy is multi-dimensional, *i.e.*, deciding on how many storage units to purchase and how to arrange the ratio between cloud-mining and self-mining. All miners' strategies can mutually affect their utilities.

To maximize his utility, m_i can either increase his expected payoff by improving his winning probability P_i or decrease his cost C_i. P_i is a complex function determined by multiple parameters. First, P_i should be positively correlated with m_i's storage size since a bigger storage size yields a higher chance of smaller hash values. Second, P_i is also affected by other miners' strategies since m_i cannot win unless his deadline is the smallest among all miners. Last, if m_i chooses self-mining, the incurred download delay d will discount P_i (detailed explanations are given later). In order to improve P_i, m_i is encouraged to buy more storage resources and increase his cloud-mining ratio, which will increase his cost C_i. Thus, improving P_i and decreasing C_i are two conflicting goals. m_i has to buy appropriate storage units and find a reasonable mining ratio to balance P_i and C_i. Based on these parameters, we derive the expression of P_i and verify its validity (Theorem 1).

Given the mutual effects on P_i and hence on U_i, we propose a non-cooperative game to characterize miners' complex interactions. Thus, we turn the original resource allocation problem into a Nash equilibrium (NE) problem, in which each miner's NE strategy is his optimal strategy if NE exists. In the uniform-delay network, we prove the uniqueness of NE (Theorem 2) in the miner game. We further extend it to a non-uniform-delay network, where the existence of NE(s) can be proven (Theorem 3). We also provide a distributed algorithm (Algorithm 1) to compute the unique NE point in the uniform-delay network and provide one NE point in the non-uniform-delay network. We also present a miner's optimal strategy in an explicit expression, given all miners are homogeneous on their budgets in the uniform-delay network (Theorem 4).

3 SYSTEM MODEL AND PROBLEM FORMULATION

This paper focuses on a mobile PoC mining network. Corresponding notations are listed in Table 1. We consider a remote CSP and a set of n miners using mobile devices. Fig. 1 depicts an overview of this

Table 1: Summary of Notations.

Symbol	Description
p_s / p_c	price of cloud storage / computation
v	cloud computation speed
d	download delay from the cloud to miners
D	mining difficulty parameter controlled by the system
R	blockchain mining reward
n	number of miners
m_i	the i-th miner
b_i	m_i's budget
U_i / P_i / C_i	m_i's mining utility / winning probability / cost
x_i / y_i	m_i's cloud / self-mining units
s_i	storage units purchased by m_i, *i.e.*, $s_i = x_i + y_i$
X / Y / S	total cloud-mining / self-mining / purchased units
X_{-i}	total cloud-mining units except m_i's, *i.e.*, $X_{-i} = X - x_i$
Y_{-i}	total self-mining units except m_i's , *i.e.*, $Y_{-i} = Y - y_i$
r_i	m_i's request vector, in the form of (x_i, y_i)
$\boldsymbol{r_{-i}}$ / $\boldsymbol{r}$	all miners except m_i's / all miners' request profile
T_i / T	m_i's / the whole network's deadline

Each storage unit is tailored as a nonce size (256KB in Burstcoin).

network. The CSP can provide resources of storage and computation at a unit price set of (p_s, p_c). Miners participate in mining processes by requesting storage and/or computation resources from the CSP. We differentiate each miner m_i in terms of his budget b_i, which gives an upper bound on the amount of resources he can afford. Thus, different types of miners have different requests. m_i's goal is to find a strategy that lead to the highest U_i.

To maximize his utility, m_i should decide on how many storage units, denoted s_i, to buy from the CSP, and the ratio between cloud-mining and self-mining. We do not directly define the ratio as a variable, instead, we denote x_i as the number of m_i's cloud-mining units and y_i as the number of his self-mining units, respectively. Thus, the could-self mining ratio is captured as x_i/y_i. Then, m_i's request is in the form of $r_i = (x_i, y_i)$. Let $\boldsymbol{r} = \{r_1, \cdots, r_n\}$ and $\boldsymbol{r}_{-i}$ represent the request profile of all miners and all other miners except m_i, respectively. For those storage units mined in the cloud, m_i has to to pay for the storage cost as well as the computation cost, *i.e.*, $(p_s + p_c)x_i$ in total, while for the remaining units mined in his own device, only storage cost of $p_s y_i$ is needed. Thus, m_i's cost is a combination of both, *i.e.*, $C_i = (p_s + p_c)x_i + p_s y_i$.

As miners all want to make as much profit as possible, a competition among miners forms, in which each miner optimizes his utility by deciding his request r_i under the current resource prices (p_s, p_c), while considering his own budget b_i. Thus, m_i's optimization problem can be defined as follows.

Problem 1 (OP_{MINER}).

$$\begin{aligned} &\textit{maximize} && U_i = R \cdot P_i - C_i, && \text{(1a)}\\ &\textit{subject to} && C_i \le b_i, \quad x_i \ge 0, \quad y_i \ge 0, && \text{(1b)}\\ &\textit{where} && C_i = (p_s + p_c)x_i + p_s y_i. && \text{(1c)} \end{aligned}$$

Since m_i's winning probability P_i is a function of multiple parameters, including m_i's request, *i.e.*, $r_i = (x_i, y_i)$ as well as all other miners' requests, *i.e.*, $\boldsymbol{r}_{-i}$, an accurate definition and detailed explanations of P_i will be given in the following section. Each miner m_i aims to maximize his utility and constraint (1b) ensures that m_i is within its budget b_i.

4 MINER'S WINNING PROBABILITY

In this section, we start with a model for traditional PoC mining (subsection 4.1), where miners contributes their disk resources and mine in their own devices like desktop computers or laptops. This basic model allows us to quantify the relation between the block finding time and miners' storage size (subsection 4.2). We then analyze how the winning probability is influenced by the individual storage size and total storage size (subsection 4.3). We extend the basic model to include mobile mining and formulate how the download delay affects the winning probability (subsection 4.4). Finally, we derive the expression for miner's winning probability with download delays after combining all the related parameters (subsection 4.5).

4.1 Overview of PoC Mining

Generally, PoC mining consists of plotting and mining. The plotting process pre-generates and stores mining-related files on miners' storage. Fig. 1 (a) shows m_i's plot file, which is arranged into s_i fixed-length units (a row in Fig. 1 (a)). Each unit is called a nonce and is evenly divided into 4096 scoops (a cell in Fig. 1 (a)). In each mining round, m_i retrieves the jth scoop from each of his units (the grey column in Fig. 1 (a)), where j is selected by the system. m_i executes hashing over every retrieved scoop and gets a hash value as a storage proof. All hash values are within the range of $[0, D]$ and the smallest one is measured as m_i's best proof, named as a deadline (the circle in Fig. 1 (a)), which represents the waiting time before m_i is allowed to publish his block. Thus, the smallest deadline in the network will be measured as the network-wide best proof and its owner will win the block. In fact, D is a parameter controlling the mining difficulty for the miners. (Note, described above is a basic model, where a miner stores his plot file and self-mines using the same device, not involving storage offloading and cloud mining.)

4.2 Block Finding Time and Individual Storage Size

To find the winning probability of each player, we start by analyzing the block finding time probability distribution. We model the block finding time, *i.e.*, the network-wide deadline, as a random variable denoted T. This is a function related to all miners' selection of storage units.

4.2.1 Single miner's distribution functions. In a mining round, a miner can get a proof (*i.e.*, a hash value), denoted h, for each of his contributed storage units. In fact, h is a random variable, of which the value is subject to a uniform distribution within the range of $[0, D]$ in each round. Given that miner m_i commits s_i units of his storage in total, we model his deadline as a random variable denoted T_i. Obviously, $T_i = \min\left\{h_1, \cdots, h_{s_i}\right\}$. T_i's cumulative distribution function (CDF), denoted $F(t, s_i, D)$, can be obtained in the below.

$$F(t, s_i, D) = \begin{cases} 0 & t \le 0, \\ 1 - (1 - t/D)^{s_i} & 0 < t < D, \\ 1 & t \ge D. \end{cases} \quad (2)$$

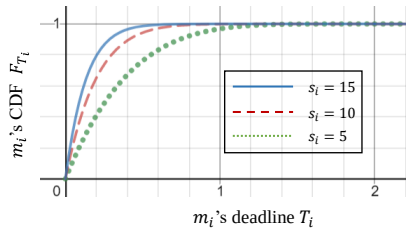


Figure 2: CDFs under different storage sizes given $D = 2$ min.

Thereupon, the corresponding probability density function (PDF), denoted $f(t, s_i, D)$, follows through performing derivative over $F(t, s_i, D)$, as is shown in Eq. (3).

$$f(t, s_i, D) = \begin{cases} \frac{s_i}{D}\left(1 - t/D\right)^{s_i - 1} & 0 < t < D, \\ 0 & otherwise. \end{cases} \tag{3}$$

4.2.2 Whole mining network's distribution functions. Given a mining network with n miners, the block finding time can be expressed as $T = \min\{T_1, \cdots, T_n\}$. We have already calculated the PDF and the CDF over T_i for $\forall i \in [1, n]$, thus the corresponding distribution functions $F_T(t, S, D)$ and $f_T(t, S, D)$ can be expressed using Eqs. (2-3), by replacing s_i with $S = \sum_j s_j$.

4.3 Influences of Total Storage Size

Now, we analyze how storage sizes affect P_i. Intuitively, a miner's winning probability should be positively related to his own storage size, since more storage units lead to a higher chance of smaller deadlines. Fig. (2) further confirms our guess, where T_i's CDF (as Eq. (2) describes) hits 1 faster under a larger s_i. Meanwhile, P_i should also be affected by the total mining storage space. To capture the exact mathematical relation between the individual winning probability and storage sizes (both individual and total), we start with a competition between only two miners m_i and m_j, owning the storage sizes of s_i and s_j, respectively. Obviously, m_i wins when he finds a smaller deadline. The probability that m_i wins is calculated as follows:

$$\begin{aligned} P_i = Pr[T_i < T_j] &= \iint f\left(t_i, s_i, D\right) f\left(t_j; s_j, D\right) \mathrm{d}t_i \mathrm{d}t_j \\ &= 1 - s_j/(s_i + s_j) = s_i/(s_i + s_j). \end{aligned}$$

Thus, the probability that m_i wins is proportional to his fraction of the total storage size. That is, given a total storage of size S, m_i's individual winning probability is $P_i = s_i/S$. Obviously, dedicating more storage space yields a proportionally higher expectation of successfully mining a block. Therefore, a PoC-based incentive mechanism can reward smaller miners fairly according to their contribution to the network, thus incurring more distributed participation.

4.4 Influences of Delay

4.4.1 Download delay in self-mining. The expression s_i/S characterizes the probability that m_i happens to hold the smallest deadline among all miners. However, it is possible that the owner of the smallest deadline is not the block winner. Suppose, in a certain mining round, a miner m_i's deadline is 100 seconds, the smallest one among all miners' deadlines, and another miner m_j's deadline is 105 seconds, only next to m_i's deadline. If m_i finds his deadline within 100 seconds, then he propagates his block until that time comes and becomes the winner. However, if m_i's mining is delayed for some reason and hence he fails to finds his deadline within 105 seconds, at which time m_j succeeds in broadcasting his block, then m_j becomes the winner, although he is not the owner of the smallest deadline. Thus, with delay, a miner's winning probability is definitely discounted.

In the traditional PoC mining, since plot file storing and mining happen in the same device, usually a desktop computer or a laptop, all deadlines can be calculated before the smallest one comes. Thus, it is just a race on miners' contributed storage. However, when applying storage offloading, a delay, denoted d, can be incurred in self-mining due to the scoop download from the cloud to a miner's device. Miners cannot start self-mining until the required scoops are downloaded. During the waiting time, if there is a deadline no greater than d calculated using VMs in the cloud, then the corresponding block can be successfully published and rewarded, although there may exist smaller deadlines not yet computed by self-mining. In reality, block propagation delay also damages a miner's winning probability. To focus on the influence of the download delay, we assume propagation delay among miners is negligible.

4.4.2 Download delay and winning probability. We now extend the basic model with the download delay d. We show how d discounts miner's winning probability. During the download delay d, if the speed of cloud mining is v, then there should be roughly vdn proofs computed in total. If the best one among them is less than d, then the corresponding block definitely wins whether or not it is a network-wide optimal deadline. The probability, denoted β, that a cloud-mined block wins before the self-mining starts can be expressed as $\beta\left(d, v\right) = 1 - \left(1 - d/D\right)^{vdn}$. We simplify our model by assuming cloud-mining can perform deadline calculations fast, *i.e.*, all deadlines over total cloud-mining units Y, *i.e.*, $\sum_{i=1}^{n} y_i$, can be calculated within d, then the corresponding probability β can be refined as below.

$$\beta\left(d, Y\right)|_{v \to +\infty} = 1 - \left(1 - \frac{d}{D}\right)^{Y}. \tag{4}$$

4.5 Expression of Winning Probability

We are now ready to express P_i in the model of storage offloading, *i.e.*, a miner stores his plot file in the cloud instead of his own device. P_i consists of two parts, P_i^c and P_i^s, jointly contributed by cloud-mining and self-mining, where P_i^c and P_i^s are functions of r_i and $\boldsymbol{r}_{-i}$ given below:

$$\begin{aligned} P_i^c(r_i, \boldsymbol{r}_{-i}) &= \frac{x_i}{S} + \frac{x_i}{X}\frac{Y}{S}\beta, \\ P_i^s(r_i, \boldsymbol{r}_{-i}) &= \frac{y_i}{S} - \frac{y_i}{Y}\frac{Y}{S}\beta = y_i\frac{1-\beta}{S}, \end{aligned} \tag{5}$$

where $X = \sum_{i=1}^{n} x_i$, $Y = \sum_{i=1}^{n} y_i$, and $\beta = \beta(d, X)$ is for simplicity. Next, we verify the validity of P_i as a probability mass function.

Theorem 1. *$P_i = P_i^c + P_i^s$ is a valid probability mass function to express the winning probability of individual miners in a mobile blockchain mining network.*

Proof. We present the full verification process by checking that $\sum_{i=1}^{n} P_i = 1$ holds, i.e.,

$$\sum\nolimits_{i=1}^{n} P_i = \sum\nolimits_{i=1}^{n} (P_i^c + P_i^s)$$

Algorithm 1 Best Response Algorithm

Output: $\boldsymbol{r} = \{r_1, \cdots, r_n\}$ where $r_i = (x_i, y_i), \quad i \in \{1, n\}$
Input: Initialize k as 1 and pick a feasible starting point $\boldsymbol{r}^{(0)}$

```
for round k do
    for miner i do
        Decide r_i^(k) = r_i^(k-1) + Δ ∂U_i(r_i, r_{-i}^(k-1)) / ∂r_i
        Send the request r_i^(k) to CSP
    CSP collects the request profile r^(k)
    if r^(k) = r^(k-1) then Stop
    else set k ← k + 1
```

$$= \sum_{i=1}^{n} (x_i + y_i)/S + Y\beta \cdot (x_i/X - y_i/Y)/S.$$
$$= 1 + \frac{Y\beta_d}{S} \cdot (\frac{X}{X} - \frac{Y}{Y}) = 1. \qquad \square$$

We can conclude that, the winning probability we use is valid, hence our model is as well. Note that m_i's winning probability and hence its utility depends not only on its request but also on those of the other miners.

5 MINER NON-COOPERATIVE GAME

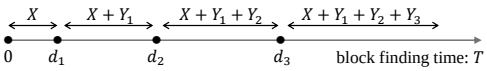
In this section, we analyze the existence and uniqueness of NE in the proposed miner game under different network setting assumptions (subsection 5.1). Since the NE point is hard to express in a closed form, we provide an algorithm where the NE point can be computed in a distributed way, as detailed in subsection 5.2. In subsection 5.3, we study a special case where all miners are homogeneous on the budget, where we can derive explicit expressions of each miner's optimal strategies in the uniform-delay network.

5.1 Nash Equilibrium

5.1.1 Mining in a uniform-delay network. In this part, we assume the download delay d experienced by each self-mining miner is uniform. We conduct analysis on Nash equilibrium (NE) in the uniform-delay network, and the result can be concluded in Theorem 2. The detailed proof is provided in Appendix A.

Theorem 2. *A unique Nash equilibrium exists in OP_{MINER} in the uniform-delay network.*

5.1.2 Mining in a non-uniform-delay network. In reality, download delay cannot be uniform among all miners since their network settings are different. To make our analysis more realistic while not over-complicating our model, we use network types to characterize network settings. Miners under different network types will experience different delays. We allow k types of network settings, and miners using type-j network experience a download delay of d_j. We assume X storage units will be mined in the cloud, and Y_j storage units will be mined by all type-j miners using their devices. Thus, the mining timeline is segmented into $k+1$ periods based on all delay values. During period 1 (*i.e.*, between time 0 and d_1), only X storage units are contributed for mining. At time d_i, Y_i storage units will join in the mining if there is still no block found at that time. Thus, during period $i+1$ (*i.e.*, between time d_i and d_{i+1}), $S_{i+1} = X + \sum_{l=1}^{i} Y_i$ storage units are contributed for mining. We are ready to conclude every possible block finding time T and its corresponding probability in the below. (p_i represents the probability that a block is found in period i, *i.e.*, between time d_{i-1} and d_i.)



T	Probability ($\alpha_j = 1 - \frac{d_j}{D}$)
$T \le d_1$	$p_1 = 1 - \alpha_1^X$
$d_1 < T \le d_2$	$p_2 = \alpha_1^X (1 - \alpha_2^{X+Y_1})$
$d_2 < T \le d_3$	$p_3 = \alpha_1^X \alpha_2^{X+Y_1} (1 - \alpha_3^{X+Y_1+Y_2})$
$T > d_3$	$p_4 = \alpha_1^X \alpha_2^{X+Y_1} \alpha_3^{X+Y_1+Y_2}$

Figure 3: Probability of the network-wide block finding time where $k = 3$.

$$p_i = \begin{cases} 1 - \alpha_1^X & i = 1, \\ (1 - \alpha_i^{S_i}) \prod_{j=1}^{i-1} \alpha_j^{S_j} & i = 2, \cdots, k, \\ \prod_{j=1}^{k-1} \alpha_j^{S_j} & i = k+1, \end{cases} \tag{6}$$

where $\alpha_j = 1 - d_j/D$.

Fig. 3 gives an example of $k = 3$. Now, we consider how to express the winning probability P_i^j for a miner m_i using type-j network. In each mining round, P_i can be considered as a sum of m_i's winning probability in each period l, which is a product of p_l and m_i's storage contribution ratio during the period l, *i.e.*, the ratio of his contributed storage to the total storage in the period l. Obviously, m_i contributes x_i storage units before time d_j and s_i storage units after that. Thus, we can calculate the winning probability for a type-j miner m_i and his corresponding utility using Eqs. (7-8).

$$P_i^j = \sum_{l=1}^{j} \frac{x_i p_l}{S_l} + \sum_{l=j+1}^{k+1} \frac{s_i p_l}{S_l}, \tag{7}$$

$$U_i^j = RP_i^j - C_i. \tag{8}$$

If we apply this function into the original miner game, then the following result can be obtained.

Theorem 3. *Given a price set $\{p_s, p_c\}$ from the CSP side, there exists at least one Nash equilibrium for the miner game in the non-uniform-delay setting.*

Proof. The uniform-delay setting is a special case where $k = 1$. Similar to the proof for NE in Theorem 2, the existence of NE for miners in a non-uniform-delay network is followed by capitalizing on the variational inequality theory. Based on the previous analysis, we need to prove P_i^j is concave in r_i.

We rewrite P_i^j to Eq. (9).

$$P_i^j = \underbrace{\sum_{l=1}^{j-1} \frac{x_i p_l}{S_l}}_{\text{part a}} + \underbrace{\frac{x_i p_j}{S_j} + \sum_{l=j+1}^{k+1} \frac{s_i p_l}{S_l}}_{\text{part b}}. \tag{9}$$

Based on the the proof in Theorem 2, we can obtain the fact that part b of Eq. (9) is a concave function. We only need to prove that part a is concave, each addend of which is concave. Since the sum of concave functions are still concave, the concavity of part a as well as Eq. (9) are determined. $\square$

Note that, we never consider the mathematical relation among Y_j's. Thus, Theorem 3 can be obtained under any number of network types and any combinations of M miners' network types. In

Algorithm 2 Best Response Algorithm

Output: $\boldsymbol{r} = \{r_1, \cdots, r_n\}$ where $r_i = (x_i, y_i), \quad i \in \{1, n\}$
Input: Initialize j as 1 and choose any feasible starting point $\boldsymbol{r}^{(0)}$: each miner chooses the decision using the local computing
1: **for** round j **do**
2: **for** miner i **do**
3: Update $r_i^{(j)}$ using Eq. (12) for x_i and Eq. (13) for y_i
4: Send the request $r_i^{(j)}$ to CSP
5: CSP collects the request profile $\boldsymbol{r}^{(j)}$
6: **if** $\boldsymbol{r}^{(j)} = \boldsymbol{r}^{(j-1)}$ **then** Stop
7: **else** increase j by 1

the experiment, we conduct experiments based on several special network type distributions to see how each distribution affects miners' decisions.

5.2 Nash Equilibrium Algorithm

Since it is hard to express each miner's equilibrium strategy in an explicit form, although it exists, we provide an algorithm using strategy iterations to find miners' equilibrium strategies.

Each miner optimizes his utility by solving the OP_{MINER} problem as follows. Using Lagrange's multipliers λ_1, λ_2, and λ_3 for the constraints, we obtain the Lagrange function L_i to reform m_i's optimization problem as follows:

$$L_i = RP_i - C_i - \lambda_1(C_i - b_i) + \lambda_2 x_i + \lambda_3 y_i, \tag{10}$$

and the complementary slackness conditions are

$$\begin{aligned} &\lambda_1(C_i - b_i) = 0, \quad \lambda_2 x_i = 0, \quad \lambda_3 y_i = 0, \\ &\lambda_1 > 0, \quad \lambda_2, \lambda_3, x_i, y_i \geq 0. \end{aligned} \tag{11}$$

By the first-order optimality condition $\nabla L_i = 0$, it immediately follows that $\lambda_2 = \lambda_3 = 0$. For simplicity, we approximates the value of β by replacing X with X_{-i}. This approximation has little effect on the final results but makes our expression clear and simple. Thus, we obtain the following result.

$$x_i = \sqrt{\frac{\beta R X_{-i}}{(1+\lambda_1)p_c}} - X_{-i}, \tag{12}$$

$$y_i = \sqrt{\frac{(1-\beta)R(X_{-i}+Y_{-i})}{(1+\lambda_1)p_s}} - (x_i + X_{-i} + Y_{-i}), \tag{13}$$

$$C_i - b_i = 0. \tag{14}$$

Solving Eq. (12) - Eq. (14) yields that

$$\begin{aligned} \lambda_1 = {} & [b + (p_s + p_c)X_{-i} + p_s Y_{-i}]^2 \\ & - \left[\sqrt{\beta R p_c X_{-i}} + \sqrt{(1-\beta)R p_s (X_{-i} + Y_{-i})}\right]^2. \end{aligned} \tag{15}$$

Hence, substituting Eq. (15) back into Eq. (12) and Eq. (13) gives the explicit form of the solution to the OP_{MINER} problem, *i.e.*, each miner's best response strategy. This naturally gives a distributed iterative algorithm, allowing each miner to iteratively update its strategy, given the strategies of other miners. We summarize the distributed iterative algorithm in Algorithm 1.

Algorithm 1 is applicable for a uniform-delay network to find the unique NE point and also can be used in a non-uniform-delay network, where one NE point can be computed. Table 2 gives an example, showing how one NE point is achieved through strategy iterations. In this example, two miners with different network delays and budgets compete with each other. Provided with different initial values, they finally stabilize to the same NE point (*i.e.*, the bold numbers in the table).

5.3 Nash Equilibrium among Homogeneous Miners

It is difficult to express miners' strategies explicitly. Fortunately, we are able to get the closed-form computation offloading solutions for the OP_{MINER} in a homogeneous-miner case, where each miner is homogeneous with an identical budget b. We will provide the explicit-form expression of the offloading strategy for the homogeneous-miner case in the uniform-delay network.

Theorem 4. *The unique Nash equilibrium for miner m_i in the homogeneous-miner case is given below, provided that the network delay is uniform among miners.*

$$x_i^* = \frac{b\beta(n-1)}{p_c(n-\beta)}, \tag{16}$$

$$y_i^* = \frac{b[(1-\beta)np_c - \beta(n-1)p_s]}{p_s p_c (n-\beta)}. \tag{17}$$

Proof. We obtain $X^2 = \beta R X_{-i}/[(1+\lambda_1)p_c]$ based on Eq. (12) and $S^2 = (1-\beta)R(X_{-i}+Y_{-i})/[(1+\lambda_1)p_s]$ based on Eq. (13), for each miner m_i. Then, $X^2/S^2 = \beta X_{-i} p_s/[(1-\beta)(X_{-i}+Y_{-i})p_c]$ immediately comes out. Next, we calculate the summation of this expression for all the miners as follows:

$$\sum_n \frac{X^2}{S^2} = \sum_n \frac{\beta X_{-i} p_s}{(1-\beta)(X_{-i}+Y_{-i})p_c}, \tag{18}$$

$$n\left(\frac{X}{S}\right)^2 = \frac{\beta p_s}{(1-\beta)p_c}(n-1)\frac{X}{S}. \tag{19}$$

Then, $X/S = \beta(n-1)p_s/(1-\beta)np_c$ easily follows. Since all miners are homogeneous, their best response strategies are identical as well, *i.e.*, $X = nx_i$ and $S = n(x_i + y_i)$. By substituting these two equations into Eq. (14), we obtain the NE for miner m_i. □

However, the closed-form solution for homogeneous miners in the non-uniform-delay network is still an open problem, which we reserve for our future work.

6 EVALUATION

This section consists of three parts. First, we validate the feasibility for mobile miners to offload their storage to an external CSP (subsection 6.1). The testbeds we use in this paper are Burstcoin [1], a PoC mining platform, and Google Cloud [9], providing resources of both storage and computation. Second, we examine how miners decide their optimal strategies using our proposed algorithm in the uniform-delay network (subsection 6.2). We conduct our experiments based on different sets of parameters to show how miners' decisions will be affected by external factors. Last, we take the network settings into consideration and analyze how non-uniform delays can influence the achieved equilibrium in our proposed game (subsection 6.3).

Table 2: Strategy iterations given $R = 800, p_s = 5, p_c = 25, d_1/d_2 = 5/6, b_1 = 200, b_2 = 500$.

			Round													
Run	Strategy	Initialization	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	x_1	7	1.2	5.1	4.2	4. 5	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	**4.4**
	y_1	11	15.4	4.3	7.2	6.3	6.61	6.5	6.5	6.5	6.5	6.5	6.52	6.5	6.5	**6.5**
	x_2	5	8.6	7.3	7.7	7.6	7.6	7.6	7.6	7.6	7.6	7.6	7.6	7.6	7.6	**7.6**
	y_2	5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.0**
2	x_1	3	0	5.3	4.1	4.5	4.8	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	**4.4**
	y_1	4	18.4	3.7	7.4	6.3	6.6	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	**6.5**
	x_2	1	8.9	7.2	7.7	7.6	7.6	7.61	7.6	7.6	7.6	7.6	7.6	7.6	7.6	**7.6**
	y_2	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.0**

6.1 Feasibility of Storage Offloading

The most important part is to validate whether storage offloading is viable for PoC mining since it is the basis of our paper. To confirm its feasibility, we show successful Burstcoin mining using Google Cloud storage.

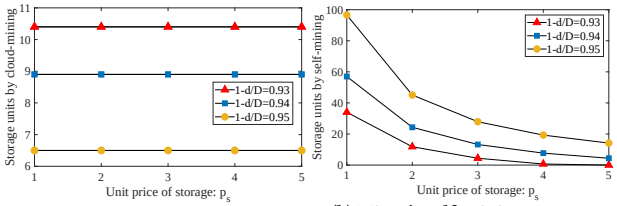
6.1.1 Feasibility of plotting. For plotting, there are two methods. One is to plot directly onto the local hard drive and upload it to cloud service, and the other way is to plot straight from the local device to the cloud service by setting the cloud as a drive letter. As we observed, it takes around 24 hours to upload 1 TB of plot files under the bandwidth of 500 Mbps if choosing the first way. Definitely, this time can be shortened with a faster Internet speed. When using the second way, the plotter just shuts down if the bandwidth is less than 1 Gbps. There is a bandwidth threshold if applying the second way, and thus we suggest the first way to offload plot files.

6.1.2 Feasibility of mining. If we choose cloud-mining using VMs provided by the CSP, the mining speed would be faster. According to our test, with a 2 Gbps network instance and 32 vCPU, mining over a plot file of 18 TB can be finished within 30 seconds, which is far less than the average block generation interval (240 seconds). For self-mining, we also test the download delay. For a plot file of 18 TB in total, we need to download 4068MB scoops in each mining round and the average delay is around 50 seconds. The gap between the cloud-mining end time and the self-mining start time rationalizes our assumption on an infinite cloud-mining speed. We also test the mining speed on mobile devices. Self-mining is slower than cloud-mining, but its speed is acceptable since we can finish the mining within 30 to 60 seconds, depending on the type of mobile devices. In general, the total time for local mining is also within the average block generation interval.

Based on the data provided in the above, we can conclude that it is feasible for miners to use mobile devices for PoC mining through storage offloading. On this basis, we conduct further experiments to confirm our theoretical analysis.

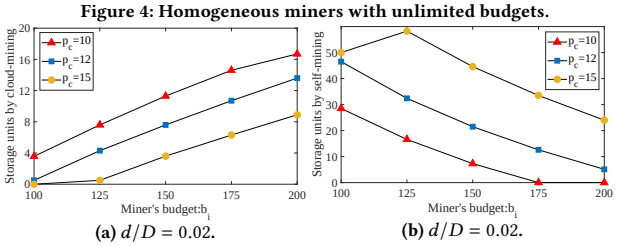
6.2 Unique Equilibrium in the Uniform-delay Setting

Our experiments evaluate the influences of important parameters on each miner's strategies. We start with a small mobile PoC mining network with 5 homogeneous miners with unlimited budgets.



(a) Miner's cloud-mining storage. **(b) Miner's self-mining storage.**

Figure 4: Homogeneous miners with unlimited budgets.

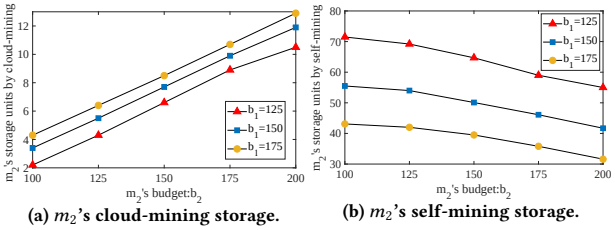


(a) $d/D = 0.02$. **(b)** $d/D = 0.02$.

Figure 5: Budget impacts on homogeneous miners.

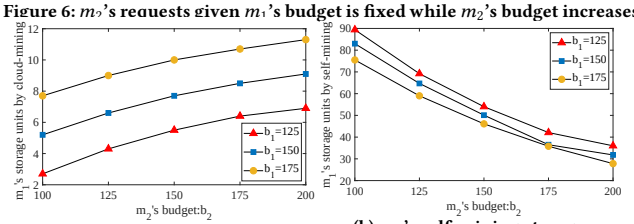
6.2.1 Influences from the CSP side. We first consider the different prices at the CSP side. Fig. 4 obviously reflects that, if the CSP unilaterally increases storage's price p_s, miners will decrease their requests on storage units as well as their self-mining ratio. However, miners never change their investment on cloud-mining, although it is expensive. This is reasonable, given that miners are budget-unlimited. Similarly, the increase of computation price p_c also discourages miners to invest on computation resources. Besides, from Fig. 4, we can also conclude that miner's utility is sensitive to the download delay, as a slight increase of d would cause an obvious decrease on the self-mining ratio. Surprisingly, this negative effect also influences the sales on total storage. This can be interpreted as follows: a longer delay decreases the mining power of the whole network, while in our experiment, we fix the mining difficulty parameter D, leading to a lower chance for miners to get reward. Thus, miners tend to reduce their cost investment for the purpose of maximizing utility. This result confirms the necessity for a blockchain-based system to dynamically adjust its difficulty.

6.2.2 Influences from the miner side. We now investigate how the budget will affect miners' strategies. We assume miners are homogeneous on their budgets b_i. In Fig. 5, we can observe that, when each miner's budget increases, he will prefer to invest more money in cloud-mining even if p_c goes up. Under the setting of $p_c = 15$, when we increase the miner's budget from 100 to 125, he just slightly increases his investment on cloud mining (from 0 to 0.5), which



(a) m_2's cloud-mining storage. (b) m_2's self-mining storage.

Figure 6: m_2's requests given m_1's budget is fixed while m_2's budget increases.



(a) m_1's cloud-mining storage. (b) m_1's self-mining storage.

Figure 7: m_1's requests given m_1's budget is fixed while m_2's budget increases.

Table 3: Miners' strategy profiles under different delay ratios.

	Type1		*Type2*		*Type3*	
$\theta_1 : \theta_2 : \theta_3$	x	y	x	y	x	y
3 : 4 : 5	7.3	88.9	11.8	0	16.8	0
4 : 5 : 6	12	31.7	13	0	14.8	0
5 : 6 : 7	12.3	4.4	13.3	0	14.2	0

Table 4: Miners' strategy profiles under different population distributions.

	5G			*4G*			*3G*		
$\gamma_1 : \gamma_2 : \gamma_3$	x	y	U/C	x	y	U/C	x	y	U/C
10 : 40 : 10	54	20.5	0.57	17	0	0.57	57	0	0.57
15 : 40 : 5	35	34	0.76	18.8	0	0.73	57	0	0.73
20 : 20 : 20	30	4	0.5	32	0	0.5	32	0	0.5
30 : 20 : 10	20	5	0.54	33.2	0	0.54	57	0	0.54

is less than his budget increase, allowing him to invest more on self-mining.Next, we analyze the heterogeneous-miner scenario. We assume there are two miners, m_1 and m_2. We fix m_1's budget and constantly increase m_2's budget. From Fig. 6, we can see m_2 invest more on computation resources. By taking advantage of cloud-mining, m_2 can buy less storage units while still keeping his utility high. Since miners mutually affect each other, although m_1's budget remains unchanged, he will also adjust his strategies as a best response to m_2's strategy changes. Fig. 7 shows that m_1 also increases his investment on cloud-mining, as m_2 does.

6.3 Equilibrium in the Non-uniform-delay Setting

Miners' selection of network types will bring different download delays to them. In this part, we evaluate miners' equilibrium strategies in a mining network with non-uniform delays. We construct a mining network of 60 homogeneous miners with identical budgets and three network types for miners to choose from, where a type-i network will incur a delay of $\theta_i d$ ($i = 1, 2, 3$), where θ_i depends on the type of networks used .

6.3.1 Influences of delay ratio. We first investigate the influence of different delay ratios, *i.e.*, the values of $\theta_1 : \theta_2 : \theta_3$, on the

Table 5: Miners' strategy profiles under different price sets.

	5G		*4G*		*3G*	
(p_s, p_c)	x	y	x	y	x	y
(5, 15)	0	40	10	0	10	0
(5, 20)	0	40	6.25	8.75	8	0
(5, 25)	0	40	2.5	24.7	6.7	0
(5, 30)	0	40	0.3	37.8	5.7	0

miners' strategies. We assume each miner's budget is 200 and each network type is used by 20 miners. Given the CSP price set of $(p_s, p_c) = (1, 12)$, Table 3 shows miners' strategy profiles under different delay ratios, where x and y represent the number of storage units for cloud-mining and for self-mining, respectively. In each scenario, only type-1 miners will apply self-mining for cost saving since they can take advantage of their short delay. However, as their download delay increases, they have to reduce the ratio of self-mining and turn to cloud-mining. As type-2 and type-3 miners have long delays, they prefer cloud-mining. Compared with type-2 miners, type-3 miners also buy more storage units in order to mitigate their disadvantages caused by their longer delays.

6.3.2 Influences of population distribution. Then, we investigate the influence caused by population distribution of each network type. We use 5G, 4G, and 3G to represent type-1, type-2, and type-3 network, respectively. Thus, the delay ratio $\theta_1 : \theta_2 : \theta_3 = 3 : 20 : 500$ is obtained based on the real-world data[28]. We assume all miners have unlimited budgets and the CSP price set is $(p_s, p_c) = (5, 30)$, then Table 4 shows the corresponding miner strategies under different population distributions, where γ_i represents the number of miners using the type-i network. We can conclude that, if the percentage of type-i miners increases, miners of this type will decrease their storage investment, while the remaining miners have to buy more units. Also, we observe an interesting result: even if miners use different network types, their rates of utility on cost are almost equal. This further confirms egalitarian nature of PoC consensus mechanism.

6.3.3 Influences of CSP prices. Finally, we investigate the influence incurred by the CSP's pricing strategy. We consider a small mining network with 3 miners using 5G, 4G, and 3G, respectively, *i.e.*, $\theta_1 : \theta_2 : \theta_3 = 3 : 20 : 500$. Assuming each miner has a budget of 200, Table 5 shows their equilibrium strategies under different populations. As the 5G-miner can exploit his advantage of the fast download speed, he prefers low-cost self-mining. However, the 3G-miner chooses cloud-mining to avoid the winning probability decrease caused by his slow network, even if p_c is high. For the intermediate 4G-miner, his strategies are tightly related to the value of p_c, where a smaller p_c is attractive.

7 RELATED WORK

7.1 Blockchain Mining Consensus Mechanism

A blockchain is viewed as a distributed ledger stored and maintained by a network of nodes across the world. The key to operating a blockchain is its consensus mechanism, which regulates how to update this ledger to reach a decentralized agreement. Blockchain consensus mechanisms can be divided into two categories, *i.e.*, Proof

of Concept (PoX) which requires miners to devote resources to mining, and virtual mining without real-world resource contribution.

7.1.1 PoX series. As the origin of PoX series, PoW [21] uses computing power for preimage searching and creates a huge amount of electricity waste. In [25], Proof of Exercise is proposed to replace useless searching in PoW with the useful exercise of matrix product problems. Apart from delegation of expensive computation, PoX can also be designed to incentivize cheap storage provisions. The existing mechanisms include Proof of Capacity adopted by Burstcoin [1] and Spacemint [22], Proof of Retrievability [13] used in Permacoin [20] and Ipfs [2] *etc.*

7.1.2 Virtual-mining mechanisms. Compared to PoX series, there are fewer virtual-mining mechanisms, among which PoS is a representative. In the PoS mechanism, miners decide their mining right based on stake, which is the amount of crypto-currencies one possesses. There is a refined version of PoS [24], which uses coin age, the currency amount times the holding period, to decide reward. One of its implementations is Peercoin [15]. Another extension is delegated PoS [16]. [14] adds more security measurements to ensure persistence and liveness of the system. There also exist some hybrid consensus mechanisms atop PoW protocols and BFT [3] protocols, *e.g.* bitcoin-NG [7], PeerCensus [5] and Hybrid Consensus [23].

7.2 Game Theory in Offloading Mechanism

Game theory is a widely-used model in the field of offloading mechanisms. A large body of existing literature [6, 10, 17, 19, 26, 27, 29–31, 33, 34] focuses on minimizing offloading users' computation overhead in terms of energy and latency. To this end, researchers have developed distributed decision making methodologies. In the field of mobile blockchain mining offloading [12, 18, 32], there are few works and most of them are in the PoW-mining scenario where mobile miners only offload their computation to a service provider. In our paper, we consider mining based on a PoC mechanism, thereby storage offloading, instead of computation offloading, becomes the core. Meanwhile, the computation offloading is still in need if miners decide to mine remotely in the cloud.

8 CONCLUSION

We have proposed a Nash equilibrium game among the mobile miners for optimal storage offloading in the PoC mining setting. Two practical mining strategies are investigated for miners, *i.e.*, cloud-mining and self-mining. We start with a uniform network setting where each miner experiences the same download delay for self-mining. We discuss the existence and the uniqueness of Nash equilibrium in the proposed game and a distributed algorithm is proposed to achieve NE point(s). We also find the close-form expressions of offloading and mining strategies for homogeneous miners with identical budgets. Then, we extend our results to non-uniformed delays, where miners with different network settings experience different download delays. We prove that there exists at least one Nash equilibrium in this setting, and our previously-proposed algorithm still can be applied to achieve one NE point. We also find that a miner using a fast-speed network can decrease his cost on cloud resources while obtaining higher utility. This is reasonable since the price goes higher if subscribing to a better network, which can be considered as another investment source for mobile mining. Both numerical evaluation and testbed experiments on Burstcoin and Google Cloud are conducted to show the feasibility of storage offloading and to validate the proposed models and theoretical results.

9 ACKNOWLEDGEMENTS

This research was supported in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS 1651947, and CNS 1564128.

A PROOF OF THEOREM 2

First we show the existence of NE.

Claim 1: There is at least one NE for the game $\mathrm{OP}_{\mathrm{MINER}}$. Any game has NEs if its equivalent variational inequality (VI) problem [8] has a nonempty solution set. Given a VI problem, $VI(K, G)$, if K is convex and compact, and G is monotone on K, then the solution set of $VI(K, G)$ is nonempty, closed, and convex. We define the equivalent VI problem $VI(K, G) = OP(X, U)$, where

$$G := (\nabla_i U_i)_{i=1}^n, \quad X := ((x_i, y_i))_{i=1}^n, \quad U := (U_i)_{i=1}^n,$$
$$K := \prod_{i=1}^n K_i, \quad K_i := \{(x_i, y_i) | C_i \le b_i, x_i, y_i \ge 0\}.$$

It can be easily verified that K_i is convex and closed, $\forall i$. Thus, K is convex and compact. And G is monotone if and only if $U_i(r_i, \boldsymbol{r}_{-i})$ is concave in r_i for given $\boldsymbol{r}_{-i}$, $\forall i$, which is true as shown below. Since the VI problem has a nonempty solution set, the existence of NE thus follows the sufficient conditions. Denote H for the Hessian matrix of U_i:

$$H := \begin{bmatrix} U^i_{xx} & U^i_{xy} \\ U^i_{yx} & U^i_{yy} \end{bmatrix},$$

where $$U^i_{xx} = \frac{\partial^2 U_i}{\partial x_i^2}, U^i_{xy} = U^i_{yx} = \frac{\partial^2 U_i}{\partial x_i \partial y_i}, U^i_{yy} = \frac{\partial^2 U_i}{\partial y_i^2}.$$

The expressions of the Jacobian elements are as below:

$$\begin{aligned}\frac{\partial U_i}{\partial x_i} =& p_s + p_c - R\Big\{(1 - (x_i + y_i)/S) + \alpha^X \ln\alpha\,(y_i - Y x_i/X) \\ &+ \left(1 - \alpha^X\right)\left[Y\left(1 - x_i/X - x_i/S\right)/X + y_i/S\right]\Big\}/S, \\ \frac{\partial U_i}{\partial y_i} =& p_s - R\alpha^X\left[1 - (x_i + y_i)/S\right]/S.\end{aligned}$$

Next, we show H is positive definite by proving its leading principal minors, *i.e.*, U^i_{xx} and $det(H)$, are bigger than 0.

$$det(H) = U^i_{xx}U^i_{yy} - U^i_{xy}U^i_{yx} = R^2(\psi\phi - \psi'\phi')/(X^3S^4),$$

where

$$\psi = 4X_{-i}S, \psi' = SX_{-i} + y_iX_{-i} - x_iY_{-i},$$
$$\phi = 1 - \alpha^X\left(1 - \ln a^X\right), \phi' = \alpha^X\left(\ln a^X\right)^2.$$

The sign of $det(H)$ is decided by the value of $\psi\phi - \psi'\phi'$, which is always positive for non-negative requested units.

$$\begin{aligned}\psi\phi - \psi'\phi' &= 4X_{-i}S\phi - [(S + y)X_{-i} - x_iY_{-i}]\phi' \\ &> 4X_{-i}S\phi - (2SX_{-i} - x_iY_{-i})\phi' \\ &> 4X_{-i}S\phi - 2X_{-i}S\phi' = 2X_{-i}S(2\phi - \phi').\end{aligned}$$

Since $2\phi - \phi' = 2 - 2\alpha^X\left(1 - \ln\alpha^X\right) - \alpha^X\left(\ln\alpha^X\right)^2$ is a monotone increasing function, it is obvious that $2\phi - \phi' \ge 2\phi - \phi'\,|_{X=0} = 0$,

$\forall Y \geq 0$. Thus, $det(H) > 0$ holds. Obviously, $U^i_{xx}U^i_{yy} > 0$ and $U^i_{yy} > 0$, then $U^i_{xx} > 0$ is logically well-reasoned. As $det(H) > 0$ and $U^i_{yy} > 0, \forall (x_i, y_i) \in K_i$, and the positive definiteness holds for any i. Therefore, $VI(K, G)$ is equivalent to $OP(X, U)$ and has a nonempty solution set. We thus prove that *Claim 1* is legitimate. We finish the proof for the uniqueness of NE.

Claim 2: There is at most one NE for the game OP_{MINER}. We first introduce the matrices J_{low}, defined as

$$[J_{low}]_{ij} := \inf_{x \in K} \begin{cases} |\nabla^2_{ii} U_i|, & \text{if } i = j, \\ -\frac{1}{2}(|\nabla^2_{ij} U_i| + |\nabla^2_{ji} U_j|), & \text{else.} \end{cases}$$

We prove the uniqueness of NE solution by showing that J_{low} is a strictly copositive matrix. We first give the explicit-form expression of $\nabla^2_{ii} U_i$ and $\nabla^2_{ij} U_i$ as follows:

$$\nabla^2_{ii} U_i = U^i_{xx} + U^i_{yy}, \qquad \nabla^2_{jj} U_j = U^j_{xx} + U^j_{yy},$$

$$\nabla^2_{ij} U_i = \frac{\partial^2 U_i}{\partial x_i \partial y_j} + \frac{\partial^2 U_i}{\partial x_i \partial x_j} + \frac{\partial^2 U_i}{\partial y_i \partial y_j} + \frac{\partial^2 U_i}{\partial x_i \partial y_j},$$

$$\nabla^2_{ji} U_j = \frac{\partial^2 U_j}{\partial x_j \partial y_i} + \frac{\partial^2 U_j}{\partial y_j \partial x_i} + \frac{\partial^2 U_j}{\partial x_j \partial y_i} + \frac{\partial^2 U_j}{\partial y_j \partial x_i}.$$

W.L.O.G. we show that the second-order J_{low} is strictly copositive.The uniqueness to generalized cases can be simply proved using induction, due to the repetitive pattern of the objective function U_i. By the symmetry given in Eq. (20), J_{low} can be written into the form:

$$J_{low} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},$$

where

$$a_{11} = \inf_{(x_1, y_1) \in K} |\nabla^2_{11} U_1|, \quad a_{22} = \inf_{(x_2, y_2) \in K} |\nabla^2_{22} U_2|,$$

$$a_{12} = a_{21} = (-\frac{1}{2}) \inf_{(x_1, y_1) \in K}^{(x_2, y_2) \in K} (|\nabla^2_{12} U_1| + |\nabla^2_{21} U_2|).$$

It suffices to show that $a_{11}, a_{22} \geq 0$ and $a_{12} + \sqrt{a_{11} a_{22}} > 0$, where the non-negativity of the first two terms are trivial.

$$a_{12} + \sqrt{a_{11} a_{22}} = \inf_{(x_1, y_1) \in K}^{(x_2, y_2) \in K} R(1 - \beta)[1 - 2(S - x_i - y_i)]/S^2 + \beta(2x_i - X)/Y^3 > 0.$$

Then, J_{low} is strictly copositive as shown above. Since we have shown that G is continuously differentiable with the derivatives bounded on K (as the derivatives are all linear on the compact solution space K), G is strictly monotone. Therefore OP_{MINER} has at most one solution. Now, we conclude our proof since the uniqueness of NE immediately follows by combining *Claim 1* and *Claim 2*.

REFERENCES

[1] P Andrew. 2019. What is Burstcoin?

[2] Juan Benet. 2014. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561* (2014).

[3] Miguel Castro and Barbara Liskov. 2002. Practical Byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)* (2002).

[4] Bram Cohen and Krzysztof Pietrzak. 2019. The Chia Network Blockchain.

[5] Christian Decker, Jochen Seidel, and Roger Wattenhofer. 2016. Bitcoin meets strong consistency. In *Proceedings of the 17th International Conference on Distributed Computing and Networking*. ACM.

[6] Jianbo Du, Liqiang Zhao, Jie Feng, and Xiaoli Chu. 2018. Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee. *IEEE Transactions on Communications* 66, 4 (2018), 1594–1608.

[7] Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert Van Renesse. 2016. Bitcoin-ng: A scalable blockchain protocol. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*. 45–59.

[8] Franco Giannessi and Antonino Maugeri. 1995. *Variational inequalities and network equilibrium problems*. Springer.

[9] Google. 2019. Google Cloud. https://cloud.google.com/

[10] Tai Manh Ho, Nguyen H Tran, Cuong T Do, SM Ahsan Kazmi, Tuan LeAnh, and Choong Seon Hong. 2015. Data offloading in heterogeneous cellular networks: Stackelberg game based approach. In *2015 Asia-Pacific Network Operations and Management Symposium*. IEEE, 168–173.

[11] Markus Jakobsson and Ari Juels. 1999. Proofs of work and bread pudding protocols. In *Secure Information Networks*. Springer, 258–272.

[12] Suhan Jiang, Xinyi Li, and Jie Wu. 2019. Hierarchical Edge-Cloud Computing for Mobile Blockchain Mining Game. In *Proc. of the 39th IEEE International Conference on Distributed Computing Systems (ICDCS 2019)*, Vol. 15.

[13] Ari Juels and Burton S Kaliski Jr. 2007. PORs: Proofs of retrievability for large files. In *Proceedings of the 14th ACM conference on Computer and communications security*. Acm.

[14] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. 2017. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*. Springer.

[15] Sunny King and Scott Nadal. 2012. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August* (2012).

[16] Daniel Larimer. 2014. Delegated proof-of-stake (dpos). *Bitshare whitepaper* (2014).

[17] Liqing Liu, Zheng Chang, Xijuan Guo, Shiwen Mao, and Tapani Ristaniemi. 2017. Multiobjective optimization for computation offloading in fog computing. *IEEE Internet of Things Journal* 5, 1 (2017), 283–294.

[18] Mengting Liu, F Richard Yu, Yinglei Teng, Victor CM Leung, and Mei Song. 2018. Joint computation offloading and content caching for wireless blockchain networks. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops*.

[19] Yang Liu, Changqiao Xu, Yufeng Zhan, Zhixin Liu, Jianfeng Guan, and Hongke Zhang. 2017. Incentive mechanism for computation offloading using edge computing: a Stackelberg game approach. *Computer Networks* 129 (2017), 399–409.

[20] Andrew Miller, Ari Juels, Elaine Shi, Bryan Parno, and Jonathan Katz. 2014. Permacoin: Repurposing bitcoin work for data preservation. In *2014 IEEE Symposium on Security and Privacy*. IEEE.

[21] Satoshi Nakamoto et al. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

[22] Sunoo Park, Krzysztof Pietrzak, Albert Kwon, Joël Alwen, Georg Fuchsbauer, and Peter Gazi. 2018. Spacemint: A cryptocurrency based on proofs of space. *Financial Cryptography and Data Security* (2018).

[23] Rafael Pass and Elaine Shi. 2017. Hybrid consensus: Efficient consensus in the permissionless model. In *31st International Symposium on Distributed Computing (DISC 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[24] Fahad Saleh. 2019. Blockchain without waste: Proof-of-stake. *Available at SSRN 3183935* (2019).

[25] Ali Shoker. 2017. Sustainable blockchain through proof of exercise. In *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*. IEEE, 1–9.

[26] Lingyang Song, Dusit Niyato, Zhu Han, and Ekram Hossain. 2014. Game-theoretic resource allocation methods for device-to-device communication. *IEEE Wireless Communications* 21, 3 (2014), 136–144.

[27] Youming Sun, Hongxiang Shao, Xin Liu, Jian Zhang, Junfei Qiu, and Yuhua Xu. 2015. Traffic Offloading in Two-Tier Multi-Mode Small Cell Networks over Unlicensed Bands: A Hierarchical Learning Framework. *TIIS* 9, 11 (2015), 4291–4310.

[28] Ken's Tech Tips. 2018. Download Speeds: What Do 2G, 3G, 4G and 5G Actually Mean?

[29] Xiumin Wang, Xiaoming Chen, Weiwei Wu, Ning An, and Lusheng Wang. 2016. Cooperative application execution in mobile cloud computing: A Stackelberg game approach. *IEEE Communications Letters* 20, 5 (2016), 946–949.

[30] Qiufen Xia, Weifa Liang, Zichuan Xu, and Bingbing Zhou. 2014. Online algorithms for location-aware task offloading in two-tiered mobile cloud environments. In *Proceedings of the 2014 IEEE/ACM 7th international conference on utility and cloud computing*. IEEE Computer Society, 109–116.

[31] Liang Xiao, Caixia Xie, Tianhua Chen, Huaiyu Dai, and H Vincent Poor. 2016. A mobile offloading game against smart attacks. *IEEE Access* 4 (2016), 2281–2291.

[32] Zehui Xiong, Shaohan Feng, Dusit Niyato, Ping Wang, and Zhu Han. 2018. Optimal pricing-based edge computing resource management in mobile blockchain. In *2018 IEEE International Conference on Communications*.

[33] Huaqing Zhang, Yong Xiao, Shengrong Bu, Dusit Niyato, Richard Yu, and Zhu Han. [n.d.]. Fog computing in multi-tier data center networks: a hierarchical game approach. In *2016 IEEE international conference on communications*.

[34] Xiaonan Zhang, Linke Guo, Ming Li, and Yuguang Fang. 2016. Social-enabled data offloading via mobile participation-a game-theoretical approach. In *2016 IEEE Global Communications Conference*.