Bitcoin Fee Decisions in Transaction Confirmation Queueing Games Under Limited Multi-Priority Rule

Juanjuan Li, Yong Yuan* , Fei-Yue Wang

Abstract- **In the Bitcoin system, transaction fees serve not only as the fundamental economic incentive to stimulate miners, but also as an important tuner for the Bitcoin system to define the priorities in the transaction confirmation process. In this paper, we aim to study the priority rules for queueing transactions based on their associated fees, and in turn users' decision-making in formulating their fees in the transaction confirmation queueing game. Based on the queueing theory, we first analyzed the waiting time of users under non-preemptive limited multi-priority (LMP) rule, which is formulated to adjust users' waiting time over different priorities. We then established a game-theoretical model, and analyze users' equilibrium fee decisions. Towards the end, we conducted computational experiments to validate the theoretical analysis. Our research findings can not only help understand users' fee decisions under the LMP rule, but also offer useful managerial insights in optimizing the queueing rules of Bitcoin transactions.**

Keywords- **Bitcoin; Blockchain; Transaction fees; Queueing game; The limited multi-priority rule**

I. Introduction

In 2008, an anonymous researcher with the pseudonym of Satoshi Nakamoto designed a peer-to-peer decentralized blockchain network named Bitcoin [1]. It is considered to be capable of dealing with the security and privacy concerns [2, 3, 21, 4], thus has attracted both the academic and industrial attentions. Within the Bitcoin blockchain system, miners (both individual-level solo miners and group-level mining pools) compete to solve computational puzzles via searching a random number that satisfies specific difficulty requirements using a brute force approach, and this process is widely known as proof-of-work mining [6]. New blocks are created via mining and appended to the main-chain of previously agreed upon blocks, creating a complete record of all data updatings that have ever taken place [7]. Any transaction is allowed to be recorded into a block and thus be confirmed only after being successfully verified by all miners [8, 17].

Essentially, the Bitcoin system can be viewed as a queueing system of transactions with their priorities defined mainly on the associated fees [10]-[12]. Bitcoin users usually submit transactions with associated fees for faster confirmation, while miners determine the transaction confirmation according to a predetermined priority rule, in which transaction fee is one of the most important influential factors. Since the Bitcoin block size is generally restricted [13], the number of transactions that miners can confirm and record into one block is limited. Therefore, revenue-maximizing miners naturally first confirm those transactions with higher fees.

In the transaction confirmation process, transaction fee plays a critical role. For miners, transaction fees that currently serve as alternative reward schemes will inevitably develop to be the most important incentive as the new block reward gradually decreases [13]. For users, transaction fees greatly influence their priorities and in turn waiting time. It has been speculated that higher fees will lead to faster confirmation [14], while exorbitant transaction fees will render Bitcoin uneconomical for micro payments [15, 16]. Therefore, there is a critical need for researchers to study transaction fees and their impacts on participants' strategies and revenues.

In this paper, we established a game-theoretical model to help understand the users' transaction fee decisions in the transaction confirmation queueing games, based on the analysis of users' waiting time with respect to the transaction fees using the queueing theory. In our model, we introduced a priority rules mainly defined on the transaction fees, namely the non-preemptive limited multi-priority (LMP) rule. It is easy to understand that when a specific priority has a lot of transactions, their expected waiting time cannot decrease significantly, instead, the expected waiting time for those with lower priorities may be greatly prolonged. Targeting to avoid the over-long retention of the low-priority transactions, we introduce the LMP rule in this paper aiming to adjust users' waiting time over different priorities. We also designed computational experiments to further investigate users' equilibrium fee decisions.

The remainder of this paper is organized as follows. Section II analyzed users' waiting time under the LMP rule. Section III investigated the transaction confirmation queueing game played by the users. Section IV conducted experiments to validate the theoretical analysis. Section V concluded this paper and presented the future work.

II. Users' Waiting Time Under Non-Preemptive Limited Multi-Priority

In this section, we first consider that the confirmation priority of n users' transactions are completely dependent on transaction fee f in Bitcoins, which is formulated as follows:

$$k(i) = rank(f^i \mid f^1 >, .., > f^i, ... > f^n) \tag{1}$$

where, $k(i)$ represents that the user i is with the kth priority to have his/her transaction confirmed and recorded into a block.

This work is supported by National Key Research and Development Program of China (Grant 2018AAA0101400) and National Natural Science Foundation of China (Grant 61533019, 71702182)

Juanjuan Li, Yong Yuan (corresponding author), Fei-Yue Wang is with The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China; Qingdao Academy of Intelligent Industries, Qingdao, China. Emails: juanjuan.li@ia.ac.cn, yong.yuan@ia.ac.cn, feiyue.wang@ia.ac.cn

978-1-7281-2853-5/19/$31.00©2019 IEEE

There are K priority classes of arriving unconfirmed transactions, and those who submit no transaction fees are with the lowest priority. The transaction with the kth priority has non-preemptive priority over the priority $k+\Delta, \Delta > 0$. Within a priority class, users pay the identical fee, and transactions are confirmed following the first-come-first-served (FCFS) rule. Generally, the transactions with the same transaction fees paid by users are with the same priority.

We consider each user submitting only one transaction. Users with the kth priority submit transactions according to the independent Poisson process with the arrival rate of γ_k, and thus the total submission rate of memory pool is $\Gamma = \sum_k \gamma_k$. Assuming that one block consists of only one transaction and the transaction confirmation rate is equal to the new block arrival rate Λ. Only if $\Gamma < \Lambda$, the memory pool will converge and stablize in the long run. This is the crucial prerequisite for our following analysis.

We introduce a parameter β to define the limited priority. Specifically, the first transaction with the priority $k+1$ can jump forward in the queue to get confirmed after successively waiting for β_k transactions with the priority k, but transactions with the priority $k+2$ cannot move up until all transactions with priority k are confirmed. Here, β_k is an integer in the interval of $[1, N]$, and can be adjusted according to the arrival rate γ_k and γ_{k+1}. Usually a higher γ_k will result in a smaller β_k while a higher γ_{k+1} will lead to a higher β_k. Intuitively, the queueing rule of the limited priority should satisfy the condition that $\gamma_k > \beta_k \gamma_{k+1}$, otherwise, it cannot exert any effect. As follows, we discuss the users' waiting time under non-preemptive limited multi-priority.

First, we discuss the case that the user i submitting a transaction confirmation request in the steady memory pool is with the priority k. Denote his/her waiting time as t_k. Here, we can line up users' transactions with the priority k and $k+1$ together to make further discussion of the following three subcases.

Case 1(1): When the user i submits a confirmation request, transactions with the priority $k+1$ are enough to line up at the queueing interval β_k, e.g., $L_k < \beta_k L_{k+1}$. Then, the user's waiting time is

$$t1_k = \rho_k t1_k + \frac{\rho}{\Lambda}. \tag{2}$$

Here, $\rho_k t1_k$ is the confirmation time for the waiting transactions with the priority k ranked ahead of it, and ρ/Λ is the waiting time to expect those in-service transactions to be confirmed. The further calculation of $t1_k$ gets that

$$t1_k = \frac{\rho}{\Lambda(1-\rho_k)}. \tag{3}$$

Case 1(2): When the user i submits a confirmation request, transactions with the priority $k+1$ are not enough to line up at the queueing interval β_k, and this situation keeps unchanged during his/her waiting period. That is, $L_k \geq \beta_k(L_{k+1} + \rho_{k+1}L_k/\rho_k)$. Under this case, the user's waiting time can be computed by

$$t2_k = \rho_k t2_k + \frac{\rho}{\Lambda} + \rho_{k+1} t2_k. \tag{4}$$

We have $\rho_k t2_k$ representing the confirmation time for the waiting transactions with the priority k ranked ahead of it, and $\rho_{k+1} t2_k$ indicating the time delay due to the queue jumping of transactions with the priority $k+1$. Then, we obtain

$$t2_k = \frac{\rho}{\Lambda(1-\rho_k-\rho_{k+1})}. \tag{5}$$

Case 1(3): When the user i submits a confirmation request, transactions with the priority $k+1$ are not enough to line up at the queueing interval β_k, but the newly arriving transactions with the priority $k+1$ during his/her waiting period make queueing at the interval β_k be possible. That is, $\beta_k L_{k+1} \leq L_k < \beta_k(L_{k+1}+\rho_{k+1}L_k/\rho_k)$. Then, we get the user's waiting time as

$$t3_k = \rho_k t3_k + \frac{\rho}{\Lambda} + \frac{J_{k+1}}{\Lambda}. \tag{6}$$

Similarly, $\rho_k t2_k$ is the confirmation time for the waiting transactions with the priority k ranked ahead of it. Also, the user i's waiting time is prolonged due to the queue jumping of transactions with the priority $k+1$, and the number of queue jumpers is

$$J_{k+1} = \frac{\gamma_k t3_k}{\beta_k} - \gamma_{k+1} t3_{k+1}. \tag{7}$$

Accordingly, there is

$$\frac{J_{k+1}}{\Lambda} = \frac{\rho_k t3_k}{\beta_k} - \rho_{k+1} t3_{k+1}. \tag{8}$$

Substituting it into $t3_k$ to get

$$t3_k = \rho_k t3_k + \frac{\rho}{\Lambda} + \frac{\rho_k t3_k}{\beta_k} - \rho_{k+1} t3_{k+1}. \tag{9}$$

Then, through further calculations, we obtain

$$t3_k = \frac{\rho}{\Lambda(1-\rho_k-\frac{\rho_k}{\beta_k})} - \frac{\rho_{k+1} t3_{k+1}}{1-\rho_k-\frac{\rho_k}{\beta_k}}. \tag{10}$$

Next, we discuss another case that a confirmation request in the steady memory pool submitted by the user i is with the priority $k+1$. Denote his/her waiting time as t_{k+1}. Here, transactions with two different priorities should queue separately. As follows, we will conduct the analysis of three subcases in detail.

Case 2(1): When the user i submits a confirmation request, transactions with the priority $k+1$ are not enough to line up at the queueing interval β_k. That is, $L_k \geq \beta_k L_{k+1}$. Then, his/her waiting time is computed by

$$\hat{t}1_{k+1} = \rho_{k+1}\hat{t}1_{k+1} + \frac{J_k}{\Lambda} + \frac{\rho}{\Lambda}. \tag{11}$$

$\rho_{k+1}\hat{t}1_{k+1} + J_k/\Lambda$ is the confirmation time for the waiting transactions with the priority k and $k+1$ ranked ahead of it, where J_k is the numbers of transactions with the priority k ranked ahead. We have $J_k = \beta_k + \epsilon$, and $\epsilon \in (0, \beta_k)$. Accordingly, there is $J_k/\Lambda = \beta_k \rho_{k+1}\hat{t}1_{k+1} + \epsilon/\Lambda$, and substitute it to $\hat{t}1_{k+1}$ to get

$$\begin{aligned} \hat{t}1_{k+1} &= (\beta_k+1)\rho_{k+1}\hat{t}1_{k+1} + \frac{\epsilon}{\Lambda} + \frac{\rho}{\Lambda} \\ &= \frac{\epsilon+\rho}{\Lambda(1-\rho_{k+1}-\beta_k\rho_{k+1})}. \end{aligned} \tag{12}$$

Case 2(2): When the user i submits a confirmation request, transactions with the priority $k+1$ are enough to line up at the queueing interval β_k, and the situation keeps unchanged during his/her waiting period. That is, $L_k+\rho_k L_{k+1}/\rho_k+1 < \beta_k L_{k+1}$. Then, the user's waiting time is computed by

$$\hat{t}2_{k+1} = \rho_k \hat{t}2_k + \rho_{k+1}\hat{t}2_{k+1} + \frac{\rho}{\Lambda} + \rho_k \hat{t}2_{k+1}, \tag{13}$$

where $\rho_k \hat{t}2_k + \rho_{k+1}\hat{t}2_{k+1}$ is the confirmation time for the waiting transactions with the priority k and $k+1$ ranked ahead of it, and $\rho_k \hat{t}2_{k+1}$ is the time delay due to the newly submitted transactions with the priority k during his/her waiting period. Then, through further computing, we obtain

$$\hat{t}2_{k+1} = \frac{\rho}{\Lambda(1-\rho_k-\rho_{k+1})} + \frac{\rho_k \hat{t}2_k}{(1-\rho_k-\rho_{k+1})}. \tag{14}$$

Case 2(3): When the user i submits a confirmation request, transactions with the priority $k+1$ are enough to line up at the queueing interval β_k, but they get to be not enough after the arrival of more transactions with the priority k during the waiting period. That is, $L_k < \beta_k L_{k+1} \le L_k + \frac{\rho_k}{\rho_k+1} L_{k+1}$. Then, the user's waiting time is

$$\hat{t}3_{k+1} = \rho_k t3_{k+1} + \rho_{k+1}\hat{t}3_{k+1} + \frac{\rho}{\Lambda} + \frac{\hat{J}_k}{\Lambda}. \tag{15}$$

Similarly, $\rho_k t3_{k+1} + \rho_{k+1}\hat{t}3_{k+1}$ refers to the confirmation time for the unconfirmed transactions with the priority k and $k+1$ ranked ahead of it. Also, the user i's waiting time is prolonged due to $\hat{J}_k$ newly arrived transactions with the priority k, where

$$\hat{J}_k = \beta\gamma_{k+1} t3_{k+1} - \gamma_k \hat{t}3_k + \epsilon. \tag{16}$$

Accordingly, we can calyculate the time delay $\hat{J}_k/\Lambda$, and substitute it to $t3_{k+1}$ to get

$$\begin{aligned} \hat{t}3_{k+1} &= (\beta_k+1)\rho_{k+1}\hat{t}3_{k+1} + \frac{\rho}{\Lambda} + \frac{\epsilon}{\Lambda} \\ &= \frac{\epsilon+\rho}{\Lambda(1-\rho_{k+1}-\beta_k\rho_{k+1})}. \end{aligned} \tag{17}$$

From the above analysis, we find that $\hat{t}1_{k+1} = \hat{t}3_{k+1}$. Also, it is easy to prove that $t1_k = \hat{t}2_k$, since the transaction with the priority k has the same average waiting time under Case 1(1) and Case 2(2). Similarly, we have $t3_{k+1} = \hat{t}1_{k+1}$. Substituting these conditions to $t3_k$ and $\hat{t}2_{k+1}$, respectively, we obtain

$$\begin{aligned} t3_k = & \frac{\rho}{\Lambda(1-\rho_k-\frac{\rho_k}{\beta_k})} \\ & - \frac{\rho_{k+1}(\epsilon+\rho)}{(1-\rho_k-\frac{\rho_k}{\beta_k})(1-\rho_{k+1}-\beta_k\rho_{k+1})}, \end{aligned} \tag{18}$$

$$\begin{aligned} \hat{t}2_{k+1} = & \frac{\rho}{\Lambda(1-\rho_k-\rho_{k+1})} \\ & + \frac{\rho_k\rho}{\Lambda(1-\rho_k-\rho_{k+1})(1-\rho_k)}. \end{aligned} \tag{19}$$

Since the transaction confirmation requests with each priority are arriving according to an independent Poisson process, there is $p(L_k = m) = \rho_k^m(1-\rho_k)$. Based on this, we have

$$\begin{aligned} p1 &= p(L_k < \beta_k L_{k+1}) \\ &= 1 - p(L_k \ge \beta_k L_{k+1}) \\ &= 1 - \sum_{m=0}^{\infty} \rho_k^{\beta_k m}\rho_{k+1}^m(1-\rho_{k+1}) \\ &= 1 - \frac{1-\rho_{k+1}}{1-\rho_k^{\beta_k}\rho_{k+1}}. \end{aligned} \tag{20}$$

Consequently, we obtain

$$\hat{p}1 = \frac{1-\rho_{k+1}}{1-\rho_k^{\beta_k}\rho_{k+1}}. \tag{21}$$

Analogously, we can compute the probability of Case 1(2) and 2(2) to get

$$p2 = \frac{1-\rho_{k+1}}{1-\rho_k^{\beta_k\rho_k/(\rho_k-\beta_k\rho_{k+1})}\rho_{k+1}}, \tag{22}$$

$$\hat{p}2 = 1 - \frac{1-\rho_{k+1}}{1-\rho_k^{\beta_k-\rho_k/\rho_{k+1}}\rho_{k+1}}. \tag{23}$$

Also, the probability of Case 1(1) is equal to the sum of that of Case 2(2) and 2(3), that is $p1 = \hat{p}2 + \hat{p}3$; similarly, we have $\hat{p}1 = p2 + p3$. Besides, we have $p1 + \hat{p}1 = 1$. Based on these discussions, we can then figure out the probability of Case 1(3) and 2(3) through $p3 = 1 - p1 - p2$ and $\hat{p}3 = 1 - \hat{p}1 - \hat{p}2$, respectively.

Under a certain β_k, we can figure out the average waiting time of the user with the priority k and $k+1$, which are $t_k = p1t1_k + p2t2_k + p3t3_k$ and $t_{k+1} = \hat{p}1\hat{t}1_{k+1} + \hat{p}2\hat{t}2_{k+1} + \hat{p}3\hat{t}3_{k+1}$, respectively. Using the recursive method, we can finally get the waiting time for users with all priorities.

III. The Transaction Confirmation Queueing Game Under The LMP Rule

In the memory pool, the users play the transaction confirmation queueing game through determining the proper transaction fees to maximize their expected profits. For the user i, only if his/her transaction is confirmed and recorded to the blockchain, the revenue v^i can be generated, which will be then subtracted by the transaction fee f^i and the time cost $\alpha^i * (waiting \quad time)$ to formulate the user's expected profit as follows,

$$R^i = v^i - f^i - \alpha^i * (waiting \quad time). \tag{24}$$

Here, α^i represents the user's unit time cost.

The users' fee decision mainly lies on the balance between the cost of transaction fees and the gain from shortened waiting time. If the waiting time is not long or the user is patient enough (i.e., the unit time cost is low), the user prefers not to pay the transaction fee; while if the waiting time is significantly long or the user is impatient (i.e. the unit time cost is very high), the user would like to pay the transaction fee to jump forward in the waiting queue so as to shorten the waiting time.

In the transaction confirmation queueing game with non-preemptive limited multi-priority (The LMP game for short), the profit of the user is formulated as

$$R^i = v^i - f^i - \alpha^i t_{k(i)}. \tag{25}$$

In the LMP game, if k is large, it is very difficult for us to compute the users' waiting time $t_{k(i)}$. Therefore, we simplify the game to be with two priority classes to conduct the following analysis, where the fee-paying users are with the high priority and the users not paying the fees are with the low priority.

In the same way, the LMP game is also a finite game, therefore, Nash equilibrium exists.

Theorem 1. *If $f > \max[\alpha^{1(i)}(w-t_1)]$, the LMP game has the Nash equilibrium with no user paying the transaction fee.*

Proof. In this game, the equilibrium of no user paying the transaction fee has the condition that

$$f > \alpha^i(w-t_1), \quad \forall i \in N, \tag{26}$$

which equates to that $f > \max[\alpha^i(w-t_1)] = \max[\alpha^{1(i)}(w-t_1)]$, where $\alpha^{1(i)}$ represents the unit time cost of the user i with the high priority. □

Theorem 2. *If $0 < f < \min[\alpha^{2(i)}(t_2-w)]$, the LMP transaction confirmation queueing game has the Nash equilibrium with all users paying the transaction fees.*

Proof. For the Nash equilibrium of all users paying the transaction fees, if one user deviates to quit paying, he/she will definitely suffer a loss. As such, there must be the following condition satisfied for all users.

$$v^i - f - \alpha^i w > v^i - \alpha^i t_2 \tag{27}$$

Then, we obtain

$$f < \alpha^i(t_2-w), \quad \forall i \in N. \tag{28}$$

From the above condition, we derive that $f < \min[\alpha^{2(i)}(t_2-w)]$, where $\alpha^{2(i)}$ represents the unit time cost of the user i with the low priority. □

Theorem 3. *If $\max[\alpha^{2(i)}(t_2-t_1), \alpha^{1(i)}(t_2-w)] < f < \min[\alpha^{1(i)}(t_2-t_1), \alpha^{2(i)}(w-t_1)]$, the LMP game has the Nash equilibrium with some users paying the transaction fees.*

Proof. Under the equilibrium with some users paying the transaction fees, we have the following conditions. For the users with the high priority, there is

$$v^{1(i)} - f - \alpha^{1(i)}t_1 \geq v^{1(i)} - \alpha^{1(i)}t_2; \tag{29}$$

and for the users with the low priority, there is

$$v^{2(i)} - \alpha^{2(i)}t_2 \geq v^{2(i)} - f - \alpha^{2(i)}t_1. \tag{30}$$

Otherwise, the users have incentives to change their transaction fee decisions to get higher profits. Through calculations, we get

$$\alpha^{2(i)}(t_2-t_1) \leq f \leq \alpha^{1(i)}(t_2-t_1). \tag{31}$$

According to the analysis of the former two cases, we can also get another condition for the equilibrium of some users paying the transaction fees.

$$\alpha^i(t_2-w) \leq f \leq \alpha^i(w-t_1). \tag{32}$$

Since both conditions should be satisfied for every user, then we obtain

$$\begin{aligned} &\max[\alpha^{2(i)}(t_2-t_1), \alpha^{1(i)}(t_2-w)] \leq f \\ &\leq \min[\alpha^{1(i)}(t_2-t_1), \alpha^{2(i)}(w-t_1)]. \end{aligned} \tag{33}$$

□

Theorem 4. *In a profit-maximizing blockchain system, the optimal transaction fee in the LMP game should be $f^* = argmax[Nf^*(1), \theta(f^*(2))Nf^*(2)]$.*

Proof. Given a transaction fee f, we can get the corresponding transaction confirmation rate Λ and the expected waiting time t_1 and t_2, under which the equilibrium transaction fee of each user is achieved as f^i. Then, the total transaction fee got by the miners is $\sum_i f^i$.

Through analyzing users' fee decisions, we can also find the optimal transaction fee for the blockchain system [17]. With the purpose of the profit maximization, the Nash equilibrium with no user paying the transaction fee is not desirable for miners. Consequently, the optimal transaction fee f^* should satisfy the condition that $0 < f^* \leq \alpha^i(w-t_1)$. Since all fee-paying users are with the identical transaction fee in the LMP game, we have $\sum_i f^i = \theta(f)Nf$, where $\theta(f)$ is the percentage of fee-paying users under the transaction fee f. Therefore, we obtain the optimal transaction fee as $f^* = \underset{0<f\leq\alpha^i(w-t_1)}{argmax} [\theta(f)Nf]$.

The optimal transaction fee may be a relatively low fee making all users pay or a relatively high fee making some users pay. Under the former case, we have the optimal transaction fee as $f^*(1) = \min[\alpha^i(\tilde{t}_2-w)] - \varepsilon$, where ε is an infinitely small positive number. Meanwhile, the corresponding maximal profit is $P^*(1) = Nf^*(1)$. Under the later case, the optimal transaction fee $f^*(2)$ should be set to achieve the maximal profit $\theta(f^*(2))Nf^*(2)$, where $f^*(2) \in [\alpha^i(\tilde{t}_2-w), \alpha^i(w-t_1)]$. □

IV. Experiments

In this section, we design computational experiments to conduct an in-depth analysis on the users' waiting time and equilibrium transaction fees in the transaction confirmation queueing game [18, 19]. Using the real-world data collected from https://blockchain.info/stats, we can figure out that the daily submission rate is about 257987 and the daily confirmation rate is about 258204, which serve as the basis of the following computational experiments. In this section, we take the transaction confirmation queueing game with 2 priorities for example, where the high priority is got through paying the transaction fee while the low priority is with no transaction fee. Under these assumptions, we study the users' waiting time under different distributions of these two priorities' arrival rates.

In the LMP game, we consider the example of the limited priority parameter $\beta = 3$, and the rule can executed only when the arrival rate of the low priority exceeds that of the high priority. Under this condition, the average waiting time of all users in the LMP game is 1.8428 on average.

TABLE I
Users' waiting time in the MLP game (Unit: minutes)

	Mean	Range	Std. Error
t_1	0.0331	[0.0056, 1.8406]	0.0958
t_2	10.7250	[1.8429, 609.5886]	31.7668

The experimental results are summarized in Table I, where t_1 represents the waiting time of the high-priority users and t_2 represents the waiting time of the low-priority users.

From this experiment, we can also obtain that

- Users with the high priority has a dominant advantage over those with the low priority on the waiting time.
- The limited priority results in the increase of t_1 and the slightly decrease of t_2, compared with the case that there is no limited priority, which means the limited priority can not distinctly shorten the waiting time of users with the low priority, but distinctly prolong that of users with the high priority on the contrary.
- The limited priority leads to convincing increase of t_1 when the arrival rate of the high priority is low, but does not produce distinct impact on t_2 no matter what is the arrival rate distribution in the LMP game. Because the limited priority does not work in the context of either the low arrival rate of the high priority or the low arrival rate of low priority.

On the basis of the analysis of the users' waiting time, we study the equilibrium of the LMP game. Then, we can get that

- If $f > \alpha^i/9.0909, \forall i \in N$, there is the equilibrium of no user paying transaction fee;
- If $f < \alpha^i/200, \forall i \in N$, there is the equilibrium of all users paying transaction fee;
- If $\alpha^i/200 \leq f \leq \alpha^i/9.0909, \forall i \in N$, there is the equilibrium of some users paying transaction fee.

V. Conclusions and Future Works

The importance of transaction fees is not only reflected in their influence on the individual decisions and profits of both miners and users, but also in their importance to guarantee the vitality and sustanability of the Bitcoin blockchain system. As such, it is of great necessity to study the transaction fees in the transaction confirmation queueing game played by users.

In this paper, we utilize the queueing theory to study the users' waiting time under the LMP rule. Besides, we establish the game-theoretical model to study the equilibrium transaction fee decisions. Finally, we design the computational experiments to validate our theoretical analysis and make the in-depth analysis of the users' equilibrium fee decisions. Our research can not only understand users' fee decisions under the LMP rule, but also offer useful managerial insights in optimizing the queueing rules of Bitcoin transactions.

In the future work, we plan to address the limitations of current research. In fact, the monetary price of the Bitcoin is vital for the participants' decisions because of its high exchange rate with US dollars and high volatility over time. Therefore, we will incorporate the dynamic Bitcoin price into the game-theoretical model to make further study of the transaction fees. Second, we will try to design novel efficient transaction queueing rules to achieve some useful goals, e.g., reducing the waiting time difference between different priorities, shortening the longest waiting time, or decreasing the waiting time deviation of all users. Also, the parallel management method will be adopted to bridge the real-world Bitcoin blockchain system and the artificial system to make an in-depth research of the transaction fees [20]-[23].

References

[1] Nakomoto S. A Peer-to-Peer Electronic Cash System, White Paper, https://bitcoin.org/bitcoin, 2008.

[2] Qin R, Yuan Y, Wang F Y. Research on the selection strategies of blockchain mining pools. IEEE Transactions on Computational Social Systems, 5(3):748-757, 2018.

[3] Li J, Yuan Y, Wang S, Wang F. Transaction Queue game in Bitcoin Blockchain, the 2018 IEEE Intelligent Vehicles Symposium (IV'18), Changshu, China, June 26-30, pp. 114-119, 2018.

[4] Wang F-Y, Yuan Y, Rong C, Zhang J, Qin R, Smith M H, Blockchainized Internet of minds: A new opportunity for cyber-physical-social systems, IEEE Transactions on Computational Social Systems, 5(4):897-906, 2018.

[5] Wang S, Wang J, Wang S, Qiu T, Yuan Y, OuYang L, Guo Y, Wang F-Y. Blockchain-Powered Parallel Healthcare Systems Based on the ACP Approach, IEEE Transactions on Computational Social Systems, 5(4):942-950, 2018.

[6] Cong L W, He Z. Blockchain Disruption and Smart Contracts. Social Science Electronic Publishing, 2017.

[7] Pisa M, Juden M. Blockchain and Economic Development: Hype vs. Reality. Center for Global Development Policy Paper, 2017.

[8] Yuan Y, Wang F-Y. Blockchain: The State of The Art And Future Trends, Acta Automatica Sinica, 42(4):481-494, 2016.

[17] Yuan Y, Wang F-Y. Parallel Blockchain: Concept, Methods and Issues, Acta Automatica Sinica, 43(10):1703-1712, 2017.

[10] Easley D, O'Hara M, Basu S. From Mining to Markets: The Evolution of Bitcoin Transaction Fees. Social Science Electronic Publishing, 2017.

[11] Huberman G, Leshno J, Moallemi C. Monopoly Without a Monopolist: An Economic Analysis of the Bitcoin Payment System. Social Science Electronic Publishing, 2017.

[12] Kasahara S, Kawahara J. Effect of Bitcoin Fee on Transaction-Confirmation Process. Journal of Industrial& Management Optimization, https://doi.org/10.3934/jimo.2018047.

[13] Houy N. The Economics of Bitcoin Transaction Fees. Social Science Electronic Publishing, 2014.

[14] Moser M, Bohme R. Trends, Tips, Tolls. A Longitudinal Study of Bitcoin Transaction Fees. International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2015:19-33.

[15] Kaskaloglu K. Near Zero Bitcoin Transaction Fees Cannot Last Forever. The International Conference on Digital Security and Forensics. 2014:91-99.

[16] Wong J I. New Study: Low Bitcoin Transaction Fees Unsustainable. http://www.coindesk.com/new-study-low-Bitcoin-transaction-fees-unsustainable. Accessed Oct.13, 2014.

[17] Y. Yuan, F. Wang, and D. Zeng, Competitive Analysis of Bidding Behavior on Sponsored Search Markets, IEEE Transactions on Computational Social Systems, 4(3):179-190, 2017.

[18] Li J, Yuan Y, Wang F-Y. A Novel GSP Auction

Mechanism for Ranking Bitcoin Transactions in Blockchain Mining. Decision Support Systems, https://doi.org/10.1016/j.dss.2019.113094, Volume 124, September 2019, 113094.

[19] Li J, Ni X, Yuan Y. The Reserve Price of Ad Impressions in Multi-Channel Real-Time Bidding Markets. IEEE Transactions on Computational Social Systems, 2018, 5(2): 583-592.

[20] Wang F, Zeng D, Yuan Y. An ACP-based Approach for Complexity Analysis of E-commerce System. Complex Systems and Complexity Science, 2008,3.

[21] Wang F, Yuan Y, Rong C, Zhang J. Parallel Blockchain: An Architecture for CPSS-based Smart Societies. IEEE Transactions on Computational Social Systems, 5(2):303-310, 2018.

[22] Wang F Y, Yuan Y, Li J, et al. From intelligent vehicles to smart societies: A parallel driving approach[J]. IEEE Transactions on Computational Social Systems, 2018, 5(3): 594-604.

[23] Zhang J J, Wang F Y, Wang X, et al. Cyber-physical-social systems: The state of the art and perspectives[J]. IEEE Transactions on Computational Social Systems, 2018, 5(3): 829-840.