NAIST-IS-MT1651036

Master's Thesis

Queueing analysis of transaction-confirmation time in Bitcoin

Yoshiaki Kawase

February 1, 2018

Graduate School of Information Science
Nara Institute of Science and Technology

A Master's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
MASTER of ENGINEERING

Yoshiaki Kawase

Thesis Committee:

Professor Shoji Kasahara	(Supervisor)
Professor Kazushi Ikeda	(Co-supervisor)
Associate Professor Masahiro Sasabe	(Co-supervisor)
Assistant Professor Jun Kawahara	(Co-supervisor)

Queueing analysis of transaction-confirmation time in Bitcoin*

Yoshiaki Kawase

Abstract

Bitcoin is a virtual currency based on a transaction-ledger database called blockchain. The blockchain is maintained and updated by mining process in which a number of nodes called miners compete for finding answers of very difficult puzzle-like problem. Transactions issued by users are grouped into a block, and the block is added to the blockchain when an algorithmic puzzle specialized for the block is solved. A transaction is given a priority value according to its attributes such as the remittance amount and fee, and transactions with high priorities are likely to be confirmed faster than those with low priorities. A recent study reveals that newly arriving transactions are not included in the block being under mining. In this paper, we model the mining process with two queueing systems with batch service, analyzing the transaction-confirmation time.

In the first queueing model, we consider an $\mathrm{M/G}^b/1$ with batch service, in which a newly arriving transaction cannnot enter the service facility even when the number of transactions in the service facility does not reach the maximum batch size, i.e., the block-size limit. In this model, the sojourn time of a transaction corresponds to its confirmation time. We consider the joint distribution of the number of transactions in system and the elapsed service time, deriving the mean transaction-confirmation time. Using the result, we derive the recursive formulae of mean transaction-confirmation times of an $\mathrm{M/G}^b/1$ queue with priority service discipline. In numerical examples, we show how the block-size limit affects the transaction-confirmation time. We also discuss how the increase in

*Master's Thesis, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT1651036, February 1, 2018.

micropayments, which are likely to be given low priorities, affects the transaction-confirmation time.

It is reported that the block-generation time, which is equivalent to service time, can be approximated with an exponential distribution. It is, however, known that the variation of transaction arrival interval is larger than that of exponential distribution in real Bitcoin system. We consider the second queueing model, GI/M^b/1, assuming that the transaction arrival intervals are independent and identically distributed (i.i.d.), and follow a general distribution. We define the number of transactions in queue just before a transaction arrival as the system state, deriving the steady distribution and the mean transaction-confirmation time by matrix analytic method. In numerical examples, we compare the result of the theoretical analysis, whose parameter is based on real data, with the result of trace-driven simulation in order to verify our model.

Keywords:

Blockchain, Bitcoin, Queueing Theory, Priority Queueing Theory, Transaction-Confirmation Time

ビットコイン・トランザクション承認時間の待ち行列理論解析*

河瀬 良亮

内容梗概

ビットコインは分散型取引台帳ブロックチェーンに基づいた仮想通貨である。マイナーと呼ばれるノードが難解なパズル的問題を競争して解くことで、ブロックチェーンは維持・更新されている。利用者によって発行されたトランザクションは複数個まとめてブロックに格納され、パズル的問題が解かれたときにそのブロックはブロックチェーンに接続される。送金金額や手数料などに基づいた優先権がトランザクションに与えられており、高優先権のトランザクションは低優先権のトランザクションよりも早く承認される傾向がある。最近の研究によると、新たに到着したトランザクションはマイニング中のブロックには含まれないことが分かった。本研究では、このマイニングプロセスを集団サービスを持つ待ち行列によりモデル化し、トランザクション承認時間の解析を行う。

最初のモデルとして、集団サービスを行う $\mathrm{M/G}^b/1$ を考える。新規トランザクションの到着時点でサービス中の集団サイズが制限未満であってもそのトランザクションは待ち行列で待機するモデルを考える。このモデルでは、トランザクションの系内滞在時間はそのトランザクションの承認にかかる時間に相当する。このモデルに対してサービス中のトランザクション数と待ち行列内トランザクション数の結合分布を考え、それを基に平均トランザクション承認時間を導出する。この結果を用いて、優先権付きの集団サービスを持つ $\mathrm{M/G}^b/1$ についての平均トランザクション承認時間の再帰的な式を導出する。数値例において、ブロックサイズの上限がトランザクション承認時間に与える影響を示す。低優先権になりやすいマイクロペイメントの増加がトランザクション承認時間に与える影響についても議論する。

*奈良先端科学技術大学院大学 情報科学研究科 修士論文, NAIST-IS-MT1651036, 2018 年 2 月 1 日.

関連研究によると、サービス時間であるブロック生成時間間隔は指数分布で近似できることが報告されている。しかしながら、実際のビットコインではトランザクションの到着間隔は指数分布よりも変動が大きいことが知られている。そこで二番目のモデルとして、トランザクションは独立同一な一般分布に従って到着し、マイニングの終了時点で複数のトランザクションが離脱する GI/M^b/1 モデルを考える。トランザクション到着直前の待ち行列内トランザクション数をシステムの状態として定義し, 行列解析法により定常分布および平均トランザクション滞在時間を導出する。数値例において、実データに基づいてパラメータを決定した理論解析結果とシミュレーションを比較することでモデルの妥当性を検証する.

キーワード

ブロックチェーン, ビットコイン, 待ち行列, 優先権付き待ち行列, トランザクション承認時間

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Background of Bitcoin

Bitcoin is an autonomous decentralized virtual currency that does not have a central server or an administrator, and it succeeds to prevent fraud such as multiple payment and impersonation by encryption and peer-to-peer network technologies [1]. Bitcoin can provide immediate and secure service of international money transfer, and it is expected to be used for micropayment such as small amount remittance and billing of a piece of Internet content, due to its low fee [2]. Bitcoin demand grows rapidly in recent years, and the average number of transactions per day in 2016 is 226,000, which is about twice as much as the previous year's 125,000 [3]. In Japan, it is possible to pay with Bitcoin, for example, at home electronics retailer biccamera [4].

The Bitcoin virtual-currency system is based on two data types: transaction and block. A transaction includes information indicating a specific amount of money transferred from sender(s) to recipient(s). On the other hand, a block consists of several transactions, and a newly generated block is confirmed by solving a puzzle-like problem, i.e., a mathematical problem based on a cryptographic hash algorithm or the block generation. This confirmation process is called mining, and a number of nodes called miners compete for finding its answer. The miner who finds its solution first becomes a winner, and adds the new block to the blockchain. At this time, the solution is also included in the new block, and a new mathematical problem for the next mining is created from this newly added block. The difficulty of the puzzle-like problem is automatically adjusted so that mining finishes in 10 minutes on average [5]. The miner who succeeds in mining receives reward called coinbase (currently 12.5 BTC [1] in 2017) and transactions' remittance fees in the block. Note that miners keep mining work [2] in order to get coinbase even when there are no transactions to be included into a block.

1.2. Priority Mechanism

There exists much literature on the introduction of Bitcoin. The readers are referred to e.g. [13]. In this section, we summarize the priority mechanism of default Bitcoin client mechanism [5].

In Bitcoin system, the maximum block size is limited to 1 Mbyte. This implies that when a large number of transactions are issued in one-block mining period, all the transactions cannot be included into a new block. The reference Bitcoin client, called Bitcoin Core, constructs a new block in the following manner. When a miner constructs a new block, the miner selects transactions in the memory pool according to a priority-service discipline based on the amount of remittance, the transaction-data size, transaction fee, and the age of the unspent transaction output (UTXO). The age of a UTXO is the number of blocks that have confirmed since the UTXO was registered in the blockchain.

Let p denote the priority value of a transaction. The value of p is calculated by

$$p = \frac{1}{l} \sum_{k} v_k \cdot a_k,$$

where l is the transaction data size, v_k is the UTXO of kth bitcoin address in the transaction, and a_k is the corresponding UTXO age.

After priority allocation, the transaction is stored into a new block according to the following steps.

[1]BTC is a unit of currency for Bitcoin.

[2]The coinbase, the reward for the winner of a puzzle-like problem, is generated even for a block which includes no user transactions. A primary incentive for miners is to get the coinbase, and hence the miners keep mining work even in the absence of user transactions.

1. 50 Kbytes in the block are allocated to the highest-priority transactions, regardless of transaction fee.

2. Transactions with a fee of at least 0.00001 BTC/Kbyte are added to the block in highest-fee-per-kilobyte transactions first order. This process continues until the block size is no more than DEFAULT_BLOCK_MAX_SIZE, which is set to 750 Kbytes as default but can be changed within 1 Mbyte.

3. When the block size reaches the block-size limit, the transactions that are not included into the block remain in the memory pool of miners. Then the procedure is repeated.

In [14], the authors study trends of Bitcoin transaction fee conventions by analyzing the transaction fees paid with 55.5 million transactions that are recorded in the blockchain. They find that the confirmation time of transactions without fee are longer than those with fee. It is also reported that there are no significant difference between transaction-confirmation times for different fees. If the demand of transactions for micro payment increases in future, those transactions may suffer from a very long confirmation time because payers of micro payment are not willing to pay fee and the resulting priority of their transactions is low.

In order to verify the impact of the priority mechanism on the transaction-confirmation time, we analyzed the transaction data[3] in [3] from October 2013 to September 2015. Table 1.1 shows mean transaction-confirmation times. Here, classless implies the mean confirmation time of all the transactions in the data. In terms of priority classes, we classify a transaction which has more than 0.001 BTC (100,000 satoshis) in output value as high-priority class, and that with output value smaller than or equal to 0.001 BTC as low-priority one.

In Table 1.1, the first (resp. second) row shows the results of period October 2013 to September 2014 (resp. October 2014 to September 2015), and the third row shows the results of two-year data. (In the following, we call the period of October 2013 - September 2014 (resp. October 2014 to September 2015) the 1st (resp. 2nd) period. The two-year period of October 2013 - September 2015 is called the overall one.) We observe that for classless, high, and low priorities, the

[3]In this data, confirmation times of some transactions are smaller than or equal to zero second. We ignore them for calculating the transaction-confirmation time.

Table 1.1. Mean transaction confirmation time for each priority.

Period		Classless [s]	Low priority [s]	High priority [s]
1st	(2013/10/1-2014/9/30)	1060.67797	1947.47115	1044.25043
2nd	(2014/10/1-2015/9/30)	1171.98866	4887.56496	1070.32818
overall	(2013/10/1-2015/9/30)	1124.13286	3888.06977	1059.06133

results of the 2nd period are greater than those of the 1st one. In high priority case, the transaction-confirmation time is almost the same for those periods. In low priority case, the mean transaction-confirmation time for the 2nd period is significantly larger than that for the 1st period. We will discuss these observations in Section 2.4, where we compare the analytical results with trace-driven simulation.

1.3. Scalability Issue

One of technical issues in Bitcoin is low transaction-processing speed due to the maximum block size [6]. Currently the block size is limited to 1 Mbyte and the time for mining is 10 minutes in average. As a result, the number of transactions processed per second is very small. In addition, the Bitcoin system has a transaction-priority mechanism, in which each transaction is prioritized according to its remittance amount, the elapsed time from previous approval, and the transaction-data size. A transaction is included in a block according to its priority value and the fee paid by user in advance [5]. Transactions with low remittance and/or low fee are likely not to be included in a block when the transaction arrival rate is high. It is reported in [3] that the recent block size is approaching the maximum block size, and therefore reducing the transaction size by separating a part from a transaction [7] have been proposed, however, no agreement has been achieved for Bitcoin community[4].

[4] A large portion of miners agree with the segwit2x, in which the limit of block size is extended to 2 Mbytes and the transaction size is reduced by separating signature [3, 7, 19]. The rest of the miners have conducted developing another coin called Bitcoin cash via hard fork in August

In order to consider the scalability issue of Bitcoin, it is important to quantitatively characterize the transaction-confirmation process in which the priority mechanism is taken into consideration. Since transactions are processed in a block basis, the transaction-confirmation process can be modeled as a single-server queueing system with batch service. In terms of the analysis for the single-server queue with batch service, the authors in [8, 9] consider an $M/G^b/1$ queue and analyze the joint distribution of the number of customers in the queue and the elapsed service time.

In [10], the authors consider the impact of a fee amount on the transaction-confirmation time. They model the transaction-confirmation process as an $M/G^b/1$ queue with priority mechanism, deriving the mean transaction-confirmation time for each transaction-priority class. They also show from both measured data and extreme-value theory that the block-generation time follows an exponential distribution. From numerical examples, it is found that most of miners don't follow the transaction inclusion specified for the default Bitcoin client mechanism [5]. In the default mechanism, a newly arriving transaction is included in the block under mining if the number of transactions in the block is smaller than the maximum block size. Comparing measured data and analytical results, it is conjectured that a newly arriving transaction is not included in the block currently processed.

1.4. Queueing Analytical Approaches to Bitcoin

In this thesis, we develop the queueing model of [10] into the ones in which a newly arriving transaction is not served in the current service even when the number of transactions in the server is smaller than the maximum block size. As we noted before, miners keep mining work even when there are no transactions to be included into a block. We assume the multiple-vacation policy in which the server takes vacation repeatedly until a transaction arrives at the system in idle.

Here, we consider two queueing models for analyzing the transaction-confirmation time in Bitcoin. In the first queueing model, we consider an $M/G^b/1$ with batch

1st, 2017. In Bitcoin cash, the block size limit can be immediately changed up to 8 Mbytes [20].

service, in which a newly arriving transaction cannot enter the service facility even when the number of transactions in the service facility does not reach the maximum batch size, i.e., the block-size limit. In this model, the sojourn time of a transaction corresponds to its confirmation time. We consider the joint distribution of the number of transactions in system and the elapsed service time, deriving the mean transaction-confirmation time.

In the Bitcoin system, each transaction is given a priority before confirmation processing. The priority of a transaction is dependent on its amount, the transaction size in bit, the fee for miners, and the age of input coins, i.e., the time elapsed from the latest usage of the coin. For example, if the amount of a transaction is small and the coin in the transaction was used by another recent transaction, the transaction is given a low priority. When the amount of a transaction is high, on the other hand, a high priority is given to the transaction.

In order to investigate the effect of the priority mechanism on the transaction-confirmation time, we further develop the first fundamental queueing model to the one in which priority mechanism is taken into consideration. Using the analytical result of [10], we calculate the mean transaction-confirmation time of an M/G^b/1 queue with priority service discipline, comparing the results with measurement data.

Note that the queueing model of [10] is not work conserving[5], and hence they approximately analyzed the mean transaction-confirmation time of each priority class assuming that the system is work conserving. In the M/G^b/1 with priority mechanism, on the other hand, the system is work conserving and hence we can analyze the mean transaction-confirmation time in a rigorous manner. In numerical examples, we quantitatively evaluate the effect of the block size on the transaction-confirmation time. We also investigate how the demand of low-priority transactions affects the transaction-confirmation time. Especially, we

[5]In [10], it is assumed that a newly arriving transaction is inclued in the block being processed if the resulting number of transactions in the block is smaller than or equal to the maximum block size. Under this assumption, consider the following two sample paths for busy period. The first (resp. second) sample path is that a low-priority (resp. high-priority) transaction arrives at the system in idle and starts a busy period. Please note that the remaining service time of high-priority transactions in the first sample path is likely to be smaller than that in the second one. This implies the system is not work conserving.

classify the transactions into two types according to the remittance amount, and compare the transaction-confirmation time of the analysis result with real data. It is found that the transaction-confirmation time of the low priority takes a larger value than the theoretical one. This result implies that miners tend not to include the transactions with low priority into the block.

It is reported that the block-generation time, which is equivalent to service time, can be approximated with an exponential distribution. It is, however, known that the variation of transaction arrival interval is larger than that of exponential distribution in real Bitcoin system. We consider the second queueing model, GI/M^b/1, assuming that the transaction arrival intervals are independent and identically distributed (i.i.d.), and follow a general distribution. We define the number of transactions in queue just before a transaction arrival as the system state, deriving the steady distribution and the mean transaction-confirmation time by matrix analytic method. In numerical examples, we estimate the inter-arrival time distribution of transactions with the EM algorithm, and verify the validity of the analytical model by comparing the result of analysis with that of trace-driven simulation.

1.5. Overview of Thesis

The rest of this thesis is organized as follows.

In Chapter 2, we analyze the transaction-confirmation time with the M/G^b/1 queueing model. We describe our basic queueing model in Section 2.1, and the analysis of the queueing model is presented in Section 3.2. We consider the transaction-confirmation time of a queueing system with priority mechanism in Section 2.3. Some numerical examples are shown in Section 2.4, and we discuss how the increase in micropayment affects the transaction-confirmation time. We give concluding remarks of the Chapter 2 in Section 2.5.

In Chapter 3, we consider the GI/M^b/1 queue. In section 3.1, we describe the queueing model of transaction processing in detail. In 3.2, we define the system state as the number of transactions in queue just before a transaction arrives at the system, and derive the steady-state distribution by the matrix analytic method. In 3.3 we show some numerical examples, and we summarize Chapter 3 in section 3.4.

We conclude the thesis in Chapter 4, showing our conclusion in Section 4.1, and future work in Section 4.2.

Chapter 2

Analysis of Transaction-Confirmation Time with M/G^b/1 Queueing Model

In this chapter, we model the mining process with a queueing system with batch service, analyzing the transaction-confirmation time. We consider an $M/\mathrm{G}^b/1$ with batch service, in which a newly arriving transaction cannot enter the service facility even when the number of transactions in the service facility does not reach the maximum batch size, i.e., the block-size limit. In this model, the sojourn time of a transaction corresponds to its confirmation time. We consider the joint distribution of the number of transactions in system and the elapsed service time, deriving the mean transaction-confirmation time.

Using the result, we derive the recursive formulae of mean transaction-confirmation times of an M/G^b/1 queue with priority service discipline. In numerical examples, we show how the block-size limit affects the transaction-confirmation time. We also discuss how the increase in micropayments, which are likely to be given low priorities, affects the transaction-confirmation time.

2.1. Basic Queueing Model

In this section, we describe a basic queueing model without priority mechanism. We will use the analytical result of the basic queueing model for deriving mean transaction-confirmation times for a priority queueing model later.

Transactions arrive at the system according to a Poisson process with rate λ. The transactions are grouped into a block, and the block is confirmed when one of miners finds the answer of the puzzle-like problem. We define the block-generation time as the time interval between consecutive block-confirmation time points. Note that the block-generation time can be regarded as the service time for our queueing model.

Let S_i $(i = 1, 2, \ldots)$ denote the ith block-generation time. We assume $\{S_i\}$'s are independent and identically distributed (i.i.d.), and follow a distribution function $G(x)$. Let $g(x)$ denote the probability density function of $G(x)$. The mean block-generation time $E[S]$ is given by

$$E[S] = \int_0^\infty x\,dG(x) = \int_0^\infty xg(x)\,dx.$$

Transactions arriving to the system are served in a batch manner, and the maximum batch size is b.

When a transaction arrives at the system, the transaction enters the queue. The transaction cannot enter the server at its arrival point even when the batch size under service is smaller than b or when the system is idle. In other words, the arriving transaction is served in the next block-generation time or later. This service is regarded as the gated service with multiple vacations [15], in which vacation periods are i.i.d. and follow the same distribution of the service time[1].

2.2. Analysis

In this section, we analyze the transaction-confirmation time for the basic model presented in the previous section.

Let $N_s(t)$ denote the number of transactions in the server at time t, $N_q(t)$ the number of transactions in the queue at time t, and $X(t)$ the elapsed service time at t. We define $P_{m,n}(x,t) (m = 0, 1 \ldots, b,\ n = 0, 1, \ldots,\ x, t \geq 0)$ as

$$P_{m,n}(x,t)\,dx = Pr\left\{N_s(t) = m, N_q(t) = n, x < X(t) \leq x + dx\right\}.$$

[1]Note that the mining process for a block without transactions is identical to that for a block with transactions. This enables us to assume that vacation periods follow the same distribution of the service time.

Let $\xi(x)$ denote the hazard rate of the service time S, which is given by

$$\xi(x) = \frac{g(x)}{1 - G(x)}.$$

When $\lambda E[S] < b$ holds, the system is stable and limiting probabilities exist. Letting $P_{m,n}(x) = \lim_{t\to\infty} P_{m,n}(x,t)$, we obtain from the assumptions the following differential-difference equations

$$\frac{d}{dx} P_{m,n}(x) = -\{\lambda + \xi(x)\} P_{m,n}(x) + \lambda P_{m,n-1}(x), \qquad 0 \le m \le b, \quad n \ge 1, \tag{2.1}$$

$$\frac{d}{dx} P_{m,0}(x) = -\{\lambda + \xi(x)\} P_{m,0}(x), \qquad 0 \le m \le b. \tag{2.2}$$

In (2.1), the first term of the right-hand side (r.h.s) is obtained from the event that the numbers of transactions in the server and the queue does not change during a small time interval, and the second term comes from the event that a transaction arrives at the system with $n-1$ transactions in the queue. The equation of (2.2) is the special case of no transaction in the queue, and hence the r.h.s contains only the term for no event occurring during a small time interval.

We have the following boundary conditions at $x = 0$

$$\begin{aligned}
P_{b,n}(0) &= \sum_{m=0}^{b} \int_0^\infty P_{m,n+b}(x)\xi(x)dx, \qquad n \ge 0, \\
P_{m,n}(0) &= 0, \quad m = 0, 1, \ldots, b-1, \quad n \ge 1, \\
P_{k,0}(0) &= \sum_{m=0}^{b} \int_0^\infty P_{m,k}(x)\xi(x)dx, \qquad k = 0, 1, \ldots, b.
\end{aligned}$$

In the above equations, note that $P_{k,0}(0)$ $(0 \le k \le b)$ is the probability that the number of transactions in the server is k and that in the queue is 0 at the beginning of a new service. This event can occur if the number of transactions in the server is m $(0 \le m \le b)$ and that in the queue is k at the end of the previous service.

Similarly, $P_{m,n}(0)$ for $n \ge 1$ is the probability that the number of transactions in the server is m and that in the queue is $n(\ge 1)$ at the beginning of a new service. Reminding that transactions are served in a batch manner with the maximum batch size of b, the event of $P_{m,n}(0)$ for $n \ge 1$ can occur only if the number of transactions in the queue is greater than or equal to $b+n$ at the end of the previous service. That is, the event of non-empty queue at the beginning of a service can occur with the event of b transactions in the server. This is why $P_{m,n}(0) > 0$ for $n \geq 1$ and $m = 0, 1, \ldots, b-1$ is zero.

The normalizing condition is given by

$$\sum_{n=0}^{\infty} \sum_{m=0}^{b} \int_{0}^{\infty} P_{m,n}(x) dx = 1.$$

We define the following probability generating functions (pgf's)

$$P(z_1, z_2; x) = \sum_{n=0}^{\infty} \sum_{m=0}^{b} P_{m,n}(x) z_1^m z_2^n, \tag{2.3}$$

$$P(z_1, z_2) = \int_{0}^{\infty} P(z_1, z_2; x)\, dx. \tag{2.4}$$

From (2.1) and (2.2), we obtain

$$\begin{aligned} &\sum_{n=0}^{\infty} \sum_{m=0}^{b} \frac{d}{dx} P_{m,n}(x)\, z_1^m z_2^n \\ &= \sum_{n=0}^{\infty} \sum_{m=0}^{b} -\{\lambda + \xi(x)\} P_{m,n}(x)\, z_1^m z_2^n + \lambda z_2 \sum_{n=0}^{\infty} \sum_{m=0}^{b} P_{m,n}(x)\, z_1^m z_2^n. \end{aligned}$$

From the above equation and (2.3), we obtain

$$\begin{aligned} \frac{d}{dx} P(z_1, z_2; x) &= -\{\lambda + \xi(x)\} P(z_1, z_2; x) + \lambda z_2 P(z_1, z_2; x) \\ &= -\{\lambda(1 - z_2) + \xi(x)\} P(z_1, z_2; x). \end{aligned}$$

From this differential equation, $P(z_1, z_2; x)$ is given by

$$P(z_1, z_2; x) = P(z_1, z_2; 0) e^{-\lambda(1-z_2)x} \{1 - G(x)\}. \tag{2.5}$$

Multiplying (2.5) by $\xi(x)$ and integrating the equation yield

$$\int_{0}^{\infty} P(z_1, z_2; x) \xi(x)\, dx = P(z_1, z_2; 0) G^*(\lambda - \lambda z_2), \tag{2.6}$$

where $G^*(s)$ is the Laplace-Stieljes transform (LST) of $G(x)$ and given by

$$G^*(s) = \int_0^\infty e^{-sx} dG(x).$$

From (2.3) and (2.6), we obtain

$$P(z_1, z_2; 0)G^*(\lambda - \lambda z_2) = \int_0^\infty \sum_{n=0}^{\infty} \sum_{m=0}^{b} P_{m,n}(x)\xi(x)\,dx\, z_1^m z_2^n. \tag{2.7}$$

Substituting $x = 0$ into (2.3) yields

$$\begin{aligned} P(z_1, z_2; 0) &= \sum_{n=0}^{\infty} \sum_{m=0}^{b} P_{m,n}(0)\, z_1^m z_2^n \\ &= \sum_{n=0}^{\infty} \sum_{m=0}^{b} \int_0^\infty P_{m,n+b}(x)\xi(x)\,dx\, z_1^b z_2^n + \sum_{n=0}^{b-1} \sum_{m=0}^{b} \int_0^\infty P_{m,n}(x)\xi(x)\,dx\, z_1^n. \end{aligned} \tag{2.8}$$

Using (2.7) to the first term of (2.8), we obtain

$$\begin{aligned} &\sum_{n=0}^{\infty} \sum_{m=0}^{b} \int_0^\infty P_{m,n+b}(x)\xi(x)\,dx\, z_1^b z_2^n \\ &= \left(\frac{z_1}{z_2}\right)^b \left\{ P(1, z_2; 0)G^*(\lambda - \lambda z_2) - \sum_{n=0}^{b-1} \sum_{m=0}^{b} \int_0^\infty P_{m,n}(x)\xi(x)\,dx\, z_2^n \right\}. \end{aligned}$$

Then, (2.8) can be rewritten as

$$\begin{aligned} P(z_1, z_2; 0) = &\left(\frac{z_1}{z_2}\right)^b \left\{ P(1, z_2; 0)G^*(\lambda - \lambda z_2) - \sum_{n=0}^{b-1} \sum_{m=0}^{b} \int_0^\infty P_{m,n}(x)\xi(x)\,dx\, z_2^n \right\} \\ &+ \sum_{n=0}^{b-1} \sum_{m=0}^{b} \int_0^\infty P_{m,n}(x)\xi(x)\,dx\, z_1^n. \end{aligned} \tag{2.9}$$

Substituting $z_1 = 1$ into (2.9), we obtain

$$\begin{aligned} P(1, z_2; 0) = &\left(\frac{1}{z_2}\right)^b \left\{ P(1, z_2; 0)G^*(\lambda - \lambda z_2) - \sum_{n=0}^{b-1} \sum_{m=0}^{b} \int_0^\infty P_{m,n}(x)\xi(x)\,dx\, z_2^n \right\} \\ &+ \sum_{n=0}^{b-1} \sum_{m=0}^{b} \int_0^\infty P_{m,n}(x)\xi(x)\,dx. \end{aligned}$$

Multiplying the above equation by z_2^b yields

$$\left\{z_2^b - G^*(\lambda - \lambda z_2)\right\} P(1, z_2; 0) = \sum_{n=0}^{b-1}(z_2^b - z_2^n) \sum_{m=0}^{b} \int_0^\infty P_{m,n}(x)\xi(x)\, dx.$$

From the above equation, we obtain

$$P(1, z_2; 0) = \frac{\sum_{n=0}^{b-1}(z_2^b - z_2^n)\alpha_n}{z_2^b - G^*(\lambda - \lambda z_2)}, \tag{2.10}$$

where α_n is given by

$$\alpha_n = \sum_{m=0}^{b} \int_0^\infty P_{m,n}(x)\xi(x)\, dx.$$

Applying Rouche's theorem [15] to (2.10), we can show that the equation

$$z_2^b - G^*(\lambda - \lambda z_2) = 0, \tag{2.11}$$

has b roots inside $|z_2| = 1 + \epsilon$ for a small real number $\epsilon > 0$. One of them is $z_2 = 1$. Let $z_{2,k}^*(k = 1, 2, ..., b-1)$ denote the root of (2.11). From (2.10), we have the following $b - 1$ equations

$$\sum_{n=0}^{b-1} \left\{ \left(z_{2,k}^*\right)^b - \left(z_{2,k}^*\right)^n \right\} \alpha_n = 0, \qquad k = 1, 2, ..., b-1. \tag{2.12}$$

From (2.9) and (2.10), we have

$$\begin{aligned} P(z_1, z_2; 0) = {}& \left(\frac{z_1}{z_2}\right)^b \left\{ \frac{\sum_{n=0}^{b-1}\left(z_2^b - z_2^n\right)\alpha_n}{z_2^b - G^*(\lambda - \lambda z_2)} G^*(\lambda - \lambda z_2) - \sum_{n=0}^{b-1} \alpha_n z_2^n \right\} \\ & + \sum_{n=0}^{b-1} \alpha_n z_1^n. \end{aligned} \tag{2.13}$$

From (2.4) and (2.5), we obtain

$$\begin{aligned} P(z_1, z_2) &= P(z_1, z_2; 0) \int_0^\infty e^{-\lambda(1-z_2)x}\{1 - G(x)\}\, dx \\ &= P(z_1, z_2; 0) \frac{1 - G^*(\lambda - \lambda z_2)}{\lambda(1 - z_2)}. \end{aligned} \tag{2.14}$$

Multiplying (2.14) by $\lambda(1-z_2)$ and partially differentiating it by z_2, we have

$$\begin{aligned}&\frac{\partial P(z_1,z_2)}{\partial z_2}\lambda(1-z_2) - P(z_1,z_2)\lambda \\ &= \frac{\partial P(z_1,z_2;0)}{\partial z_2}\left\{1-G^*(\lambda-\lambda z_2)\right\} - P(z_1,z_2;0)\frac{\partial G^*(\lambda-\lambda z_2)}{\partial z_2}.\end{aligned}$$

Substituting $z_1 = z_2 = 1$ into the above equation, and noting that $P(1,1) = 1$, we have

$$P(1,1) = P(1,1;0)E[S] = 1.$$

Multiplying (2.14) by $z_2^b(z_2^b - G^*(\lambda-\lambda z_2))$ in order to calculate $P(1,1;0)$, we have

$$\begin{aligned}P(z_1,z_2;0)\,z_2^b\left\{z_2^b - G^*(\lambda-\lambda z_2)\right\} &= z_1^b\sum_{n=0}^{b-1}\left(z_2^b - z_2^n\right)\alpha_n G^*(\lambda-\lambda z_2) \\ &\quad - z_1^b\left\{z_2^b - G^*(\lambda-\lambda z_2)\right\}\sum_{n=0}^{b-1}\alpha_n z_2^n \\ &\quad + z_2^b\left\{z_2^b - G^*(\lambda-\lambda z_2)\right\}\left(\sum_{n=1}^{b-1}\alpha_n z_1^n + \alpha_0 z_1\right).\end{aligned}$$

Partially differentiating the above equation by z_2 and substituting $z_1 = z_2 = 1$, we obtain under the stability condition of $b > \lambda E[S]$

$$P(1,1;0) = \frac{\sum_{n=0}^{b-1}(b-n)\,\alpha_n}{b-\lambda E[S]}.$$

Hence, the normalizing condition is given by

$$\frac{\sum_{n=0}^{b-1}(b-n)\,\alpha_n}{b-\lambda E[S]}E[S] = 1. \tag{2.15}$$

From (2.12) and (2.15), α_n's are uniquely determined. From (2.13) and (2.14), we have

$$\begin{aligned}P(z_1,z_2) = &\left\{\left(\frac{z_1}{z_2}\right)^b \frac{\sum_{n=0}^{b-1}(z_2^b - z_2^n)\alpha_n}{z_2^b - G^*(\lambda-\lambda z_2)}G^*(\lambda-\lambda z_2) - \left(\frac{z_1}{z_2}\right)^b\sum_{n=0}^{b-1}\alpha_n z_2^n\right. \\ &\left. + \sum_{n=0}^{b-1}\alpha_n z_1^n\right\}\frac{1-G^*(\lambda-\lambda z_2)}{\lambda(1-z_2)}.\end{aligned} \tag{2.16}$$

Partially differentiating (2.16) by z_1 and substituting $z_1 = z_2 = 1$, we obtain the mean number of transactions in the server as

$$\begin{aligned}\left(\frac{\partial P(z_1, z_2)}{\partial z_1}\right)_{z_1=1, z_2=1} &= \left\{\frac{\sum_{n=0}^{b-1}(b-n)\alpha_n}{b-\lambda E[S]}E[S]\right\}\lambda E[S] \\ &= \lambda E[S].\end{aligned}$$

Similarly partially differentiating (2.16) by z_2, and substituting $z_1 = z_2 = 1$, we obtain the mean number of transactions in the queue as

$$\begin{aligned}\left(\frac{\partial}{\partial z_2}P(z_1, z_2)\right)_{z_1=1, z_2=1} &= \frac{1}{2(b-\lambda E[S])}\Bigg(\lambda^2 E[S^2] - 2b(b-\lambda E[S]) - b(b-1) \\ &\quad + \sum_{n=0}^{b-1}\Big\{\lambda E[S^2](b-n) + E[S]\{b(b-1) - n(n-1)\} + 2bE[S](b-n)\Big\}\alpha_n\Bigg).\end{aligned}$$

Hence, the mean number of transactions in the system $E[N]$ is given by

$$\begin{aligned}E[N] &= \frac{1}{2(b-\lambda E[S])}\Bigg(\lambda^2 E[S^2] - b(b-1) - 2(b-\lambda E[S])^2 \\ &\quad + \sum_{n=0}^{b-1}\Big\{\lambda E[S^2](b-n) + E[S]\{b(b-1) - n(n-1)\} + 2bE[S](b-n)\Big\}\alpha_n\Bigg).\end{aligned}$$

Let T denote the transaction-confirmation time, the time interval from the arrival time point of a transaction to its departure one. From Little's theorem, the transaction-confirmation time is given by

$$\begin{aligned}E[T] &= \frac{E[N]}{\lambda} \\ &= \frac{1}{2\lambda(b-\lambda E[S])}\Bigg(\lambda^2 E[S^2] - b(b-1) - 2(b-\lambda E[S])^2 \\ &\quad + \sum_{n=0}^{b-1}\Big\{\lambda E[S^2](b-n) + E[S]\{b(b-1) - n(n-1)\} + 2bE[S](b-n)\Big\}\alpha_n\Bigg) \\ &\equiv f(\lambda).\end{aligned} \tag{2.17}$$

2.3. Priority Queue Analysis

We consider the same queueing model as the basic one except the transaction arrival process. Suppose that transactions are classified into c priority classes. For $1 \le i, j \le c$, i class transactions have priority over transactions of class j when $i \le j$. Let λ_i $(i = 1, 2, \ldots, c)$ denote the arrival rate of i-class transactions. We assume that $\sum_{i=1}^{c} \lambda_i E[S] < b$. We define T_i as the sojourn time of class i transactions. For simplicity, we introduce the following notation

$$\bar{\lambda_i} = \sum_{k=1}^{i} \lambda_k, \qquad i = 2, 3, \ldots, c. \tag{2.18}$$

When the system is busy (resp. idle), an arriving transaction must wait for the remaining block-generation time (resp. vacation period) regardless its priority. Note that the vacation period follows the same distribution as the block generation time. Therefore, this priority queueing system with batch service is work conserving. We obtain

$$f(\bar{\lambda_c}) = \sum_{k=1}^{c} \frac{\lambda_k}{\bar{\lambda_c}} E[T_k], \tag{2.19}$$

where $E[T_k]$ is the sojourn time of class k transactions. Since class 1 transactions are served similarity to the batch service analyzed in the previous section, $E[T_1]$ is given by

$$E[T_1] = f(\lambda_1). \tag{2.20}$$

Note that for $i < j$, any class-j transactions don't affect the service of class-i transactions. In other words, T_i is independent of transactions whose priority class is lower than i, and hence (2.19) holds not only c but also $i = 2, 3, \ldots, c-1$. This yields

$$\begin{aligned} f(\bar{\lambda_i}) &= \sum_{k=1}^{i} \frac{\lambda_k}{\bar{\lambda_i}} E[T_k] \\ &= \sum_{k=1}^{i-1} \frac{\lambda_k}{\bar{\lambda_i}} E[T_k] + \frac{\lambda_i}{\bar{\lambda_i}} E[T_i]. \end{aligned}$$

We then obtain

$$E[T_i] = \frac{1}{\lambda_i}\Big(\bar{\lambda_i} f(\bar{\lambda_i}) - \sum_{k=1}^{i-1} \lambda_k E[T_k]\Big), \quad i = 2, 3, \ldots, c. \tag{2.21}$$

$E[T_i]$'s can be calculated recursively by (2.20) and (2.21).

In the following section of numerical examples, we consider two priority-class case: high and low. Let λ_H and λ_L denote the arrival rate of high-priority transactions and that of low-priority ones, respectively. Let also T_H and T_L denote the sojourn time of high-priority transactions and that of low-priority ones, respectively. In this two priority-class case, we obtain

$$\begin{aligned} E[T_H] &= f(\lambda_H), \\ E[T_L] &= \Big(\frac{\lambda_H}{\lambda_L} + 1\Big) f(\lambda_H + \lambda_L) - \frac{\lambda_H}{\lambda_L} f(\lambda_H). \end{aligned}$$

Remark 1 *As we noted in Introduction, the analytical results in [10] were approximately derived due to the lack of work conserving property. Though the resulting equations in this subsection are the same as [10], those hold in a rigorous sense for the queueing model in the present paper.*

2.4. Numerical Examples

2.4.1 Basic parameter setting

It is reported in [10] that the distribution of the block-generation time $G(x)$ is an exponential one given by

$$G(x) = 1 - e^{-\mu x},$$

where $\mu = 0.0018378995$. Then, $E[S]$ and $E[S^2]$ are given by

$$E[S] = \frac{1}{\mu} = 544.0993884, \qquad E[S^2] = \frac{2}{\mu^2} = 592088.2889,$$

The LST of $G(x)$ is given by

$$G^*(s) = \frac{\mu}{s + \mu}.$$

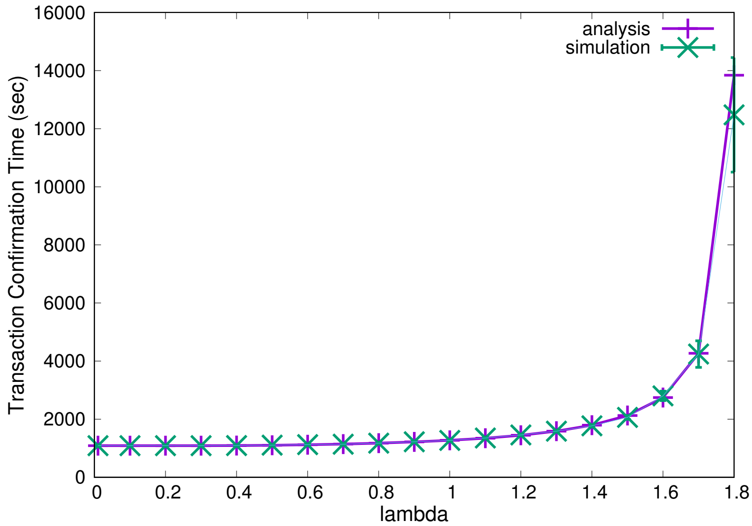

Figure 2.1. Comparison of analysis and simulation for basic queueing model.

With these settings, we calculate the mean transaction-confirmation time $E[T]$.

In order to calculate mean confirmation times of high- and low-priority transactions, we define η as the ratio of the transaction-arrival rate for high-priority class to that for low-priority one. We analyzed mean arrival rates of λ_H and λ_L from the data of October 2013 to September 2015 in [3]. According to the result, we set

$$\eta = \frac{\lambda_H}{\lambda_L} = \frac{0.9349109906}{0.0360010622}.$$

With η, λ_H and λ_L are expressed as

$$\lambda_H = \frac{\eta\lambda}{1+\eta}, \quad \lambda_L = \frac{\lambda}{1+\eta}.$$

By using these relations, we can calculate mean transaction-confirmation times of high and low priority classes, keeping the ratio of λ_H to λ_L constant.

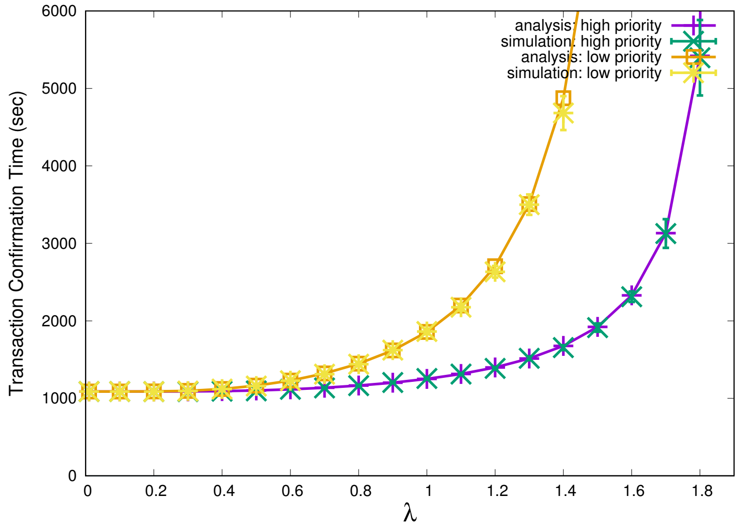

Figure 2.2. Comparison of analysis and simulation for priority-queueing model.

2.4.2 Model validation

Comparison of analysis and simulation

In order to confirm the validity of our analysis, we first conduct the Monte-Carlo simulation with the same basic queueing model in Section 2.1. Figure 2.1 shows the comparison of analysis and simulation for the transaction-confirmation time. In this figure, the horizontal axis represents the transaction arrival rate λ and the vertical one is the mean transaction-confirmation time $E[T]$. The block size is fixed at $b = 1000$ in the numerical simulation. It is shown from Figure 2.1 that the analytical result is the same as simulation, confirming the validity of analysis for the basic queueing model.

In Figure 2.2, we consider the queueing model with two priority classes. Here, the horizontal axis represents the transaction arrival rate λ, which is the sum of transaction arrival rates of high- and low-priority transactions, and the vertical one is the mean transaction-confirmation time $E[T]$. The block size is also fixed at $b = 1000$. From Figure 2.2, the analytical results of the two priority cases agree with the simulation, confirming the validity of analysis.

Table 2.1. Comparison of analysis and measurement.

Transaction Type	Arrival Rate	Measurement [s]	Analysis[s]
Classless	0.9709120529	1124.13286	1112.025339
High Priority	0.9349109906	1059.06133	1108.773595
Low Priority	0.0360010622	3888.06977	1196.469817

Comparison of analysis and measurement

Next, we compare the analysis and measurement. In [10], the authors analyzed two-year transaction data obtained from [3], reporting statistics such as the block-generation time, number of transactions in a block, and transaction-confirmation time. From [10], the mean transaction size is 571.34 bytes, and we set the maximum block size b equal to 1750.

Table 2.1 shows mean transaction-confirmation times of analysis and measurement. The analytical result is calculated with the mean transaction-arrival rate equal to 0.97091, which is obtained from the measured data. We observe in this table that the analytical result is almost the same as the measurement value with relative error of 0.43% in the classless case. This result confirms that the analytical model in the present paper captures the behavior of transactions more accurately than the model in [10], in which the mean transaction-confirmation time in classless case is calculated as 568.10[s].

In terms of priority, the measurement result of high-priority class transactions is slightly larger than the analytical one, while we observe a large discrepancy between measurement and analytical results for low-priority transactions. This result implies that the priority queueing model considered in the paper does not capture the real behavior of Bitcoin transactions. A possible conjecture of this discrepancy is that transactions with low priority are likely to be excluded from the block generation process.

Comparison of analysis and trace-driven simulation: classless case

We conduct trace-driven simulation experiments for further validating our analytical model. From the two-year data of October 2013 to September 2015 in [3],

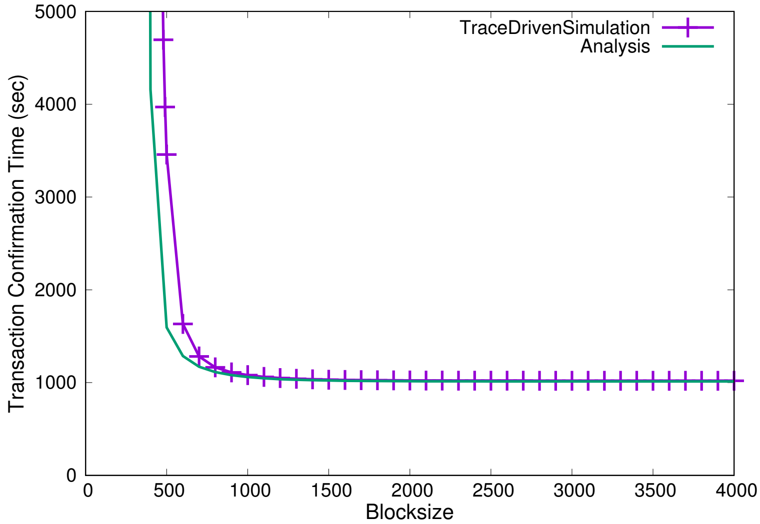

Figure 2.3. Transaction-confirmation time vs. block size, 1st period (October 2013 to September 2014). $\lambda = 0.7336929$, $\mu = 0.0019748858$, and the coefficient of variation of transaction inter-arrival time: 3.72401599.

we collected the sequences of the transaction-arrival time and block-generation one. We used the two sequences such that transactions arrive at the system according to the arrival-time sequence and blocks are generated according to the block-generation-time sequence. We investigate how the block size affects the transaction-confirmation time.

Figures 2.3 and 2.4 represent the transaction-confirmation time against the block size. In Figure 2.3, we use the trace data measured from October 2013 to September 2014 (the 1st period), while the simulation result of Figure 2.4 is based on the trace data measured from October 2014 to September 2015 (the 2nd period). Figure 2.3 shows a good agreement of analysis and simulation, while we observe in Figure 2.4 a large discrepancy between analysis and simulation when the block size is small.

In order to clarify the reason of these discrepancies, we investigate how the transaction-arrival process evolves over time. Figure 2.5 shows the mean transaction-arrival rate per day. In this figure, we observe little variation during the first 12

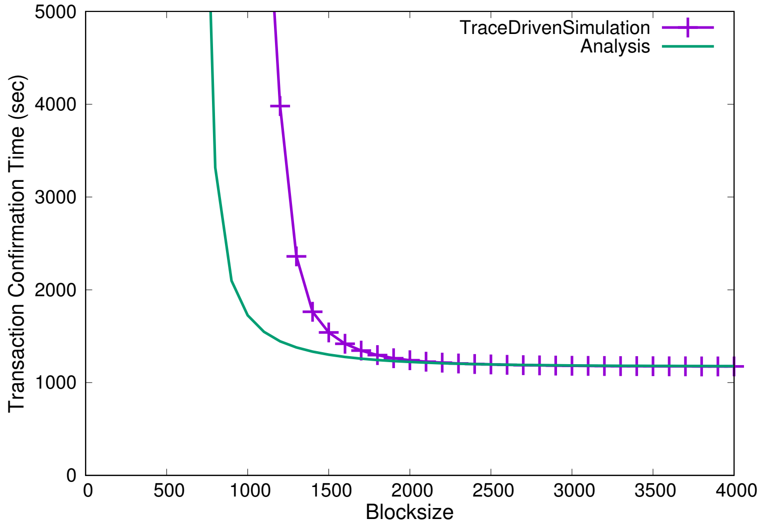

Figure 2.4. Transaction-confirmation time vs. block size, 2nd period (October 2014 to September 2015). $\lambda = 1.2081311$, $\mu = 0.0017009449$, and the coefficient of variation of transaction inter-arrival time: 15.3250509.

Table 2.2. Coefficients of variation for transaction interarrival time.

Period	1st 2013/10-2014/09	2nd 2014/10-2015/09	Overall 2013/10-2015/09
Classless	3.72401	15.32505	10.17893

months, while the mean transaction-arrival rate significantly varies for the last three months in the measured period.

Table 2.2 shows coefficients of variation for transaction interarrival times in the three measurement periods: the 1st period, 2nd period, and the overall period (October 2013 - September 2015). In this table, the coefficient of variation of the 2nd period is larger than that of the 1st period. This large coefficient of variation of the 2nd period results in a large coefficient of variation of the overall period, causing the discrepancy between analysis and simulation.

When the block size is large, there is enough space to include transactions in the next block, and hence burst transaction arrivals are likely to be served in the

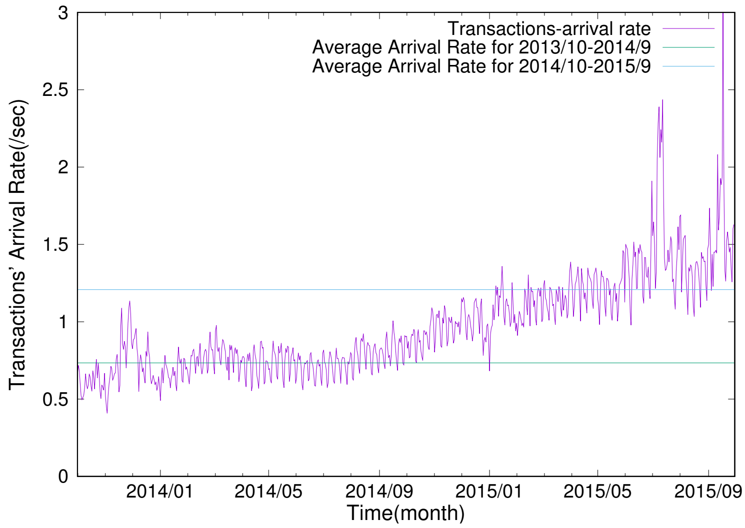

Figure 2.5. The mean transaction-arrival rate.

next block. This causes little difference between analysis and simulation. On the other hand, when the block size is small, the system is likely to be congested due to the bursty nature of the transaction-arrival process. This results in a larger transaction-confirmation time for simulation than that for analysis.

Comparison of analysis and trace-driven simulation: two-priority case

In this subsection, we consider the validation of the two-priority queueing model. Figures 2.6 and 2.7 show the transaction-confirmation time of high priority transaction against the block size. In Figure 2.6, we use the trace data of the 1st period (October 2013 - September 2014), while the simulation result of Figure 2.7 is based on the trace data of the 2nd period (October 2014 - September 2015). Figure 2.6 shows a good agreement of analysis and simulation, while we observe in Figure 2.7 a discrepancy similar to the Figure 2.4.

Figures 2.8 and 2.9 show the transaction-confirmation time of low priority transaction against the block size. In Figure 2.8, we use the trace data of 1st period, while the simulation result of Figure 2.9 is based on the trace data of the

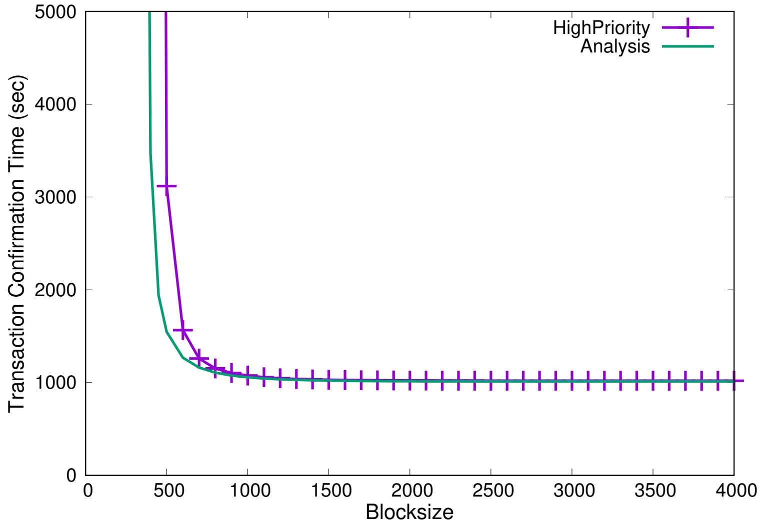

Figure 2.6. Transaction-confirmation time vs. block size, October 2013 to September 2014. $\lambda_H = 0.7203544$, $\mu = 0.0019748858$.

2nd period. From Figure 2.8, the analysis agrees well with simulation, while we observe in Figure 2.9 a large discrepancy between analysis and simulation.

In terms of the coefficient of variation for the interarrival time, the value of the 2nd period is greater than that of the 1st period in both high- and low-priority transaction cases. (See Table 2.3.)

The above observations suggest that the priority queueing model works well when the variation of the transaction-interarrival time is small and the maximum block size is large. When the transaction-interarrival time varies greatly or when the maximum block size is small, however, the priority queueing model does not work well for predicting the mean transaction-confirmation time.

As we discussed in subsection 2.4.2, our priority queueing model does not capture the real behavior of Bitcoin transactions, in particularly for low-priority transactions. Further study of the priority mechanism of Bitcoin is needed and this is our future work.

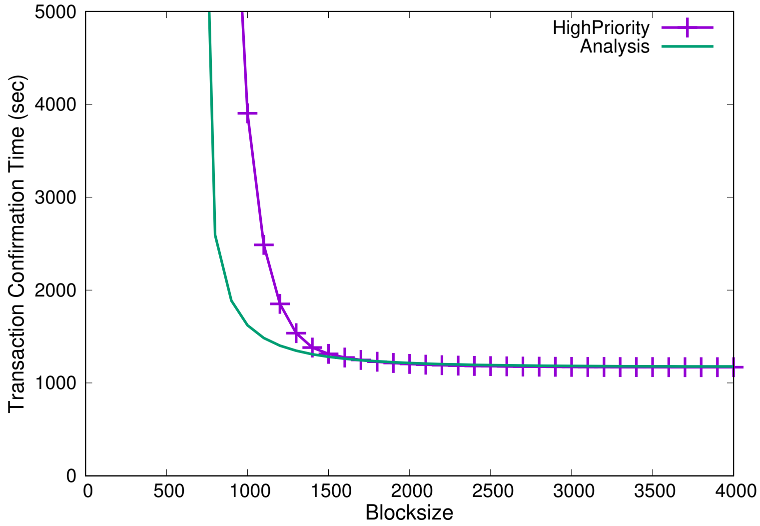

Figure 2.7. Transaction-confirmation time vs. block size, October 2014 to September 2015. $\lambda_H = 1.1494675$, $\mu = 0.0017009449$.

2.4.3 Impact of the block size on the transaction-confirmation time

In this section, we investigate the effect of the block size on the transaction-confirmation time.

Figure 2.10 shows the analytical results with block size $b = 1000$, 2000, 4000, and 8000. We observe that the transaction-confirmation time grows with the increase in the arrival rate. From [10], the maximum number of transactions included in the current maximum block size of 1 Mbyte is approximately given by $b = 1750$. This value is close to $b = 2000$, diverging around $\lambda = 3.6$.

We also observe that enlarging the block size results in a small transaction-confirmation time. However, the transaction-confirmation time for $b = 8000$ rapidly increases when λ is greater than 13 transactions per second. This implies that enlarging the block size does not solve the scalability issue fundamentally.

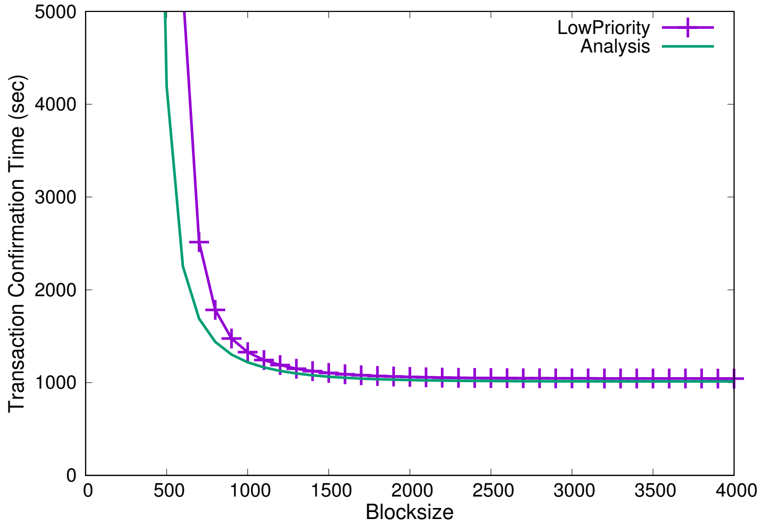

Figure 2.8. Transaction-confirmation time vs. block size, October 2013 to September 2014. $\lambda_L = 0.0133385$, $\mu = 0.0019748858$.

2.4.4 Impact of micropayments: two-priority case

In this subsection, we consider how the mean transaction-confirmation time is affected by priority mechanism of Bitcoin. We consider two scenarios in terms of the growth of transaction-arrival rates. The first scenario is the growth of micropayments, in which we investigate how the increase in the arrival rate of low-priority transactions affects the mean transaction-confirmation time. In the second scenario, we consider the growth of Bitcoin-user population. We increase the overall transaction-arrival rate, keeping the ratio of the arrival rate of low-priority transactions to that of high-priority ones.

Impact of increase in the low priority transaction

Figure 2.11 shows mean transaction-confirmation times of two-priority transactions. We calculate the mean transaction-confirmation time for each priority class by analytical results for the block size b =1000, 2000, 4000, and 8000. In this figure, the transaction-arrival rate of the high priority (λ_H) is fixed at 0.96292,

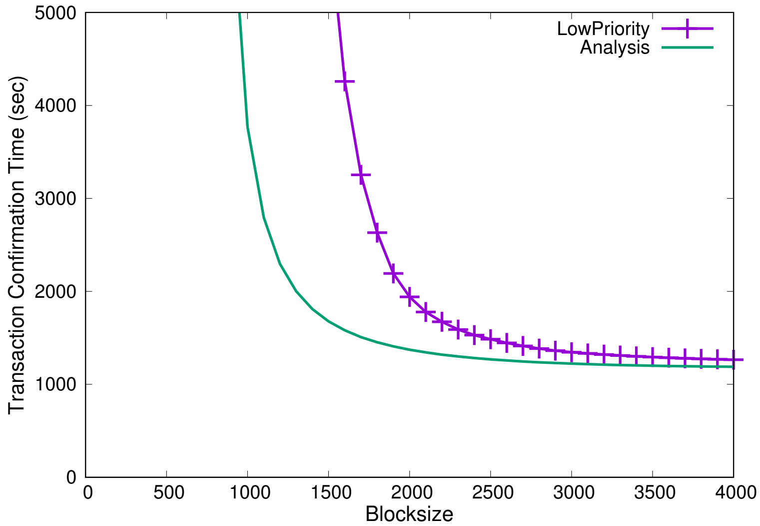

Figure 2.9. Transaction-confirmation time vs. block size, October 2014 to September 2015. $\lambda_L = 0.0586636$, $\mu = 0.0017009449$.

and that of the low priority (λ_L) is changed.

In Figure 2.11, the confirmation time for high-priority transactions is constant in each block-size case. This implies that the transaction-confirmation time for high-priority class is insensitive to the increase in the arrival rate of low-priority transactions. For $b = 2000$, the transaction-confirmation time for low-priority class rapidly increases when $\lambda_L = 2.0$. In case of $b = 4000$, which is almost twice larger than the current maximum block size, the transaction-confirmation time exponentially grows around $\lambda_L = 6.0$. We also observe the same exponential growth for case of $b = 8000$. These results suggest that enhancing the maximum block size does not improve the transaction-confirmation time for low-priority class.

Impact of population growth in Bitcoin

Figures 2.12 and 2.13 show mean transaction-confirmation times of high-priority class and low-priority one, respectively. In both figures, we observe the rapid

Table 2.3. Coefficients of variation for transaction interarrival time, two-priority classes.

Period	1st 2013/10-2014/09	2nd 2014/10-2015/09	Overall 2013/10-2015/09
Low Priority	1.66569	3.90654	3.01387
High Priority	3.70045	14.94895	9.99103

increase in the transaction-confirmation time for any b. For example, when the block size is $b = 2000$, the transaction-confirmation time of high-priority class rapidly increases at around $\lambda_H = 3.5$ in Figure 2.12, and that with low priority grows at around $\lambda_L = 3.0$ in Figure 2.13. Similarly, when $b = 4000$, the confirmation-time of high-priority class diverges at around $\lambda_H = 7.0$ in Figure 2.12, and that of low-priority class rapidly increases at around $\lambda_L = 5.5$ in Figure 2.13. These results suggest that enhancing the maximum block size is effective to mitigate the rapid growth of the transaction-confirmation time. However, its improvement is not enough to solve the scalability issue of Bitcoin.

2.5. Summary

In this chapter, we analyzed the transaction-confirmation time for Bitcoin using a single-server queue model with batch service $M/G^b/1$ and priority service discipline. In this queueing model, newly arriving transactions are temporarily stored in the queue first even when the number of transactions in the server is smaller than the batch size. Assuming that the priority of a transaction depends only on its remittance amount, we derived the mean transaction-confirmation time of each priority class. We validated the analytical results by simulation experiments, and evaluated effects of the block size and transaction-arrival rate on the transaction-confirmation time. We found that the transaction-confirmation time can be decreased by changing the maximum block size. However, its improvement is not effective enough to increase the number of transactions processed per unit time.

The numerical results also indicated that the inaccuracy of the model comes

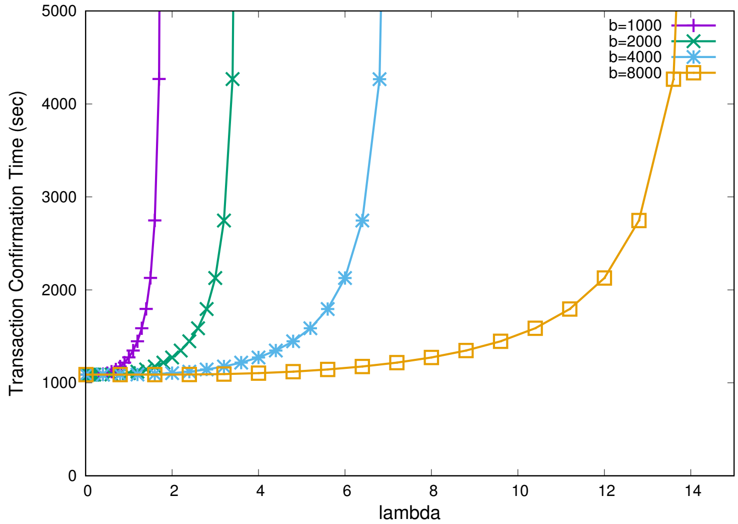

Figure 2.10. The effects of the block size on the transaction-confirmation time.

from the assumption of the Poisson arrival process. In next chapter, we will develop a queueing model with a general process, which captures bursty nature of real transaction-arrival process.

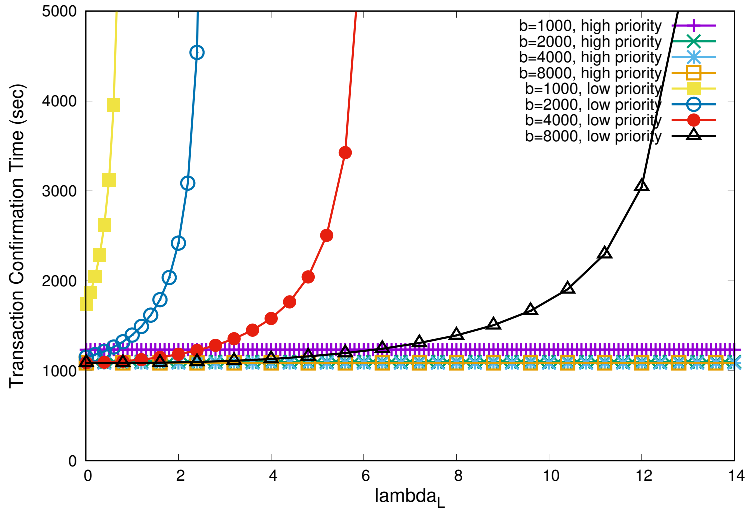

Figure 2.11. The effects of the micropayments on the transaction-confirmation time.

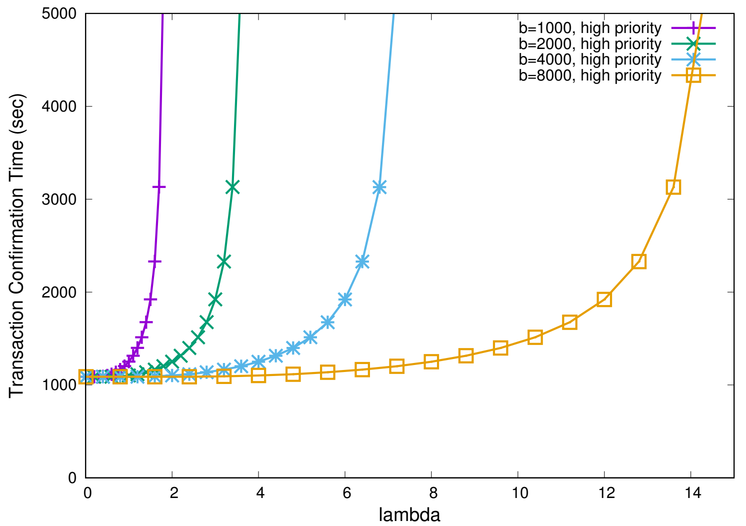

Figure 2.12. The effect of the block size on the transaction-confirmation time of high-priority transactions.

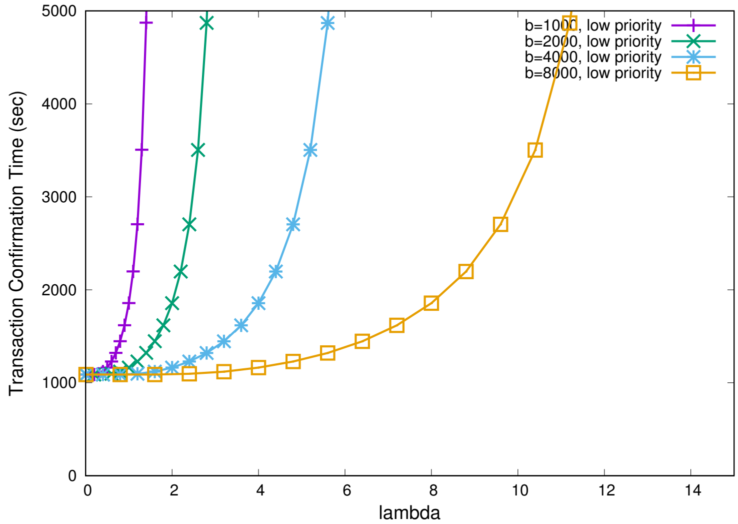

Figure 2.13. The effects of the block size on the transaction-confirmation time of low-priority transactions.

Chapter 3

Analysis of Transaction-Confirmation Time with GI/M^b/1 Queueing Model

In [10], it is reported that the block-generation time, which is equivalent to service time, can be approximated as exponential distribution. It is, however, known that the variation of transaction arrival interval is larger than that of exponential distribution in real Bitcoin system. In this chapter, we assume that the transaction arrival intervals are independent and identically distributed (i.i.d.), and follow a general distribution. The resulting queueing model is GI/M^b/1. The system state is defined as the number of transactions in queue just before a transaction arrival point, and we derive the steady-state distribution and the mean transaction-confirmation time by matrix analytic method. In numerical examples, we compare the result of the theoretical analysis, whose parameter is based on real data, with the result of simulation in order to verify our model.

3.1. GI/M^b/1 Queueing Model

We model the bitcoin system as GI/M^b/1 queue. Transaction interarrival times, $\{A_i : i = 1, 2, \ldots\}$, are independent and identically distributed (i.i.d.), and follow a general distribution $G(x)$. The mean interarrival time is given by

$$E[A] = \int_0^\infty x dG(x) \equiv \frac{1}{\lambda}.$$

Transactions arriving at the system are grouped into a block. The block is confirmed by mining, and we call the block-generation time as the service time hereafter. In [10], it is reported that the block-generation times are independent and identically distributed (i.i.d.), and follow an exponential distribution. We assume the block-generation time S follows an exponential distribution with rate μ, that is,

$$F(x) = \Pr\{S \le x\} = 1 - e^{-\mu x}.$$

Transactions are served in a batch-service manner, and we define the maximum batch size as b.

It is reported in [10] that in Bitcoin system, an arrived transaction is not included in the current mining block, and tends to be included in the next or later block. When the system is busy at a transaction arrival point, the transaction waits in the queue and is included in the next or later blocks in arrival order. Even when there is no transaction in the system, the mining service is continued. We assume that a newly arriving transaction waits in queue until the next service begins.

3.2. Analysis

Let X_n ($n = 1, 2, \ldots$) denote the number of transactions in the mempool, i.e. queue, just before the nth transaction arrival. Let Y_n denote the number of mining completions from the nth and $n+1$st transaction arrival points. We define $D_k^{(n)}$ as the number of transactions simultaneously leaving at the kth mining completion point from the nth transaction arrival epoch.

Noting that the maximum block size is b and that $D_k^{(n)} = b$ ($k = 1, \ldots, Y_n - 1$) for $Y_n > 1$, X_n can be represented as

$$X_{n+1} = \max\left\{X_n - (Y_n - 1)b - D_{Y_n}^{(n)} + 1, 0\right\}, \quad n = 1, 2, \ldots. \tag{3.1}$$

Mining is conducted regardless of the number of transactions in the system, and mining intervals are i.i.d. and follows an exponential distribution. This implies that $\{Y_n : n = 1, 2, \ldots\}$ are i.i.d. and follows a Poisson distribution. Then, $\{X_n : n = 1, 2, \ldots\}$ satisfying (3.1) is a discrete-time Markov chain whose state-transition probability is given by

$$p_{i,j} = P(X_{n+1} = j | X_n = i) = \begin{cases} a_{(j-i-1)/b}, & \text{if } (j-i-1)/b \text{ is an integer,} \\ \overline{a}_{\lfloor i/b \rfloor + 1}, & j = 0, \\ 0, & \text{otherwise,} \end{cases}$$

where

$$\begin{aligned} a_k &= \int_0^\infty e^{-\mu x} \frac{(\mu x)^k}{k!} dG(x), \\ \overline{a}_k &= \sum_{i=k}^\infty a_i. \end{aligned} \tag{3.2}$$

Here, $\lfloor x \rfloor$ means the maximum integer smaller than or equal to x.

The transition-probability matrix $P = (p_{i,j})$ is then given by

$$\boldsymbol{P} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ b-1 \\ b \\ b+1 \\ b+2 \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccccccc} 0 & 1 & 2 & 3 & \cdots & b & b+1 & b+2 & \cdots \end{array} \\ \left(\begin{array}{ccccccccc} \bar{a}_1 & a_0 & 0 & 0 & \cdots & & & & \\ \bar{a}_1 & 0 & a_0 & 0 & \ddots & \ddots & & O & \\ \bar{a}_1 & 0 & 0 & a_0 & \ddots & \ddots & \ddots & & \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ \bar{a}_1 & 0 & 0 & 0 & \ddots & a_0 & 0 & 0 & \ddots \\ \bar{a}_2 & a_1 & 0 & 0 & \ddots & 0 & a_0 & 0 & \ddots \\ \bar{a}_2 & 0 & a_1 & 0 & \ddots & 0 & 0 & a_0 & \ddots \\ \bar{a}_2 & 0 & 0 & a_1 & \ddots & 0 & 0 & 0 & \ddots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{array} \right) \end{array}.$$

From the above expression, it is easy to see that $\{X_n : n = 1, 2, \ldots\}$ is irreducible and aperiodic. If $\lambda/\mu < b$ holds, then the system is stable and the steady-state distribution

$$\pi_k = \lim_{n \to \infty} \Pr\{X_n = k\}, \quad k = 0, 1, \ldots,$$

exists. Let $\boldsymbol{\pi} = (\pi_k)$ denote the probability vector. Then, $\boldsymbol{\pi}$ satisfies the following system of equations.

$$\boldsymbol{\pi} = \boldsymbol{\pi} \boldsymbol{P}, \quad \boldsymbol{\pi} \mathbf{1} = 1,$$

where $\mathbf{1} = (1, 1, \ldots)^\top$ ($\top$ means transpose).

$\boldsymbol{P}$ can be rewritten as the following block matrix representation

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{B}_0 & \boldsymbol{C}_0 & O & O & \cdots \\ \boldsymbol{B}_1 & \boldsymbol{A}_1 & \boldsymbol{A}_0 & O & \cdots \\ \boldsymbol{B}_2 & \boldsymbol{A}_2 & \boldsymbol{A}_1 & \boldsymbol{A}_0 & \cdots \\ \boldsymbol{B}_3 & \boldsymbol{A}_3 & \boldsymbol{A}_2 & \boldsymbol{A}_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

where $\boldsymbol{C}_0$ is a $1 \times b$ matrix, $\boldsymbol{B}_0$ a 1×1 matrix, $\boldsymbol{B}_k$ $(k = 1, 2, \ldots)$ a $b \times 1$ matrix, and A_l $(l = 0, 1, \ldots)$ a $b \times b$ matrix, and those are given by

$$\boldsymbol{C}_0 = (a_0, 0, \ldots, 0), \quad \boldsymbol{B}_0 = (\overline{a}_1), \quad \boldsymbol{B}_k = (\overline{a}_k, \ldots, \overline{a}_k, \overline{a}_{k+1}),$$

$$\boldsymbol{A}_l = \begin{pmatrix} 0 & a_l & & & & \\ 0 & 0 & a_l & & O & \\ 0 & 0 & 0 & & & \\ \vdots & & & \ddots & & \\ 0 & 0 & 0 & & 0 & a_l \\ a_{l+1} & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}.$$

We define $\boldsymbol{\pi}_0$ and $\boldsymbol{\pi}_n$ as follows.

$$\boldsymbol{\pi}_0 = (\pi_0), \quad \boldsymbol{\pi}_n = (\pi_{(n-1)b+1}, \pi_{(n-1)b+2}, \ldots, \pi_{nb}), \ n = 1, 2, \ldots,$$

Then, the stationary distribution vector $\boldsymbol{\pi}$ can be represented as $\boldsymbol{\pi} = (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots)$. From the structure of the transition probability matrix for GI/M/1-type Markov chain,

$$\boldsymbol{\pi}_{n+1} = \boldsymbol{\pi}_1 \boldsymbol{R}^n, \qquad n = 0, 1, \ldots,$$

holds [17]. $\boldsymbol{R}$ can be iteratively computed by the following matrix recursive equation of $\boldsymbol{R}(n)$

$$\boldsymbol{R}(n) = \left(\boldsymbol{A}_0 + \sum_{k=2}^{\infty} [\boldsymbol{R}(n-1)]^k \boldsymbol{A}_k \right) (\boldsymbol{I} - \boldsymbol{A}_1)^{-1}.$$

$\boldsymbol{\pi}_0$ and $\boldsymbol{\pi}_1$ can be obtained by solving the following equations.

$$\begin{aligned}
\boldsymbol{\pi}_0 &= \boldsymbol{\pi}_0 \boldsymbol{B}_0 + \boldsymbol{\pi}_1 (\sum_{l=1}^{\infty} \boldsymbol{R}^{l-1} \boldsymbol{B}_l), \\
\boldsymbol{\pi}_1 &= \boldsymbol{\pi}_0 \boldsymbol{C}_0 + \boldsymbol{\pi}_1 (\sum_{l=1}^{\infty} \boldsymbol{R}^{l-1} \boldsymbol{A}_l),
\end{aligned}$$

$$\boldsymbol{\pi}_0 \mathbf{1}^{\top} + \boldsymbol{\pi}_1 (\boldsymbol{I} - \boldsymbol{R})^{-1} \mathbf{1}^{\top} = 1.$$

The mean transaction waiting time $E[W]$ and mean transaction-sojourn time $E[T]$ can be yielded as

$$\begin{aligned}
E[W] &= \sum_{k=0}^{\infty} \pi_k \left(\left\lfloor \frac{k}{b} \right\rfloor + 1 \right) E[S], \\
E[T] &= E[W] + E[S].
\end{aligned}$$

We call $E[T]$ as the mean transaction-confirmation time hereafter.

3.3. Numerical Examples

3.3.1 Estimation of transaction interarrival-time distribution

As we showed in the previous chapter, the coefficient of variation of the transaction interarrival time is significantly larger than that of exponential distribution. Here, we assume that the transaction interarrival time follows a hyper-exponential distribution. We estimate the parameter values of the hyper-exponential distribution by EM algorithm [18].

The hyper exponential distribution is a combination of phases each of which is an exponential distribution independent of the other phases. We consider the following hyper-exponential distribution with m phases

$$G(x) = 1 - \sum_{i=1}^{m} c_i e^{-\lambda_i x}, \qquad x \geq 0,$$

where $\sum_{i=1}^{m} c_i = 1$. The probability density function $g(x)$ is given by

$$g(x) = \sum_{i=1}^{m} c_i \lambda_i e^{-\lambda_i x}, \qquad x \geq 0.$$

Table 3.1. Estimation result of EM algorithm. (2013/10/1-2014/9/30)

i	1	2	3	4	5
c_i	0.446648	0.327843	0.126944	0.098488	7.41350 $\times 10^{-5}$
λ_i	0.866223	0.866223	0.866223	0.331555	0.00293430

Table 3.2. Estimation result of EM algorithm. (2014/10/1-2015/9/30)

i	1	2	3	4	5
c_i	0.566151	0.259492	0.161661	0.012449	2.45808 $\times 10^{-4}$
λ_i	1.119559	1085.472	1.119559	0.326394	0.001765193

In order to estimate the parameters of $G(x)$, we use the two-year data of [3] measured from October 1st 2013 to September 30th 2015, and retrieve the sequence of the transaction-arrival time. We divide the two-year sequence of the transaction-arrival time into two one-year sequences, the first year of October 1st 2013 to September 30th 2014, and the second year of October 1st 2014 to September 30th 2015. We apply the EM algorithm of [18] to the two data sets under the setting of the number of phases $m = 5$ and convergence criterion $\epsilon = 10^{-10}$.

Tables 3.1 and 3.2 show the parameter-estimation results from the first-year data and the second-year one, respectively. Table 3.3 (resp. Table 3.4) represents the statistics of transaction interarrival time obtained from measured data (resp. the hyper-exponential distribution estimated by EM algorithm). In Tables 3.3 and 3.4, the average of the hyper exponential distribution estimated by the EM algorithm is the same as that of the actual data, and standard deviation and coefficient of variation estimated by the EM algorithm are almost the same as that of the actual data. Comparing the statistics of the first-year data with the second-year one in Table3.3, the mean interarrival time of the first-year data is larger than that of the second-year one, while the variation of the first-year data is smaller than that of the second-year one.

Table 3.3. Statistics of transaction interarrival time for actual data.

period	mean	standard deviation	coefficient of variation
2013/10/1-2014/09/30	1.36296	5.07571	3.72401
2014/10/1-2015/09/30	0.82772	12.68489	15.32505

Table 3.4. Statistics of hyper-exponential distribution estimated by EM Algorithm.

period	mean	standard deviation	coefficient of variation
2013/10/1-2014/09/30	1.36296	4.42236	3.24465
2014/10/1-2015/09/30	0.82772	12.58913	15.20936

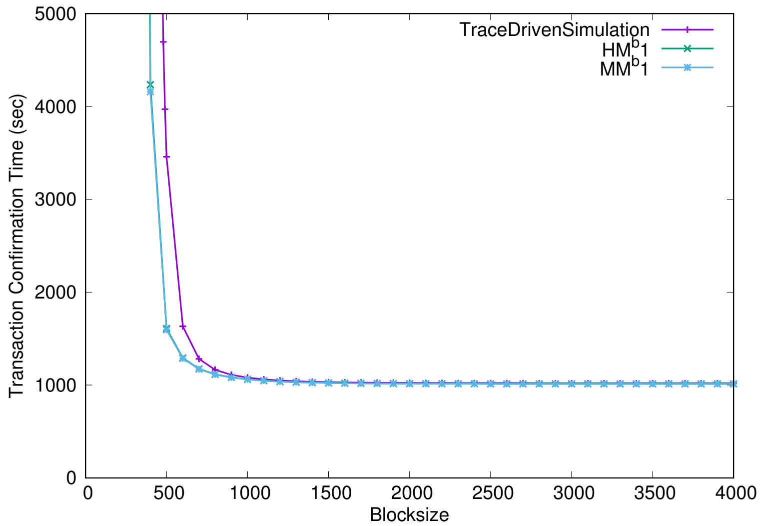

Figure 3.1. Comparison of trace-driven simulation and analysis. （2013/10/1-2014/9/30）

3.3.2 Comparison of trace-driven simulation and analysis

In this section, comparing the analysis and simulation based on trace data for the transaction-arrival time and block-generation time [3], we verify the validity

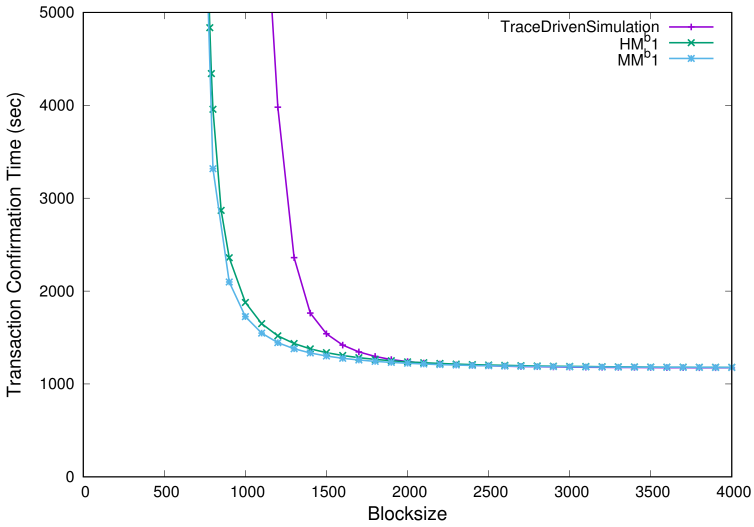

Figure 3.2. Comparison of trace-driven simulation and analysis. （2014/10/1-2015/9/30）

of our analytical model.

Figure 3.1 shows the effect of the block size on the mean transaction-confirmation time. Here, we plot results of analysis and simulation. For the parameters of the hyperexponential distribution in the transaction-arrival process, we use the values in Table 3.1, the estimation results for the first-year data of October 1st 2013 to September 30th 2014. The mean block-generation time is also calculated from the same data. As a reference, the mean transaction-confirmation time of $M/M^b/1$ whose transaction-arrival and service rates are calculated from the first-year data is also shown in the figure.

In Figure 3.1, a discrepancy is observed among analytical results and simulation one when b is small. When b is greater than 700, all the results are the same, implying that the analytical model gives a good approximation. From this figure, it is also observed that the result of $GI/M^b/1$ is almost the same as that of $M/M^b/1$. Note that in Table 3.1, the phase with the transaction-arrival rate 0.866223 occupies about 90% of the first-year period. This implies that the resulting hyper-exponential distribution is similar to an exponential distribution with the mean interarrival time $1/0.866223 = 1.15444$, which is close to 1.36296 of the average of measured data for the first-year period in Table 3.4.

Figure 3.2 shows the results of analysis and simulation for the second-year data. From this figure, we observe a large discrepancy between analysis and simulation when b is smaller than or equal to 1500. Comparing this figure with Figure 3.1, the difference between analysis and simulation in Figure 3.2 is larger than that in Figure 3.1. This is because the variation of the transaction arrival process for the second-year data is larger than that for the first-year one.

From Figure 3.2, it is observed that when the block size is small, the mean transaction-confirmation time of $\mathrm{H}_m/\mathrm{M}^b/1$ is slightly larger than that of $\mathrm{M}/\mathrm{M}^b/1$, but both results are almost the same. From Table 3.2, the hyper-exponential distribution estimated by EM algorithm can approximate the average, standard deviation, and coefficient of variation of the transaction interarrival time with high accuracy. This suggests that it is difficult to improve the accuracy with an i.i.d. hyper-exponential distribution.

3.3.3 Effect of block size and arrival rate on transaction-confirmation time

In this section, we examine the effect of the transaction-arrival rate on the transaction-confirmation time. Let λ denote the mean transaction-arrival rate, which is given by

$$\lambda = \frac{\alpha}{\sum_{i=1}^{m} c_i/\lambda_i}. \tag{3.3}$$

In the following numerical examples, we change α (i.e., λ), keeping c_i and λ_i the same as estimated values.

Figure 3.3 shows the mean transaction-confirmation time when the mean arrival rate increases. Here, we set the average block generation rate as $\mu = 0.0019749$ based on the real data in 2013/10/1 to 2014/09/30, and we use the parameter values of Table 3.1 for the hyper-exponential distribution. Considering block size of $b = 1000$, 2000, and 4000, we calculate the mean transaction-confirmation time for each. For reference, the result of $\mathrm{M}/M^b/1$ is also shown in the same figure.

From Figure 3.3, it is observed that the mean transaction-confirmation time

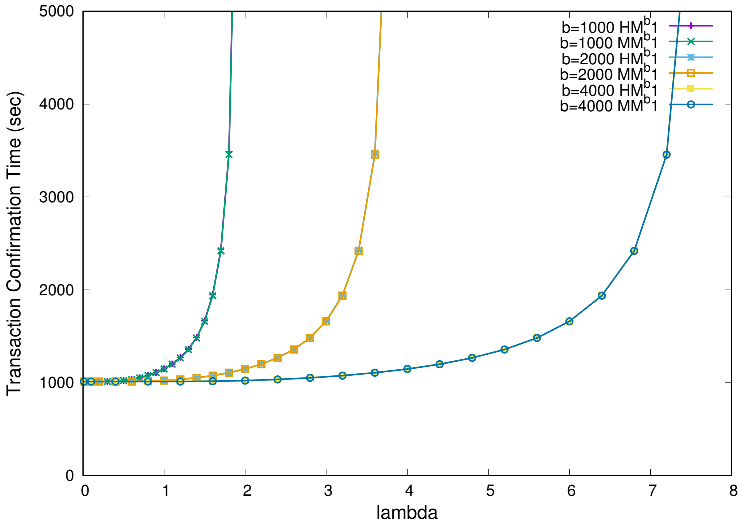

Figure 3.3. The effect of block size and arrival rate on transaction-confirmation time. (2013/10/1-2014/9/30)

increases with the increase in λ, as expected. The mean transaction-confirmation time of $b = 1000$ is the largest, and that of $b = 4000$ gradually increases compared to the other two cases. From [10], it is reported that the mean number of transactions in a block is 1750, and note that $b = 2000$ is close to that value. Even when we set the block size doubled to $b = 4000$, the average transaction-confirmation time rapidly increases over the arrival rate exceeding $\lambda = 7$. That is, increasing the block size does not improve transaction processing significantly.

Figure 3.4 shows the mean transaction-confirmation time against the mean transaction-arrival rate. Here, we use the parameters estimated from the second-year data. The other settings are the same as Figure 3.3.

In Figure 3.4, it is shown that the mean transaction-confirmation time in each block size grows with the increase in λ, similarly to Figure 3.3. Comparing Figure 3.4 with Figure 3.3, the mean transaction-confirmation time in Figure 3.4 increases more rapidly than that in Figure 3.3 for any block size. This is because the variation of transaction arrival process in the second-year period is larger than that in the first-year period. In addition, the mean block-generation time

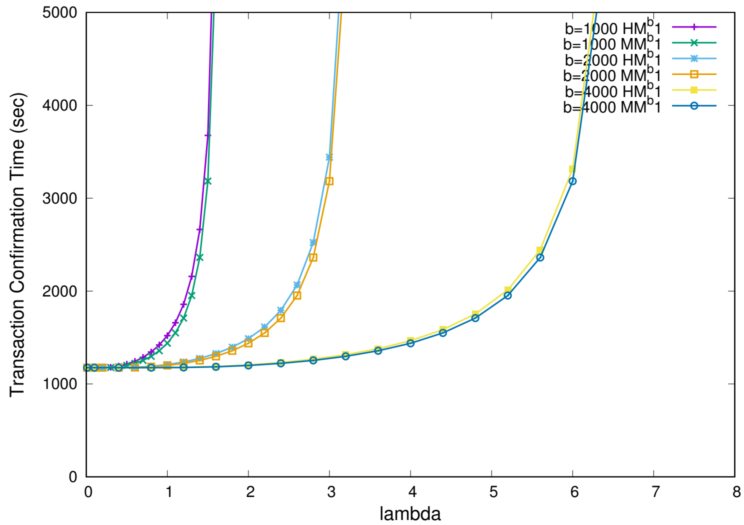

Figure 3.4. The effect of block size and arrival rate on transaction-confirmation time. (2014/10/1-2015/9/30)

for the second-year period is smaller than that in the second-year period.

For both Figures 3.3 and 3.4, it is observed that the result of M/M^b/1 is almost the same as that of H$_m$/M^b/1, suggesting that it is difficult to improve the accuracy of the mean transaction-confirmation time with GI/M^b/1 model.

3.3.4 Effect of block size and coefficient of variation on transaction-confirmation time

In this section, we evaluate the effect of coefficient of variation in the transaction interarrival time on the transaction-confirmation time. Here, we consider the following two-phase hyper-exponential distribution for the transaction-arrival process

$$g(x) = (1-p)\lambda_1 e^{-\lambda_1 x} + p\lambda_2 e^{-\lambda_2 x}.$$

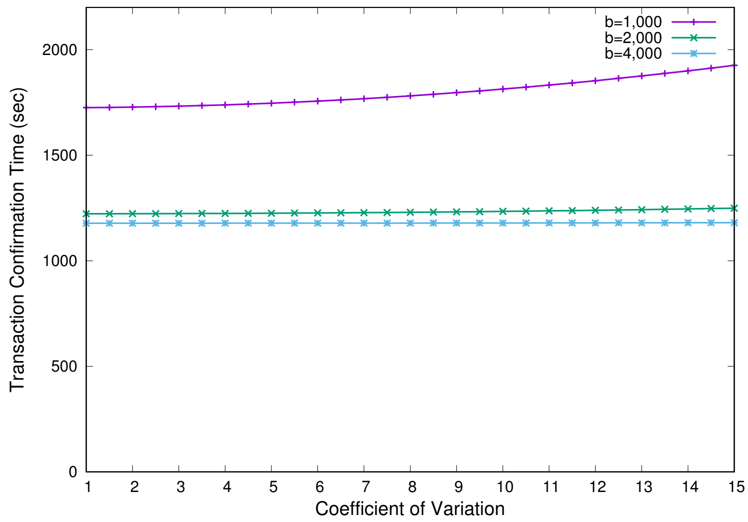

Figure 3.5. The effect of block size and coefficient of variation on transaction-confirmation time. (2014/10/1-2015/9/30)

Let α denote the coefficient of variation of the transaction interarrival time A. Noting that $\alpha = \sqrt{V[A]}/E[A]$, we obtain

$$E[A^2] = (1+\alpha^2)(E[A])^2. \tag{3.4}$$

The first and second moments of A are given by

$$E[A] = \frac{1-p}{\lambda_1} + \frac{p}{\lambda_2}, \tag{3.5}$$

$$E[A^2] = \frac{2(1-p)}{\lambda_1^2} + \frac{2p}{\lambda_2^2}. \tag{3.6}$$

Note that (3.5) is equal to the mean transaction interarrival time $1/\lambda$. Substituting (3.5) and (3.6) into (3.4), we obtain

$$\frac{2(1-p)}{\lambda_1^2} + \frac{2p}{\lambda_2^2} = \frac{1+\alpha^2}{\lambda^2}. \tag{3.7}$$

Considering $\lambda_2 = \gamma\lambda_1$, we obtain the following quadratic equation for p from (3.5) and (3.7)

$$\left(1-\frac{1}{\gamma}\right)^2 p^2 - 2\left(\left(1-\frac{1}{\gamma}\right) - \frac{1}{1+\alpha^2}\left(1-\frac{1}{\gamma^2}\right)\right)p + \frac{\alpha^2-1}{\alpha^2+1} = 0 \tag{3.8}$$

When the coefficient of variation α and the constant γ are given, p $(0 < p < 1)$ can be determined from 3.8.

Figure 3.5 shows the transaction-confirmation time against the coefficient of variation of transaction interarrival time. Here, we use the mean arrival rate $\lambda = 1.20813$ in the second-year period and the service rate $\mu = 0.0017009$. We fixed $\gamma = 1000$ and calculate the mean transaction-confirmation time for three types of block size $b = 1000$, 2000, and 4000.

From Figure 3.5, for b = 1000, the transaction-confirmation time gradually grows with the increase in the coefficient of variation. However, the mean transaction-confirmation times for $b = 2000$ and 4000 remain constant. This is because the block size is large enough to absorb the impact of variation of the transaction arrival process on the transaction-confirmation time.

3.4. Summary

In this chapter, we analyzed the transaction-confirmation time for Bitcoin with GI/M^b/1 queueing model. We defined the number of transactions in the system just before a transaction arrives as the system state, and derived the steady-state distribution and the mean transaction-confirmation time by matrix analytic method. In numerical examples, taking into account a large coefficient of variation of the transaction interarrival time in real Bitcoin system, we assumed that the transaction interarrival time follows a hyper exponential distribution, and we estimated its parameters by the EM algorithm. We confirmed that the first and second moments of the resulting hyper-exponential distribution agree with those of actual data. In terms of the mean transaction-confirmation time, however, we observed a discrepancy between the trace-driven simulation and the analytical result. This suggests that it is difficult to model the transaction interarrival time under i.i.d. assumption for the transaction interarrival time.

Chapter 4

Conclusion and Future Work

4.1. Conclusion

In this study, we analyzed the transaction-confirmation time for Bitcoin. In Chapter 2, we modeled the mining process with an M/G^b/1, a single-server queue with batch service and priority service discipline. In this queueing model, newly arriving transactions are temporarily stored in the queue first even when the number of transactions in the server is smaller than the batch size. Assuming that the priority of a transaction depends only on its remittance amount, we derived the mean transaction-confirmation time of each priority class. We compared the results of analytics to that of simulation in order to validate our model, and evaluate how the block size and transaction-arrival rate affect the transaction-confirmation time. We found that the transaction-confirmation time can be decreased by increasing the maximum block size. However, its improvement is not effective enough to increase the number of transactions processed per unit time.

In Chapter 3, we considered the variation of the transaction interarrival time. Since the block-generation time is known to be approximated with exponential distribution. we analyzed the transaction-confirmation time for Bitcoin with a GI/M^b/1 queueing model. In this queueing model, we defined the number of transactions in the system just before a transaction arrives at the system as the system state, deriving the steady-state distribution and the mean transaction-confirmation time by matrix analytic method. In numerical examples, we assumed the distribution of transaction interarrival times as a hyper-exponential distribution, and estimated its parameters by the EM algorithm. We observed that the first and second moments of the resulting hyper-exponential distribution agree with those of actual data. In terms of the transaction-confirmation time, however, there is a discrepancy between the trace-driven simulation and the result of the analysis especially when the block size is small. This suggests that it is difficult to predict the mean transaction-confirmation time with high accuracy under the assumption that interarrival times are independent and identically distributed.

We evaluated effects of the transaction-arrival rate and block size on the transaction-confirmation time. We found that it is possible to increase the transaction throughput to some extent by enhancing the block size, however, the improvement is not large enough to solve the problem of scalability fundamentally. We also evaluated how the coefficient of variation of the hyper exponential distribution with two phases affects the transaction-confirmation time. From the results, the transaction-confirmation time gradually increases with the coefficient of variation, however, the effect is negligible when the block size is large.

4.2. FutureWork

Recently, one of schemes solving the scalability issue of Bitcoin is lightening network, which provides a channel dedicated to micropayment transactions [16]. The lightening network is expected to mitigate the overloaded block-generation process, however, it is not clear how the lightening network decreases the transaction-confirmation time. For our future work, we will develop analytical models for the lightening network.

As for queueing models considered in this thesis, we extend the GI/M^b/1 to a batch service queuing model with a general arrival process such as Markovian arrival process (MAP) or batch MAP (BMAP). Note that MAP and BMAP can express a correlation nature of transaction-interarrival times. It is expected to predict more accurately the transaction-confirmation time by queueing models with MAP and BMAP inputs.

Acknowledgements

I sincerely thank to Prof. Shoji Kasahara for constructive suggestion in this research, and providing guides and ideas in daily tasks in order to achieve good results. I also thank to Prof. Kazushi Ikeda for reviewing this thesis, and providing valuable comments. I would like to thank to Associate Prof. Masahiro Sasabe for giving advice in this research. I also would like to thank to Assistant Prof. Jun Kawahara for preparing data with very careful, and providing advice. I would like to thank to Ms. Yoko Hashimoto for various support during my laboratory life.

References

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” https://bitcoin.org/bitcoin.pdf, 2008.

[2] http://www.meti.go.jp/committee/kenkyukai/sansei/fintech_kadai/pdf/003_02_00.pdf

[3] https://blockchain.info/

[4] http://www.biccamera.co.jp/shopguide/campaign/bitcoin/index.html

[5] A. M. Antonopoulos, *Mastering Bitcoin*, O’Reilly, 2014.

[6] http://www.coindesk.com/1mb-block-size-today-bitcoin/

[7] http://www.coindesk.com/segregated-witness-bitcoin-block-size-debate/

[8] M. L. Chaudhry and J. G. C. Templeton, “The Queing System $M/G^B/1$ and its Ramifications,” *European Journal of Operational Research*, vol. 6, pp. 56–60, 1981.

[9] M. L. Chaudhry and J. G. C. Templeton, *A First Course in Bulk Queues*, John Wiley & Sons, 1983.

[10] S. Kasahara and J. Kawahara, “Effect of Bitcoin Fee on Transaction-Confirmation Process,” To appear in *Journal of Industrial and Management Optimization*.

[11] Y. Kawase and S. Kasahara, “Transaction-Confirmation Time for Bitcoin: A Queueing Analytical Approach to Blockchain Mechanism,” *Queueing Theory and Network Applications (12th International Conference, QTNA2017, Qinhuangdao, China, August 21–23, 2017, Proceedings)*, W. Yue, Q.-L. Lin, S. Jin, and Z. Ma Eds., LNCS, vol. 10591, pp. 75–88, 2017.

[12] Y. Kawase and S. Kasahara, “Priority Queueing Analysis of Transaction-Confirmation Time for Bitcoin Blockchain,” Submitted for review.

[13] F. Tschorsch and B. Scheuermann, “Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies,” *IEEE Communications Surveys & Tutorials*, vol. 18, pp. 2084–2123, 2016.

[14] M. Möser and R. Böhome, “Trends, Tips, Tolls: A Longitudinal Study of Bitcoin Transaction Fees,” *Financial Cryptography and Data Security , Lecture Notes in Computer Science, Springer*, vol. 8976, pp. 19–33, 2015.

[15] H. Takagi, Queueing Analysis: *A Foundation of Performance Evaluation Vol. 1, Vacation and Priority Systems, Part1*, North-Holland, 1991.

[16] J. Poon, T. Dryja, “The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments,” https://lightning.network/lightning-network-paper.pdf, 2016.

[17] R. Nelson, *Probability, Stochastic Processes, and Queueing Theory*, Springer-Verlag, 1995.

[18] R. E. A. Khayari, R. Sadre and B. R. Haverkort, “Fitting World-Wide Web Request Traces with the EM-Algorithm,” Performance Evaluation, vol. 52, pp. 175–191, 2003.

[19] https://www.coindesk.com/segwits-slow-rollout-bitcoins-capacity-hasnt-seen-sudden-boost/

[20] https://www.bitcoincash.org/

List of Publications

Journal

1. Y. Kawase and S. Kasahara, "Priority Queueing Analysis of Transaction-Confirmation Time for Bitcoin Blockchain," Submitted for review.

International Conferences

1. Y. Kawase and S. Kasahara, "Transaction-Confirmation Time for Bitcoin: A Queueing Analytical Approach to Blockchain Mechanism," *Queueing Theory and Network Applications (12th International Conference, QTNA2017, Qinhuangdao, China, August 21–23, 2017, Proceedings)*, W. Yue, Q.-L. Lin, S. Jin, and Z. Ma Eds., LNCS, vol. 10591, pp. 75–88, 2017.

Domestic Conferences

1. 河瀬良亮, 笠原正治, "ビットコインのマイニング・モデルとトランザクション承認時間解析," 33 回（2016 年度）待ち行列シンポジウム「確率モデルとその応用」, 64–73.

2. 河瀬良亮, 笠原正治, "GI/M^b/1 モデルによるビットコイン・トランザクション承認時間解析," 34 回（2017 年度）待ち行列シンポジウム「確率モデルとその応用」, 109–118.