INVITED PAPER *Special Section on Recent Progress in Networking Science and Practice in Conjunction with Main Topics of ITC32*

Performance Modeling of Bitcoin Blockchain: Mining Mechanism and Transaction-Confirmation Process

Shoji KASAHARA[†a)], ***Fellow***

SUMMARY Bitcoin is one of popular cryptocurrencies widely used over the world, and its blockchain technology has attracted considerable attention. In Bitcoin system, it has been reported that transactions are prioritized according to transaction fees, and that transactions with high priorities are likely to be confirmed faster than those with low priorities. In this paper, we consider performance modeling of Bitcoin-blockchain system in order to characterize the transaction-confirmation time. We first introduce the Bitcoin system, focusing on proof-of-work, the consensus mechanism of Bitcoin blockchain. Then, we show some queueing models and its analytical results, discussing the implications and insights obtained from the queueing models.

key words: *Bitcoin, blockchain, performance modeling, queueing theory, extreme value theory, transaction-confirmation time*

1. Introduction

Bitcoin is a decentralized cryptcurrency system supported by blockchain technology, and it was invented by an unknown person/group called Satoshi Nakamoto in 2008 [21]. There is no central authority to manage the minting and circulation of Bitcoin. A remarkable feature of Bitcoin is the Bitcoin cryptcurrency is managed by open-source software, and the processes of minting and circulating Bitcoin are supported by volunteer nodes called miners. A key technology of Bitcoin is *blockchain*, a distributed ledger maintained by all the miner nodes joining Bitcoin peer-to-peer (P2P) network. Miner nodes hold the same replica of the blockchain through the P2P network, verifying the consistency of transactions issued by end users.

In Bitcoin system, transactions issued by end users are broadcasted over the P2P network and received by miner nodes. Then, each miner node creates a *block* from the transactions and starts solving a puzzle-like mathematical problem based on a cryptographic hash algorithm. When a miner node finds its answer, the miner appends the block to the blockchain and receives the reward called coinbase and transaction fees included in the block. The difficulty of the mathematical problem is adjusted automatically such that the block-generation interval is 10 minutes on average [1]. Transactions registered through blocks in the blockchain are acknowledged as legitimate ones, and hence called *confirmed* transactions.

One of important issues for Bitcoin is scalability. Since the maximum block-size limit is 1 Mbyte and the block-generation time is 10 minutes on average, the number of transactions processed per second (tps) is small and at most seven [25]. In order for Bitcoin to handle tens of thousands of tps, the same order as the processing speed of credit card transactions, Bitcoin developers' community has discussed the increase of the maximum block-size limit, however, no agreement has been reached. The alternative approach to the low-scalability issue of Bitcoin is decreasing the mining difficulty which results in a small interval between two consecutive blocks. However, a short block-generation time induces fork, in which more than one block has the same height because of multiple answer detection by distinct miner nodes, causing serious security issues [2], [5], [8].

In order to make Bitcoin a sustainable ecosystem, incentivisation for miner nodes is indispensable. Since the coinbase is halved every four years, one direction for realizing the Bitcoin ecosystem is designing transaction fees to give incentives for miners to contribute their computation power to maintaining the blockchain. Currently, however, there is no policy/function specified in the Bitcoin protocol. This allows miner nodes to make a benefit-based mining strategy. The authors of [16] analyzed the statistical trends of transaction fees, finding the regime shift of Bitcoin transaction fees. They reported that transactions with fee are likely to be processed faster than those without fee. It was also shown that the transaction latency is not significantly affected by the amount of fee. However, they showed the statistical analysis that transactions with a large amount of fee are likely to be served faster than those with small fee.

In order to consider the issues of scalability and incentive mechanism of Bitcoin, quantitative characterization of the transaction-confirmation process plays an important role. As we will describe in the next section, Bitcoin blockchain processes transactions in block basis. That is, transactions arriving to the Bitcoin system are simultaneously served as a batch manner. From this observation, the transaction-confirmation process can be modeled as a single-server queueing system with batch service.

We studied the queueing models for the Bitcoin blockchain in [9]–[12] for characterizing the transaction-confirmation time of Bitcoin blockchain. In this paper, we give the summary of queueing models in [9]–[12]. Through the research work, a main idea of the analysis is two-fold. First, the transaction-confirmation process is modeled as a queueing system with batch service and priority discipline,

Manuscript received March 8, 2021.
Manuscript revised April 21, 2021.
Manuscript publicized June 9, 2021.

[†] The author is with the Division of Information Science, Graduate School of Science and Technology, Nara Institute of Science and Technology, Ikoma-shi, 630-0192 Japan.
a) E-mail: kasahara@ieee.org

DOI: 10.1587/transcom.2021ITI0003

with which the mean transaction-confirmation time for each prioritized transaction is derived. Second, focusing on the block-generation time, which is corresponding to the service time of the queueing model, we apply the extreme-value theory to the mining process of the proof-of-work consensus algorithm. We also show some numerical results and discuss the implications and insights obtained from the queueing models.

The structure of the paper is as follows. In Sect. 2, we give a brief summary of Bitcoin blockchain and its mining mechanism. Some related work for the blockchain evaluation is also presented. Section 3 shows the fundamental queueing model for the Bitcoin blockchain and its analysis. In Sect. 4, we show an extended queueing model in which the transaction-arrival process is general one. Finally, we conclude the paper in Sect. 5.

2. Summary of Bitcoin Blockchain and Related Work on Performance Modeling

In this section, we briefly give a summary of Bitcoin blockchain, in particular, mining mechanism and the transaction-confirmation process. For more details, the readers are referred to [1], [20]. We also show some related work on performance modeling for Bitcoin blockchain.

2.1 Mining Mechanism and Transaction-Confirmation Process

In the Bitcoin system, virtual currency circulation is realized with two data types: transactions and blocks. A transaction is the value-transfer base of Bitcoin, while a block is a data composed of transactions to be verified.

Suppose user A makes payment to user B. User A generates a transaction including the payment to user B and transaction fee, then issuing it into the Bitcoin P2P network. The transaction is broadcasted through the P2P network, and temporally stored in memory pool of miner nodes.

Each miner node generates a block containing transactions to be validated, and then tries to solve a mathematical problem based on a cryptographic hash algorithm, which is associated with the newly generated block. This process is called *proof-of-work* [21], which makes consensus among all the miner nodes joining the P2P network. The miner who finds its solution first is given the right to add the newly generated block to the blockchain, being awarded reward[†]. Then, the miners restart to solve a new mathematical problem associated with the next block.

A key feature of the blockchain is that the solution found by the winning miner is included in the next block to be generated. The process of finding mathematical solutions is called *mining*. The inclusion of the solution of the current block to the next block makes the Bitcoin blockchain tamper resistant, i.e., preventing Bitcoin from tampering previous blocks. The difficulty of the mathematical problems in mining is automatically adjusted such that the block-generation time is kept to 10 minutes on average.

In the example of user A's payment to user B, user B can use the Bitcoin of the transaction sent by user A after the transaction is confirmed. Note that the transaction is confirmed when a block including the transaction is added to the blockchain. In the following, we define the transaction-confirmation time as the time interval from when a user issues a transaction until a block including the transaction is added to the blockchain.

From the view point of end users, the transaction-confirmation time is an important performance measure for the Bitcoin system. The authors in [16] analyzed the transaction fees paid with 55.5 million transactions recorded in the blockchain, investigating trends of transaction-fee convention in Bitcoin. It is reported that the confirmation time of transactions without fee are longer than those with fee, and that the latency of transactions with high fee are smaller than those with low fee. This result suggests some priority mechanism is equipped in the Bitcoin system, in which transactions with high fees are included in a block faster than those with low fees.

2.2 DSS Trilemma

A blockchain technology is classified into two types: public and consortium/private [24]. In a public blockchain network, any node can join the network and verify transactions and blocks without any permission. Bitcoin and Ethereum are categorized into public blockchain. In consortium/private blockchain network, on the contrary, only the authenticated nodes can activate functionalities for maintaining blockchains. Hyperledger projects [27], started by Linux Foundation and including Hyperledger Fabric and Hyperledger Iroha, are of consortium-type blockchain.

In terms of the nature of blockchain, the following three aspects are considered as important characteristics that blockchain developers must take into account.

- Decentralized: Developing a blockchain system that does not rely on a single point of control.
- Scalable: Enabling a blockchain system to process an increasingly growing number of transactions.
- Secure: Making a blockchain system to prevent from attacks, malfunctions, and unforeseen misbehavings.

It has been known as a rule of thumb that any blockchain technology satisfies at most two of the above three aspects and that no blockchain technologies satisfy all the three features. For example, the Bitcoin blockchain satisfies decentralized and security natures but is not scalable. A permissioned blockchain such as Hyperledger Fabric provides high transaction throughput (scalable) and secure transaction processing, however, only the authenticated nodes join the blockchain network (not decentralized). This

[†] The reward to the winning miner is composed of coinbase and fees of transactions included in the generated block. In 2021, the output value of the coinbase for one-block mining is 6.25 bitcoin. This output value is halved every 210,000 blocks, corresponding to a four-year halving schedule.

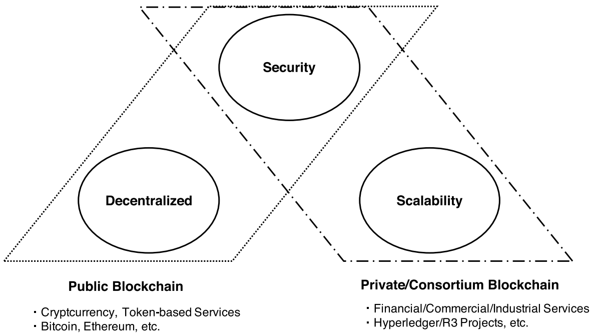
Security
Decentralized
Scalability
Public Blockchain
· Cryptcurrency, Token-based Services
· Bitcoin, Ethereum, etc.
Private/Consortium Blockchain
· Financial/Commercial/Industrial Services
· Hyperledger/R3 Projects, etc.

Fig. 1 DSS trilemma.

tradeoff among the three characteristics in blockchain mechanism is called DSS trilemma [4]. Figure 1 shows the DSS trilemma of the blockchain.

A crucial key point of blockchain technologies is how to make consensus for a newly generated block among all the nodes joining the blockchain network. A long time interval for consensus in public blockchain enables a large number of nodes to join the blockchain network, however, causing a low transaction throughput. On the contrary, a short consensus making time for consortium/private blockchain provides a high-speed transaction processing, however, increasing security threats if anonymous nodes are allowed to join the blockchain network.

2.3 Related Work for Performance Modeling of Blockchain

The performance issue of blockchains is classified into two categories: scalability issue and security one. For both issues, there exist much literature on the performance evaluation of blockchains. A comprehensive survey on the blockchain evaluation research is provided in [23]. The readers who are interested in the analytical approach to the blockchain performance are referred to [23].

In terms of the scalability issue, the transaction-confirmation time (or the transaction throughput) is an important performance measure. The transaction-confirmation time of the Bitcoin blockchain is composed of 1) the transaction-waiting time in memory pools of miner nodes, 2) the block-generation time (the mining time for the block including the transaction), and 3) the block-propagation delay.

To analyze the transaction-confirmation time, a typical approach is to model the Bitcoin blockchain as a queueing model. In [15] and our previous work [9]–[12], a main interest is to characterize the queueing dynamics of transactions in miner nodes. From this point of view, a basic model of the Bitcoin blockchain is a single-server queueing system, in which transactions waiting in the memory pool and block-generation time are taken into consideration.

In terms of the Bitcoin security issue, however, the block-propagation delay is also important. A pioneering work for analyzing the block propagation of Bitcoin is [6]. The authors of [6] focused on how the block propagation delay affects the security of the Bitcoin blockchain. They assumed that the block-generation time follows a Poisson process, deriving the probability of blockchain fork occurrence. A further refined model for the fork probability was proposed in [19].

In [3], the authors extensively studied the block-generation process, investigating the applicability of stochastic processes to the block-generation model. They considered three elements for the block-generation process: the hash-rate function, difficulty adjustments, and block-propagation delay, and several block-arrival models were compared with the real data. It is found that a nonhomogeneous Poisson process with periodic hash-rate function is the simplest model which algorithmically generates arrival sequences for simulation and reasonably approximates the block-generation process. From queueing theoretical point of view, however, queueing systems with nonhomogeneous Poisson input are hard to analyze in general.

3. Queueing System with Batch Service for Bitcoin Blockchain

In this section, we summarize the fundamental queueing model for Bitcoin Blockchain in [11]. We also show some numerical examples, discussing how transactions are handled in the real Bitcoin system.

3.1 Basic Queueing Model

As we described in Sect. 2, a transaction issued by an end user is first stored in memory pools of miner nodes. Each miner node creates a block from the transactions stored in the memory pool. Then the block created by a winning miner node is added to the blockchain and the transactions in the block are removed from the memory pool. Figure 2 shows the transaction-confirmation process in the memory pool of a miner node.

If we regard a transaction as a customer, the transaction-confirmation process can be modeled as a queueing system with batch service, in which several customers in the service facility are served and depart from the system simultaneously at service completion.

We assume that transactions arrive at the Bitcoin system according to a Poisson process with rate λ. Arriving transactions are stored in the queue with infinite capacity and those are served in a batch manner.

When a transaction arrives at the system in idle, its service immediately starts. The transactions consecutively arriving at the system are served in a batch manner until the number of batch size reaches the batch-size limit b. In other words, newly arriving transactions are included into the block under mining as long as the resulting block size does not reach the batch size $b^{\dagger}$.

Remind that the service time of the queueing model is

†The default Bitcoin client processes transactions in this manner. See [1] for details.

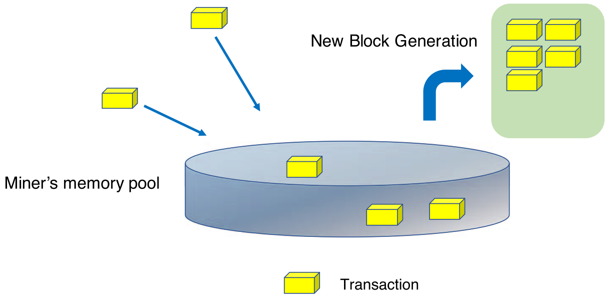

Fig. 2 Transaction processing in memory pool.

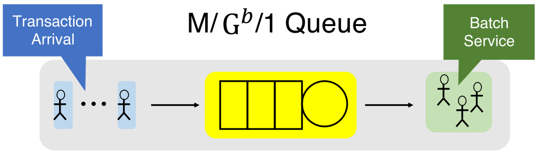

Fig. 3 M/G^b/1 queueing model.

the block-generation time. Suppose that block-generation times are independent and identically distributed (i.i.d.). Let S denote the block-generation time, whose distribution function and probability-density function are $G(x)$ and $g(x)$, respectively.

From the above assumptions, the resulting queueing model is an M/G^b/1 queue (Fig. 3).

3.2 Mean Transaction-Confirmation Time

Let $N(t)$ and $X(t)$ denote the number of transactions in the system at time t and the elapsed service time at t, respectively. We also define for $x, t \geq 0$, $n = 1, 2, \ldots$,

$$P_n(x,t)\mathrm{d}x = \Pr\{N(t) = n, x < X(t) \leq x + \mathrm{d}x\},$$
$$P_0(t) = \Pr\{N(t) = 0\}.$$

If the stability condition $\lambda E[S] < b$ holds, the following limiting probabilities exist.

$$P_n(x) = \lim_{t\to\infty} P_n(x,t),$$
$$P_0 = \lim_{t\to\infty} P_0(t).$$

From the assumptions of the queueing model, we obtain

$$\lambda P_0 = \sum_{k=1}^{b} \int_0^\infty P_k(x)\xi(x)\mathrm{d}x, \qquad (1)$$

$$\frac{\mathrm{d}}{\mathrm{d}x}P_n(x) = -\{\lambda + \xi(x)\}P_n(x) + \lambda P_{n-1}(x), \qquad n = 2, 3, \ldots, \qquad (2)$$

$$\frac{\mathrm{d}}{\mathrm{d}x}P_1(x) = -\{\lambda + \xi(x)\}P_1(x), \qquad (3)$$

where $\xi(x)$ is the hazard function (or failure rate) in terms of S and given by

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

In the equation of (1), the left-hand side (l.h.s.) implies the exiting rate from state 0, while the right-hand side (r.h.s.) is the entering rate into state 0, that is, (1) is a type of balance equation. Similarly, the first term in the r.h.s. of (2) is yielded from the event that the number of transactions remains the same during a small time interval, while the second term is induced from the event that a transaction arrival occurs when there exist n transactions in the system.

We define T as the sojourn time of a transaction in system. Note that the transaction-sojourn time T is the time interval from the transaction-arriving time to the time epoch at which the block including the transaction is confirmed. That is, T is corresponding to the transaction-confirmation time. Under the assumptions of the queueing model, we have the following theorem.

Theorem 1 ([11] Theorem 5.1): The mean transaction confirmation time $E[T]$ is given by

$$E[T] = \frac{1}{2\lambda^2(b - \lambda E[S])}\left(\sum_{k=1}^{b} \alpha_k \Big[b(b-1) + \{(b+1)b - k(k-1)\}\lambda E[S] + (b-k)\lambda^2 E[S^2]\Big] - \lambda\left\{b(b-1) - \lambda^2 E[S^2]\right\}\right) \equiv f(\lambda), \qquad (4)$$

where

$$\alpha_k = \int_0^\infty P_k(x)\xi(x)\mathrm{d}x.$$

Proof : See Appendix A.1 in [11].

For later use, we define $f(\lambda)$ as the function of λ for the mean transaction-confirmation time $E[T]$.

3.3 Transaction-Confirmation Time for Priority Queueing Model

As we described in Sect. 2.1, transactions with high fees are more likely to be included in a block than those with low fees. In order to investigate the hypothesis, we consider a queueing system with batch service and priority discipline. In this priority-queueing model, transactions are prioritized for the inclusion to a block.

Let c denote the number of priority classes of transactions. For $1 \leq i \leq j \leq c$, class-i transactions have priority over transactions of class j. We assume that class-i transactions arrive at the system according to a Poisson process with arrival rate λ_i, independently of other transaction-class arrivals. For the system stability, $\sum_{i=1}^{c} \lambda_i E[S] < b$ is assumed. Let T_i denote the sojourn time of class-i transactions. We also define $\overline{\lambda}_i$ as

$$\overline{\lambda}_i = \sum_{k=1}^{i} \lambda_k, \qquad i = 2, 3, \ldots, c.$$

We define T_i ($i = 1, \ldots, c$) as the confirmation time of class-i transactions. For the mean confirmation time of class-i transactions, we have the following theorem.

Theorem 2 ([11] Theorem 5.2): Assume the system is work conserving. Then, $E[T_1]$ is given by

$$E[T_1] = f(\lambda_1).$$

We can calculate $E[T_i]$ ($i = 2, 3, \ldots, c$) recursively with $E[T_j]$'s ($j = 1, \ldots, i-1$) by

$$E[T_i] = \frac{1}{\lambda_i}\left(\overline{\lambda}_i f(\overline{\lambda}_i) - \sum_{k=1}^{i-1} \lambda_k E[T_k]\right).$$

Proof : See Appendix A.2 in [11].

If we consider two-priority case ($c = 2$), we have the following simple formulae.

Corollary 1: Let λ_H and λ_L denote arrival rates of high- and low-priority transactions, respectively. We define T_H and T_L as the sojourn time of high-priority transactions and that of low-priority ones, respectively. Then $E[T_H]$ and $E[T_L]$ are given by

$$E[T_H] = f(\lambda_H),$$
$$E[T_L] = \left(\frac{\lambda_H}{\lambda_L} + 1\right) f(\lambda_H + \lambda_L) - \frac{\lambda_H}{\lambda_L} f(\lambda_H).$$

3.4 Block-Generation Time

In [7], the authors assumed that the block-generation time follows an exponential distribution. An idea behind the assumption of the exponential block-generation time is that the difficulty of hash calculation is too high to detect a solution, allowing us to suppose that the probability of solution detection by a miner node is constant and very small. That is, a hash calculation by a miner node can be regarded as an independent Bernoulli trial, and this leads to a geometric distribution of the number of experiments for the first success. We can approximate this geometric distribution by exponential one.

It was reported in [1] that a miner explores in the nonce-word space consisting of 4-byte nonce field in the block header and the script area of the coinbase transaction. Since the size of the coinbase script area is 100-byte data, the total amount of the nonce-word space is 104 bytes (= 832 bits), i.e., 2^{832} nonce words.

The nonce-word space is huge, however, mining is performed not only by independent miner nodes, but also by groups of miner nodes called mining pools. A mining pool is composed of several miner nodes, and those contribute their computation power to the detection of solutions. The existence of mining pools suggests that the solution-detection probability changes over time. The growth of the difficulty presented in [26] also supports the increase in the solution-detection probability.

In our previous work [11], we considered a simple urn model without replacement for modeling the block-generation process[†]. Assume that we have an urn containing M balls: one red ball and $M-1$ white ones. We withdraw a ball from the urn at a time, and then remove it from the urn without replacement. In this setting, the probability that the red ball is drawn at kth trial is $1/M$, i.e., a discrete-uniform distribution. Assuming one trial of withdrawing a ball is performed at a unit time, the probability of the event that the red ball is drawn at time k is given by $1/M$.

Let n denote the number of miner nodes in the system. We define X_i ($i = 1, 2, \ldots, n$) as the time at which miner i finds a red ball. Suppose that X_i's are i.i.d. Then the block-generation time, denoted by L_n, is given by

$$L_n = \min\{X_1, X_2, \ldots, X_n\}.$$

Assuming that X_i follows a continuous-uniform distribution $U(0, M)$, we have

$$\begin{aligned} \Pr\{L_n \le x\} &= \Pr\{\min(X_1, \ldots, X_n) \le x\} \\ &= 1 - \Pr\{\min(X_1, \ldots, X_n) > x\} \\ &= 1 - \left(1 - \frac{x}{M}\right)^n. \end{aligned}$$

Now consider a limit distribution of $(L_n - b_n)/a_n$ for sequences of constants $\{a_n > 0\}$ and b_n. It is known in extreme value theory that the distribution of $(L_n - b_n)/a_n$ converges to a Weibull distribution when X_i follows uniform distribution ([14] p. 59, Table A.1). For $0 \le z \le n$, setting $a_n = 1/n$ and $b_n = 0$ for $(L_n - b_n)/a_n$ yields

$$\begin{aligned} \Pr\left\{\frac{L_n - b_n}{a_n} \le z\right\} &= 1 - \left\{1 - \frac{(z/M)}{n}\right\}^n \\ &\to 1 - e^{-z/M}, \quad n \to \infty. \end{aligned}$$

This result enables us to approximate the distribution of L_n for a large n by

$$\Pr\{L_n \le x\} \approx 1 - e^{-(n/M)x},$$

i.e., L_n approximately follows an exponential distribution when n is large. Note that M is related to the mining difficulty. In the Bitcoin system, the value of M is automatically adjusted such that the mean block-generation time $1/\mu = M/n$ remains 600 [s].

Figure 4 illustrates the relative frequency of the block-generation time for the measured data[††] and the probability density function of the exponential distribution with

[†] As we explained in Sect. 2.1, the mining difficulty is adjusted such that the block-generation time is 10 minutes on average. The Bitcoin system performs this adjustment every two weeks. This implies that the difficulty just after system adjustment is too high for miner nodes to detect solutions, while the difficulty at a time close to the next system adjustment is relatively small so that hash solutions are likely to be detected. From this observation, the Bernoulli trial model describes the mining situation just after system adjustment, while our urn model mimics the mining process before updating the mining difficulty.

[††] We collected two-year data (October 2013 to September 2015) of blocks and transactions from [26].

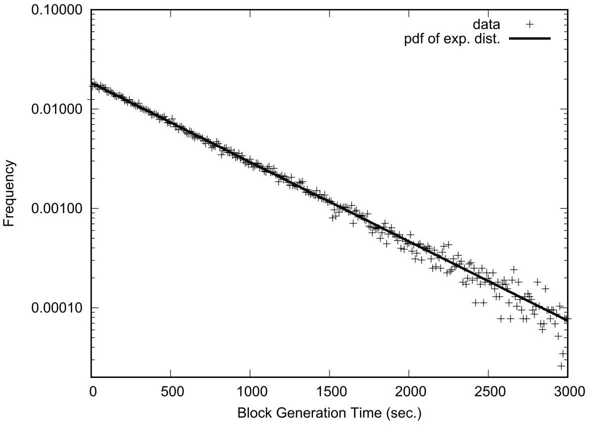

Fig. 4 Relative frequency and exponential probability density function for bock-generation time.

the same rate. The horizontal axis represents the block-generation time in second, and the vertical axis is the logarithmic scale of the frequency values. In this figure, the resulting curve for the exponential distribution exhibits a good agreement with the measured data.

3.5 Findings from Numerical Examples

3.5.1 Discrepancy between Analysis and Measurement Data

First, we compare analytical results of the mean transaction-confirmation time and measurement ones. From the previous subsection, $G(x)$ is set to the exponential distribution

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

where μ is set to 1.8379×10^{-3} from the two-year data of 2013–2015. We also obtain from the data the transaction arrival rate and batch-size limit as $\lambda = 0.97275$ and $b = 1750$, respectively.

From the parameter setting and (4), we obtained $E[T] = 568.10$ [s], while the average of transaction-confirmation times of the data is 1,075.0 [s]. That is, the analytical result is almost a half of the averaged value over the data. Note that the mean block-generation time is $1/\mu = 544.095$ [s]. It is easily verified that the stability condition $\lambda E[S] < b$ is satisfied and that the system is under low utilization.

Remind the assumption of transaction inclusion in our analytical model; an arriving transaction is included into the block under mining if the size of the block under mining does not reach the batch size b. When the system is under low utilization, the arriving transaction is likely to be included in the block following the currently processed block. The above result suggests, however, that an arriving transaction is not included in the block under mining.

The exponential block-generation time also supports this conjecture. Assume an arriving transaction is not included in the block under mining. When the system utilization is low, the confirmation time of an arriving transaction is likely to be composed of the remaining mining time of the currently processed block and the mining time for the next block. Since the exponential block-generation time is memoryless, the remaining block-generation time also follows the same exponential distribution. Therefore, the transaction-confirmation time under low utilization is the sum of two exponential block-generation times, supporting that the transaction-confirmation time is almost twice as large as the block-generation time. See Appendix for the alternate approximation discussion.

Table 1 Comparison of analysis and measurement for the transaction-confirmation time.

Transaction Type	Arrival Rate	Measurement	Analysis
H	0.90466	874.13	562.16
L	0.068082	3,744.1	647.05

3.5.2 Fee-Based Priority Mechanism

In this subsection, we consider how the transaction-confirmation time is affected by the amount of fee. We classify transactions into two types: high (H) and low (L). Transactions with fee greater than or equal to 0.0001 BTC are prioritized as H class, while those with fee smaller than 0.0001 BTC are classified into L class. Table 1 shows the results of measurement and analysis for the confirmation time for class-H and class-L transaction. We compute $E[T_H]$ and $E[T_L]$ from the equations in Corollary 1.

Table 1 shows large discrepancies between measurement and analysis for H and L classes. In particular, we observe a significant discrepancy for the L class. Note that in our queueing model, we assumed work conserving with which low class transactions are served as long as the block under mining is not filled with higher-priority class transactions. The large discrepancy for the L class in Table 1 suggests that in the real Bitcoin system, L-class transactions are less likely to be served. We can conjecture the existence of miners who intentionally exclude transactions with small fees from the block inclusion process.

3.5.3 Scalability Issue

Figure 5 illustrates how the transaction-arrival rate λ affects the mean transaction-confirmation time $E[T]$. Here, we plot four cases of $b = 1000$, 2000, 4000 and 8000. In this figure, $E[T]$ for each case rapidly increases when λ is close to the upper bound $b/E[S]$ with which the stability condition is violated. Here, we call $b/E[S]$ as the maximum transaction throughput (MTT).

Table 2 shows the relation between the block-size limit and MTT. In this table, the block-size limit of each b is calculated by $b/1750$ based on the statistical analysis of the two-year data, i.e., the 1-Mbyte block accommodates 1750 transactions on average. It is observed from this table that the MTT slightly increases with the block-size limit. Even when $b = 8000$, i.e., the block-size limit is about 4.5 Mbyte,

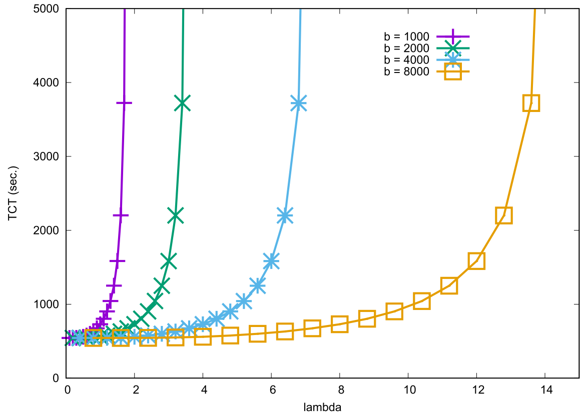

Fig. 5 Mean transaction-confirmation time.

Table 2 Approximate block size and maximum transaction throughput.

b	1000	2000	4000	8000
Block Size (Mbyte)	0.5714	1.1429	2.2857	4.5714
MTT (tps)	1.8379	3.6759	7.3517	14.703

the MTT is 14.7 tps. This result implies that enlarging the block-size limit slightly improves the transaction throughput but does not fundamentally solve the scalability issue of Bitcoin[†].

4. Batch-Service Queue with General Input

In [9], [12], we developed the M/G^b/1 model [11] of the previous section to the one in which an arriving transaction is not included in the block under mining. In [9], we considered a single-server queueing system with batch service and multiple vacation policy. The priority mechanism was further taken into consideration in [12]. In both models, the resulting mean transaction-confirmation time showed a good agreement with the measured data. In comparison of the analysis and trace-driven simulation, however, a large discrepancy was observed for the transaction-confirmation time in small block-size limit region. From the measured data, we found that the variation of the transaction-arrival process is larger than that of Poisson process.

In order for further improvement of the queueing model, we considered a queueing system with batch service and general input [10]. In this queueing model, the transaction interarrival times are i.i.d. according to a general distribution. The resulting queueing model is a GI/M^b/1. In this section, we briefly summarize the GI/M^b/1 and numerical results in [10].

4.1 GI/M^b/1 Queueing Model

Let A_i $(i = 1, 2, \dots)$ denote the i-th transaction interarrival time. We assume A_i's are i.i.d. and follow a general distribution $H(x)$. The block-generation time is i.i.d. and follows an exponential distribution with rate μ. Transactions are served in a batch-service manner, and we define the maximum batch size as b. Suppose that an arriving transaction is not included in the block under mining.

4.2 Analysis

We define N_n $(n = 1, 2, \dots)$ as the number of transactions in the system just before the n-th transaction arrival. The number of mining completions from the n-th and $(n+1)$-st transaction arrival points is denoted by Y_n. Let $D_k^{(n)}$ denote the number of transactions departing from the system at the k-th mining completion point, counted from the n-th transaction arrival epoch.

It is easy to see that for $n = 1, 2, \dots$, N_n satisfies the following equation

$$N_{n+1} = \max\left\{N_n - (Y_n - 1)b - D_{Y_n}^{(n)} + 1, 0\right\}. \qquad (5)$$

Since the block-generation time is i.i.d. and follows an exponential distribution, $\{Y_n : n = 1, 2, \dots\}$ are i.i.d. and follows a Poisson distribution. Therefore, $\{N_n : n = 1, 2, \dots\}$ satisfying (5) is a discrete-time Markov chain whose state-transition probability $p_{ij} = \Pr\{N_{n+1} = j | N_n = i\}$ is given by

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

where

$$a_k = \int_0^\infty e^{-\mu x} \frac{(\mu x)^k}{k!} \mathrm{d}H(x), \qquad \overline{a}_k = \sum_{i=k}^{\infty} a_i,$$

and $\lfloor x \rfloor$ is the floor function of x. Then, the transition-probability matrix $P = (p_{ij})$ is given by

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

where

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

[†] According to [29], VISA can process 65,000 transactions per second at peak time. We can calculate from Table 2 the maximum block size, with which Bitcoin can handle the same amount of transactions as VISA, by:

$$\frac{65,000}{1.8379} \times 0.5714 \approx 20208.0 \text{ [Mbytes]}$$

That is, the block size of more than 20 Gigabytes is needed. Such a large block makes not only the block-transfer delay large, but also the network overloaded. This large block is not a practical solution for solving the scalability issue of Bitcoin.

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

We can see from the transition matrix $\boldsymbol{P}$ that the Markov chain $\{N_n : n = 1, 2, \ldots\}$ is irreducible and aperiodic. If the stability condition $\lambda/\mu < b$ holds, we have the steady-state distribution

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

We define $\boldsymbol{\pi}$ as the steady-state probability vector, given by $\boldsymbol{\pi} = (\pi_k)$. 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). Defining

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

we can rewrite $\boldsymbol{\pi}$ as $\boldsymbol{\pi} = (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots)$. Since the structure of the transition probability matrix $\boldsymbol{P}$ is a GI/M/1-type Markov chain, we have [18]

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

We can obtain $\boldsymbol{R}$ numerically by the following matrix recursive equation of $\boldsymbol{R}(n)$

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

With $\boldsymbol{R}$, we can solve $\boldsymbol{\pi}_0$ and $\boldsymbol{\pi}_1$ from the following equations

$$\boldsymbol{\pi}_0 = \boldsymbol{\pi}_0 \boldsymbol{B}_0 + \boldsymbol{\pi}_1 (\sum_{l=1}^{\infty} \boldsymbol{R}^{l-1} \boldsymbol{B}_l),$$

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

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

Let W denote the transaction waiting time in the memory pool. Then the mean transaction waiting time $E[W]$ is given by

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

The mean transaction-confirmation time $E[T]$ can be calculated by $E[T] = E[W] + E[S]$.

4.3 Numerical Examples

In our previous work [10], we considered a hyper-exponential distribution for the transaction interarrival time.

Table 3 Comparison of transaction interarrival time statistics for measured data and estimates by EM algorithm.

Period	Type	Mean	SD	CV
Oct. 2013 – Sept. 2014	Data	1.36296	5.07571	3.72401
	EM	1.36296	4.42236	3.24465
Oct. 2014 – Sept. 2015	Data	0.82772	12.68489	15.32505
	EM	0.82772	12.58913	15.20936

SD: Standard Deviation, CV: Coefficient of Variation

Table 4 Comparison of transaction-confirmation times for data-driven simulation and analysis.

Period	Type	Block Size		
		1000	1500	2000
Oct. 2013 – Sep. 2014	DDS	1080.27	1031.53	1022.77
	Analysis	1061.35	1022.95	1015.24
Oct. 2014 – Sep. 2015	DDS	14053.6	1540.40	1240.69
	Analysis	1877.20	1337.00	1238.18

DDS: Data-Driven Simulation

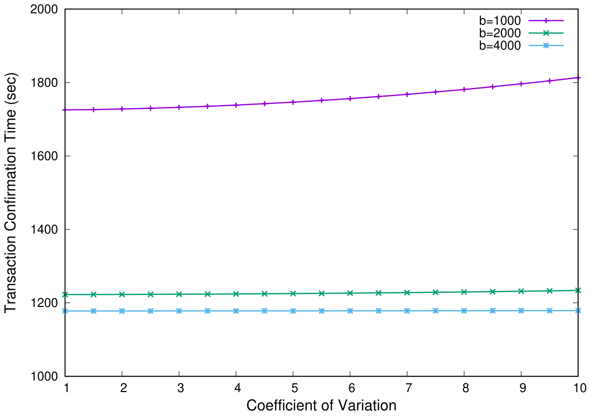

Fig. 6 The impact of coefficient of variation of the transaction-interarrival time on the transaction-confirmation time.

The parameters of the hyper-exponential distribution were estimated by the EM algorithm [13]. Table 3 shows the comparison results of statistics for transaction interarrival times of measured data and estimates by EM algorithm. We consider two measured data sequences: the period of 1 October 2013 to 30 September 2014, and that of 1 October 2014 to 30 September 2015. In Table 3, the mean, standard deviation, and coefficient of variation of the resulting hyper-exponential distribution show a good agreement with those of actual data.

In terms of the transaction-confirmation time, we also observed a good agreement between the analytical result and trace-driven simulation one for large block-size limit. When the block-size limit is small, however, we observed a discrepancy between analytical and simulation results. (See Table 4.) We also found that for the block-size limit of 1 Mbyte, Poisson process gives a good estimate of the transaction-confirmation time.

We further investigated how the coefficient of variation of the hyper-exponential distribution affects the transaction-confirmation time (Fig. 6). We found from this figure that the transaction-confirmation time slightly increases with the coefficient of variation. When the block size is large, however, the transaction-confirmation time is not affected by the variation of the transaction interarrival time. As we explained in Sect. 3.5, the mean number of transactions in a block is $b = 1750$. The result of Fig. 6 suggests that current block size is so large that the transaction-confirmation time is not affected by the variation of the transaction arrival process significantly.

5. Conclusion

In this paper, we showed queueing models for analyzing the transaction-confirmation process in the Bitcoin blockchain. A key observation was that transactions are processed in a block basis, which can be modeled by a queueing system with batch service.

We first introduced the single-server queue with batch service and priority mechanism. We also showed the queueing system with batch service and general input. The summary of the findings from numerical results of the queueing models is as follows:

- M/G^b/1 model
  The average transaction-confirmation time of measured data is almost twice larger than that of the analysis. This result implies that arriving transaction is not included in the block under mining.
- M/G^b/1 with priority discipline
  The average confirmation time of low-priority transactions for measured data is significantly larger than that for analysis. This implies that the queueing mechanism of the Bitcoin system is not work conserving, suggesting the existence of the miners that intentionally exclude transactions with low fee.
- G/M^b/1 model
  The discrepancy of the transaction-confirmation time between analysis and data-driven simulation is large when the block-size limit is small. This results from the non Poisson arrival process for transactions. However, this discrepancy decreases with the increase of the block size, and we confirmed that the proposed queueing model provides good estimates of the traction-confirmation time under the current block size of 1 Mbyte.

As we presented in Sect. 2.2, DSS trilemma is a difficult and important issue. In order to tackle the DSS trilemma and to make public blockchains further scalable, we have conducted several research projects under Grant-in-Aid for Scientific Research (A) No.19H01103 "Informatics Study on Ultra-Scalable Blockchain Technology" [28]. Here, we consider performance modeling issues in blockchain systems, advanced data structure for blockchains, and applications of blockchain technologies to IoT systems. The readers who are interested in the research products are referred to [28].

Acknowledgments

The author would like to thank the editorial committee members of this special section for giving him a chance to write this invited paper. This research was supported in part by Japan Society for the Promotion of Science under Grant-in-Aid for Scientific Research (A) No. 19H01103, and the Support Center for Advanced Telecommunications (SCAT) Technology Research Foundation.

References

[1] A.M. Antonopoulos, Mastering Bitcoin, O'Reilly, 2014.
[2] T. Bamert, C. Decker, L. Elsen, R. Wattenhofer, and S. Welten, "Have a snack, pay with Bitcoins," 2013 IEEE Thirteenth International Conference on Peer-to-Peer Computing, pp.1–5, 2013.
[3] R. Bowden, H.P. Keeler, A.E. Krzesinski, and P.G. Taylor, "Modeling and analysis of block arrival times in the Bitcoin blockchain," Stochastic Models, vol.36, no.4, pp.602–637, 2020.
[4] M. Conti, A. Gangwal, and M. Todero, "Blockchain Trilemma solver algorand has dilemma over undecidable messages," 14th International Conference on Availability, Reliability and Security (ARES'19), Canterbury, CA, UK, 2019.
[5] M. Conti, E. Sandeep Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of Bitcoin," IEEE Commun. Surveys Tuts., vol.20, no.4, pp.3416–3452, Fourthquarter 2018.
[6] C. Decker and R. Wattenhofer, "Information propagation in the Bitcoin network," 13th IEEE International Conference on Peer-to-Peer Computing, pp.1–10, 2013.
[7] J. Göbel, H.P. Keeler, A.E. Krzesinski, and P.G. Taylor, "Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay," Performance Evaluation, vol.104, pp.23–41, 2016.
[8] G.O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in Bitcoin," The 2012 ACM Conference on Computer and Communications Security, pp.906–917, 2012. http://dl.acm.org/citation.cfm?id=2382292
[9] Y. Kawase and S. Kasahara, "Transaction-confirmation time for Bitcoin: A queueing analytical approach to blockchain mechanism," 12th International Conference onf Queueing Theory and Network Applications (QTNA2017), LNCS, vol.10591, pp.75–88, 2017.
[10] Y. Kawase and S. Kasahara, "A batch-service queueing system with general input and its application to analysis of mining process for Bitcoin blockchain," 2018 IEEE International Conference on Blockchain (Blockchain-2018), pp.1440–1447, 2018.
[11] S. Kasahara and J. Kawahara, "Effect of Bitcoin fee on transaction-confirmation process," Journal of Industrial and Management Optimization, vol.15, no.1, pp.365–386, Jan. 2019.
[12] Y. Kawase and S. Kasahara, "Priority queueing analysis of transaction-confirmation time for Bitcoin blockchain," Journal of Industrial and Management Optimization, vol.16, no.3, pp.1077–1098, May 2020.
[13] R.E.A. Khayari, R. Sadre, and B.R. Haverkort, "Fitting world-wide web request traces with the EM-algorithm," Perform. Evaluation, vol.52, no.2-3, pp.175–191, 2003.
[14] S. Kotz and S. Nadarajah, Extreme Value Distributions: Theory and Applications, Imperial College Press, 2000.
[15] Q.-L. Li, J.-Y. Ma, and Y.-X. Chang, "Blockchain queue theory," Computational Data and Social Networks, Lecture Notes in Computer Science, vol.11280, pp.25–40, Springer, 2018.
[16] 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, vol.8976, pp.19–33, Springer, 2015.

[17] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," https://bitcoin.org/bitcoin.pdf, 2008.
[18] R. Nelson, Probability, Stochastic Processes, and Queueing Theory, Springer-Verlag, 1995.
[19] H. Seike, Y. Aoki, and N. Koshizuka, "Fork rate-based analysis of the longest chain growth time interval of a PoW blockchain," 2019 IEEE International Conference on Blockchain (Blockchain-2019), Atlanta, GA, USA, pp.253–260, 2019.
[20] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," IEEE Commun. Surveys Tuts., vol.18, no.3, pp.2084–2123, 2016.
[21] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," https://bitcoin.org/bitcoin.pdf, 2008.
[22] P. Rizun, "A transaction fee market exists without a block size limit," https://scalingbitcoin.org/papers/feemarket.pdf, 2015.
[23] S. Smetanin, A. Ometov, M. Komarov, P. Masek, and Y. Koucheryavy, "Blockchain evaluation approaches: State-of-the-art and future perspective," Sensors, vol.20, no.12, article-no.3358, 2020.
[24] W. Wang, D.T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D.I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," IEEE Access, vol.7, pp.22328–22370, 2019.
[25] https://en.bitcoin.it/wiki /Scalability
[26] https://blockchain.com/
[27] https://www.hyperledger.org/
[28] https://kaken.nii.ac.jp/en/grant/KAKENHI-PROJECT-19H01103/
[29] https://usa.visa.com/dam/VCOM/global/about-visa/documents/visa-fact-sheet-july-2019.pdf

Appendix: Transaction-Confirmation Time under Low Utilization

Assume that a transaction arrives at the system when the n-th block is under mining. Let S_n denote the n-th block generation time. We define $\tilde{S}_n$ as the remaining n-th block-generation time, i.e., the time interval from the transaction arriving point to the time epoch at which the n-th block mining ends. Suppose the arriving transaction is included in $(n + j)$-th block $(j = 1, 2, \dots)$. Then, the transaction-confirmation time T is given by

$$T = \tilde{S}_n + S_{n+1} + \cdots + S_{n+j}.$$

When the system utilization is low, the number of transactions in the mining pool is likely to be small, and the arriving transaction is included in $(n + 1)$-st block with a large probability. Roughly speaking, we can approximate T under low utilization by

$$T \approx \tilde{S}_n + S_{n+1}.$$

Since block-generation times $\{S_n\}$'s are i.i.d. and follow the exponential distribution of $G(x)$, $\tilde{S}_n$ also follows the same exponential distribution due to the memoryless property. Since S_n is equal to S in distribution, we have $T \approx 2S$, which yields $E[T] \approx 2E[S]$.

Shoji Kasahara received the B. Eng., M. Eng., and Dr. Eng. degrees from Kyoto University, Kyoto, Japan, in 1989, 1991, and 1996, respectively. Currently, he is a Professor of Division of Information Science, Nara Institute of Science and Technology, Nara, Japan. His research interests include stochastic modeling and analytics of large-scale complex systems based on computer/communication networks.