A Variable Bulk Arrival and Static Bulk Service Queueing Model for Blockchain

Jongho Seol
Computer Science
Oklahoma State University
Stillwater, OK, USA
jongho@okstate.edu

Abhilash Kancharla
Computer Science
Oklahoma State University
Stillwater, OK, USA
abhilak@okstate.edu

Zuqiang Ke
Computer Science
Oklahoma State University
Stillwater, OK, USA
zuqiang.ke@okstate.edu

Hyeyoung Kim
Game
Hongik University
Seoul, Korea
hykim@hongik.ac.kr

Nohphill Park
Computer Science
Oklahoma State University
Stillwater, OK, USA
npark@cs.okstate.edu

ABSTRACT

This paper proposes a novel embedded Markovian queueing model of the $M^{1,n}/M^n/1$ type in order to establish a theoretical foundation to design a blockchain-based system with specific interest in Ethereum. The model assumes variable bulk arrivals of transactions in Poisson distribution, i.e., $M^{1,n}$, where n the number of slots across all the mined transactions and static bulk service of transactions in exponential time, i.e., M^n, for posting in the current block, namely, VBASBP. The primary performance measurements to be taken are the average number of slots no matter how many transactions are mined under the assumption of the maximum number of slots per block as specified by n; the average waiting time per slot; and the throughput in terms of the average number of slots to be processed per time. The variable bulk arrival rate is assumed to vary linearly proportional to the size of the transactions in a multiple of λ per slot, and the static bulk service is assumed to take place when the number of slots in the mined transactions reaches at n, i.e., a bulk processing of multiple transactions in multiple slots for posting in a block. Numerical simulations are conducted on Matlab to demonstrate the efficacy of the model.

BSCI '20, October 6, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7610-5/20/10$15.00
https://doi.org/10.1145/3384943.3409423

KEYWORDS

Blockchain, Queueing, Transactions, Transaction pool, Block posting, Arrival time, Service time, Waiting time, Queue size, Throughput

ACM Reference format:

Jongho Seol, Abhilash Kancharla, Zuqiang Ke, Hyeyoung Kim and Nohphill Park. 2020. A Variable Bulk Arrival and Static Bulk Service Queuing Model for Blockchain. In *Proceedings of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure (BSCI '20). ACM, Taipei, Taiwan, 10 pages.* https://doi.org/10.1145/3384943.3409423

1 Introduction

Blockchain technology [1,2,4,5,7,10,18] is undergoing a tremendous growth choking itself up to its capacity and performance limits and thereby resulting in cost spike [20]. It is exigently sought to address and respond to these issues, being mostly concerned about the scalability [11,12,14,15,16,17,25], dependability of the blockchain system, and privacy issues in systems such as smart grid and IoT [19,26,27,28]. In this context, blockchain technology is emerging and gaining more and more attention from the technology community as an alternative to the current centralized-based internet protocol, namely, the decentralization with blockchain-based peer to peer internet protocol.

The central idea driving the technology shift is the trust of the internet service which the legacy centralized-based internet service providers evidently have never convinced the users' community to sustain its credibility due to a few major reported breaches of trust no matter intentional or not, which triggered the technology drive towards blockchain which is today's and tomorrow's promise and the best-known solution of trust in internet services. However, as the blockchain technology matures, it appears the technology is revealing either expected or unexpected performance flaws besides the main concern of trust, such as scalability and speed issues, as the main hurdle from success. The key to the success of this technology in the market is how to bring the performance to the next level in order to cope with the market needs and requirements from the performance perspective.

A prerequisite to the performance needs and requirements is to establish an analytical model in order to evaluate and assure the performance of concern with high efficacy in a quantitative manner during the early design cycle. There have been a few analytically-approached works reported on dependability [21,24,25]. However, they are able to readily and adequately address the queueing nature of the transactions flow and block posting in particular, which will eventually prevent a synergistic and comprehensive analytical design of a dependable and high-performance blockchain system.

In this context, the primary interest of this paper is to develop an embedded Markovian queueing model of the $M^{1,n}/M^n/1$ type in order to establish a theoretical foundation to design a blockchain-based system with focus on the stochastic behavior of the mined transactions waiting to be posted for the block delay as the bulk synchronization point. The model assumes variable bulk arrivals of transactions in Poisson distribution, i.e., $M^{1,n}$, where n the number of slots across all the mined transactions and static bulk services of transactions in exponential time, i.e., M^n, for posting in the current block, namely, VBASBP. The primary performance measurements to be taken are the average number of slots no matter how many transactions are mined under the assumption of the maximum number of slots per block as specified by n; the average waiting time per slot; and the throughput in terms of the average number of slots to be processed per time. The variable bulk arrival rate is assumed to vary linearly proportional to the size of the transactions in a multiple of λ (note that there is only a single stage of queue of waiting transactions (in terms of slots) assumed for simplicity instead of assuming two independent arrival rates of the transactions, one for the transaction pool and another for the waiting queue for the block posting, thereby assuming only a single bulk arrival rate per slot λ, which might be to some extent different in practice), and the static bulk service is assumed to take place when the number of slots in the mined transactions reaches at n, i.e., a bulk processing of multiple transactions in multiple slots for posting in a block.

The paper is organized as follows: related works to modeling and analysis of blockchain technology are reviewed in the following section as a preliminaries to the proposed approach as well; the proposed approach to modeling and analysis of the performance of the blockchain system is described in the following section in the context of a queueing system; a section will follow to demonstrate primary numerical simulations versus various blockchain-related parameters and results are shown; then conclusions and discussion are drawn in the last section.

2 Preliminaries and Literature Review

There have been a few works to address and investigate on various yet critical performance and dependability issues and problems as identified in various blockchain-based crypto computing systems.

Various hypothetical and theoretical designs [6,8,9] of a few crypto computing solutions have been developed in order to establish an engine for preliminary yet extensive parametric simulation, and some results have been demonstrated and validated through an isolated testing on Ethereum and Hyperledger open source-based prototypes [21,24,25]. As the ultimate quality of crypto computing will be determined by its likelihood to be performed as commanded or desired, referred to as the dependability, those hypothetical and theoretical models emphasized and centered around the dependability of each of those crypto solutions to accommodate such capabilities as the on/off-balanced crypto computing [24], the real-time computing [25], the slim-computing [21] and the hybrid computing. A dependability for each of those crypto solutions has been identified and defined along with various performance variables, and ultimately has provided a theoretical yet practical understanding of each crypto solution. A prototype, to demonstrate some of those crypto solutions and to validate their hypothetical and theoretical results, has been built by identifying and isolating the insertion points for necessary technology modification within Ethereum and Hyperledger open source to start out with and to ultimately realize a new core blockchain for optimal crypto computing.

The on/off-balanced chain [24] has been proposed in order to investigate on how to assure the dependability of a crypto system built across on and off the blockchain facilitated and coordinated by using the proposed adaptive checkpoint and rollback algorithm. The theoretical foundation of the proposed checkpoint and rollback algorithm is to characterize the variables affecting the dependability such as security, authenticity and reliability with respect to the rates of hit by any events of those issues, the rates to detect and diagnose, and then the rate to vote for a consensus whether to mark a checkpoint and trigger a rollback or not. Based on the variables characterization in a stochastic manner, the steady state probabilities and state transition probabilities has been derived in order to assure the ultimate effective dependability of each individual dependability variable (i.e., security, authenticity and reliability), then, finally to assure the dependability in a compound manner with each variable assigned a weight depending on the nature of the systems specifications. Based on the theoretical study [3,22], a prototype of the crypto system has been built to demonstrate the underlying architecture and operations and to justify the need for such system to take synergistic advantages from both on- and off-chains, with an experimental result of a benefit in gas fee [4], with respect to performance and dependability, which is the most exigently addressed issue today in blockchain systems especially in Ethereum network of blockchain. An astonishing result of gas fee saving has been demonstrated. It is expected that the crypto system will benefit more if more computationally intensive transactions are executed off-chain and vice versa. An isolated testing has been conducted for demonstration of the purpose on the proposed checkpoint and rollback algorithm on Ethereum open source.; the real-time chain [13,20,25] has been proposed in order to investigate on an approach how to design and realize a crypto computing (Ethereum blockchain-based [16]) under stringent real-time requirement. In order to evaluate the efficacy of the approach, a new analytical metric has been defined and developed to estimate the dependability, referred to as the block-dependability. The proposed block-dependability precisely models the probability for the mined transactions to be posted within the current, in other words, within the target block delay, further, namely, within the deadline required if their expected execution times are within the temporal range of the deadline. Various methods how to prioritize [17] and select transactions in the pending transaction pool in order to facilitate those transactions to be executed within their deadline requirements, such as the normal, random, sorted, and stratified [7], have been proposed. A set of performance variables, or parameters, such as the number of pending transactions in the pool, the average speed of the transactions, gas fees, deadlines as well as the number of miners, are identified and taken into the block-dependability analysis in order to reveal the influence of each variable on the block-dependability, versus each of those proposed prioritization and selection methods. Extensive parametric simulations are being conducted through an isolated testing on the Ethereum open source; the slim chain has been investigated in [21] in order to address and resolve the scalability issue of blockchain-based crypto computing in which it is required that every node participating in the computation carry a full load of the entire chain of blocks all the way from the genesis block. As evidenced from the preliminary results of the on/off-balanced chain [24], the performance and dependability [21] could be significantly improved by balancing the amount of computations across on- and off-chains, which depends on the type of computations such that on-chain execution is more suitable for more dependability-stringent or computation-intensive transactions, and off-chain for less dependability-stringent or less data-intensive computation. A smart balancing of computation across on/off-chain is expected to relieve the spatial and temporal overhead of managing the otherwise explosively growing size of the blockchain, as referred to as a slim chain in this paper (cf. the light chain is a technology to limit the temporal extent of the chain of blocks for synchronization). A novel theoretical model has been developed to evaluate the efficacy of the slim chain with respect to the dependability from a single transaction's standpoint in a stochastic manner. Further, the dependability will be traced with respect to whether the transaction of concern is to write off the chain or read from off the chain as the transactions writing intensively off the chain are more likely to be dependable than the ones reading intensively from off chain. Also, note that IPFS (Inter-Planetary File System) [11,16,23] is the off-chain storage system to be considered in the slim chain. A prototype with an isolated testing on the Ethereum open source is being built for validation purpose along with extensive parametric simulations; and the hybrid chain is being studied in order to investigate on a new blockchain network that is to be built particularly across private (i.e., permissioned) and main nets (i.e., permissionless). Note that the hybrid chain is distinguished from the earlier mentioned on/off-balanced chain such that the hybrid chain is concerned across two different types of nets (e.g., Hyperledger-based private net vs. Ethereum-based main net) while the on/off-balanced chain is concerned across on- and off-chain (e.g., Ethereum-based main net vs. cloud or IPFS). It is essential to build a dependable interface in between the private and the main nets if business-to-consumer (or vice versa) transactions are the primary transaction of interest. In the course of interfacing across the private and the main nets, dependability is to be considered as one of the most critical design factors in order to ensure that the private transactions stay within the private territory and publicized transactions stay public in the main net, and further in order to facilitate a seamless yet dependable migration of transactions across the border. In this context, the efficacy of the privacy of the private net side and the publicity of the main net side is addressed and modeled by tracing a transaction's stochastic process at a steady state. A prototype for an isolated testing across the Ethereum and Hypercubes open source for validation purpose is being built for validation purpose along with extensive parametric simulations.

However, the dependability models or performance models in [21,24,25] cannot readily and adequately address the queueing nature of the transactions flow and block posting other than that they will provide a sound theoretical foundation for dependability analysis in various blockchain contexts in line with the proposed VBASBS model. Ultimately, a synergistic model will be highly desired and pursued to model and assure both the dependability and the performance as two primary and concurrent variables of the blockchain system, thereby synergizing those dependability models and VBASBS model into a comprehensive and integrated analytical design tool.

3 Proposed Variable Bulk Arrival and Static Bulk Service (VBASBS) Model as A Baseline

In the proposed VBASBS model, an embedded Markovian single-server exponential queueing system (i.e., $M^{1,n}/M^{n}/1$)is considered without loss of generality, and the server (e.g., the server is the equivalence of the group of miners to select the transactions to be posted) serves the entire batch of customers (e.g., the customers are the equivalence of the transactions to posted in the block) in the queue (e.g., a queue is the equivalence of a block to be mined and posted) all at once at the same time. Whenever the server completes a service (e.g., a service is the equivalence a process of posting a block), it then purges the queue (e.g., the equivalence of the posting a block) and then serves the influx of new customers incoming. Note that it is assumed that the service takes place within a certain amount of time yet no transaction is assumed to arrive in the meantime. However, note that it is not unlikely to have new customers arrive if a significant amount of service time is assumed, from a practical point of consideration. It is assumed that the service time is exponential at $\frac{1}{\mu}$ when the server is serving the entire queue (e.g., equivalently, posting and purging the entire queue). Without loss of generality, it is assumed that customers arrive at an exponential rate of λ.

The underlying queueing process is assumed to take place with fixed-sized slots and the status of the queue is determined by the number of slots.

Given the assumptions as made above, the proposed VBASBP model employs an embedded Markovian queueing model and it defines the states as expressed in terms of the number of slots assigned to a block and it traces the normalized number of slots allocated for the transactions in steady state than the number of transactions whose size varies in the number of slots.

P_0 : the state in which there is no transaction (i.e., no slot) arrived in the queue as of yet for the posting in the block, currently.

P_n : the state in which there are n number of slots (i.e., which is the capacity of the queue, equivalently, the maximum number of slots set and voted by the miners or voters) arrived in the queue for the posting in the block, currently.

P_i : the state in which there are i number of slots (where $0 < i < n$) arrived in the queue for the posting in the block, currently.

The random variables employed to express the state transition rates are specified as follows.

λ: the rate for a slot of a transaction to arrive, and the rate for a transaction to arrive is determined by the number of slots allocated for the transaction in a prorated manner such that a transaction with a size of *j* number of slots arrives at the rate of *j*λ, without loss of generality and practicality as well.

μ: the rate for the slots of the transactions in the entire queue to be posted and purged. Notice that this is a single and unique state transition precisely from P_n back to P_0.

The balance equations for VBASBS are as follows.

$$(\lambda + 2\lambda + 3\lambda + \cdots + n\lambda)P_0 = \mu P_n$$

$$\left(\lambda \frac{n(n+1)}{2}\right) P_0 = \mu P_n$$

$$P_n = \frac{\lambda}{\mu}\frac{n(n+1)}{2} P_0 \quad (1)$$

$$(\lambda + 2\lambda + 3\lambda + \cdots + (n-i)\lambda)P_i$$

$$= \lambda P_{i-1} + 2\lambda P_{i-2} + 3\lambda P_{i-3} + \cdots + i\lambda P_0 \quad (2)$$

$$P_1 = q_1 P_0 \quad (3)$$

From Equations (1), (2) and (3), P_i, $0 < i < n$, can be expressed in terms of P_0 as follows.

$$P_2 = q_2(q_1 + 2)P_0 \quad (4)$$

$$P_3 = q_3(q_2(q_1 + 2) + 2q_1 + 3)P_0 \quad (5)$$

$$P_4 = q_4(q_3(q_2(q_1 + 2) + 2q_1 + 3) + 2q_2(q_1 + 2) + 3q_1 + 4)P_0 \quad (6)$$

$$P_4 = P_0 q_4(1(q_3 q_2 q_1 + q_3 q_2 2 + q_3 3) + 2(q_2 q_1 + q_2 2) + 3(q_1) + 4) \quad (7)$$

$$P_0 + P_1 + P_2 + \cdots + P_n = 1 \quad (8)$$

P_i can be generalized and expressed as follows.

$$P_i = q_i P_0 \left(\sum_{j=1}^{i} j \left(\sum_{k=1}^{i-j} k \left(\prod_{l=k}^{i-j} q_l \right) \right) + i \right) \quad (9)$$

Where,

$0 < i < n$

$$q_i = \frac{2}{(n-i)(n-i+1)}$$

$$q_l = \frac{2}{(n-l)(n-l+1)}$$

$$\prod_{l=1}^{k-1} q_l = \prod_{l=1}^{k-1} \frac{2}{(n-l)(n-l+1)}$$

$$= \frac{2^{k-1}}{\frac{(n!)^2}{(n-0)(n-k+1)}} = \frac{2^{k-1}(n)(n-k+1)}{(n!)^2} \quad (10)$$

Then, it can be expressed in Equation (11) as follows.

$$\sum_{k=1}^{i-1} \left(\prod_{l=1}^{k-1} q_l \right) k$$

$$= \sum_{k=1}^{i-1} \left(\frac{2^{k-1}(n)(n-k+1)}{(n!)^2} \right) k \quad (11)$$

$$= \frac{n}{(n!)^2} \left((n+1)\left(\sum_{m=0}^{i-2} 2^m m\right) - \left(\sum_{m=0}^{i-2} 2^m m^2\right) + \left(n \frac{2(2^{i-2}-1)}{(2-1)} \right) \right) \quad (12)$$

In Equation (12), there are two sigma forms to solve as follows.

The first sigma form $\sum_{m=0}^{i-2} 2^m m$ can be expressed as follows.

$$\sum_{m=0}^{i-2} 2^m m = 2^0 0 + 2^1 1 + 2^2 2 + \cdots + 2^{i-3}(i-3) + 2^{i-2}(i-2)$$
$$= 2^{i-1}(i-3) + 2 \tag{13}$$

The following Equation (14) rewrites equation (12) by taking Equation (13) in place.

$$\sum_{k=1}^{i-1}\left(\prod_{l=1}^{k-1} q_l\right)k$$
$$= \frac{n}{(n!)^2}\left((n+1)\left(2^{i-1}(i-3)+2\right) - \left(\sum_{m=0}^{i-2} 2^m m^2\right) + \left(n\frac{2\left(2^{i-2}-1\right)}{(2-1)}\right)\right) \tag{14}$$

In Equation (14), another sigma form $\sum_{m=0}^{i-2} 2^m m^2$ can be expressed as in Equation (15).

$$\sum_{m=0}^{i-2} 2^m m^2 = 2^{(i-1)}(i^2 - 6i + 11) - 6 \tag{15}$$

Then, the following is obtained.

$$\sum_{k=1}^{i-1}\left(\prod_{l=1}^{k-1} q_l\right)k$$
$$= \frac{n}{(n!)^2}\sum_{k=1}^{i-1}\left(2^{k-1}(n-k+1)\right)k$$
$$= \frac{n}{(n!)^2}\left((n+1)\left(2^{i-1}(i-3)+2\right) + \left(n\frac{2\left(2^{i-2}-1\right)}{(2-1)}\right) + \left((-1)\left(2^{(i-1)}(i^2-6i+11)-6\right)\right)\right) \tag{16}$$

Then, lastly, the following term can be solved as follows as shown in Equation (17).

$$\sum_{j=1}^{i} j\left(\sum_{k=1}^{i-j}\left(\prod_{l=k}^{i-j} q_l\right)k\right)$$
$$= \sum_{j=1}^{i} j\left[\frac{n}{(n!)^2}\left((n+1)\left(2^{i-1}(i-3)+2\right) + \left(n\frac{2\left(2^{i-2}-1\right)}{(2-1)}\right) + \left((-1)\left(2^{(i-1)}(i^2-6i+11)-6\right)\right)\right)\right]$$

$$= \sum_{j=1}^{i} j\left(\frac{n}{(n!)^2}\left(\left(2^{i-1}in - 2^{i-1}3n + 2n + 2^{i-1}i - 2^{i-1}3 + 2\right) + \left(2^{i-1}n - 2n\right) + \left(-2^{(i-1)}i^2 + 2^{(i-1)}6i - 2^{(i-1)}11 + 6\right)\right)\right) \tag{17}$$

Taking the Stirling's Approximation as shown in (18), Equation (17) can be rewritten as shown in Equation (19).

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \tag{18}$$

$$\sum_{j=1}^{i} j\left(\frac{n}{\left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right)^2}\left(\left(2^{i-1}in - 2^{i-1}3n + 2n + 2^{i-1}i - 2^{i-1}3 + 2\right) + \left(2^{i-1}n - 2n\right) + \left(-2^{(i-1)}i^2 + 2^{(i-1)}6i - 2^{(i-1)}11 + 6\right)\right)\right) \tag{19}$$

Where,

$$\sum_{j=1}^{i} j = \left(\frac{i(i+1)}{2}\right)$$

From the Equation (19), it rewrites the equation as follows.

$$\sum_{j=1}^{i} j\left(\sum_{k=1}^{i-j}\left(\prod_{l=k}^{i-j} q_l\right)k\right) + i$$
$$= \left(\frac{i(i+1)}{2}\right)\left(\frac{n}{\left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right)^2}\right)\left(2^{i-1}(in - 2n + 7i - 14 - i^2) + 8\right)$$

Now, P_i can be expressed as follows.

$$P_i = q_i P_0\left(\left(\frac{i(i+1)}{2}\right)\left(\frac{n}{\left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right)^2}\right)\left(2^{i-1}(in - 2n + 7i - 14 - i^2) + 8\right) + i\right) \tag{20}$$

where,

$0 < i < n$

$$q_i = \frac{2}{(n-i)(n-i+1)}$$

$$P_i = \frac{2}{(n-i)(n-i+1)} P_0 \left(\left(\frac{i(i+1)}{2}\right) \frac{n}{\left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right)^2} \left(2^{i-1}(in - 2n + 7i - 14 - i^2) + 8\right) + i \right)$$

From Equation (8), $P_0 + \sum_{i=1}^{n-1} P_i + P_n = 1$ and P_0 can be solved as shown in Equations (21), (22) and (23).

$$P_0 \left(1 + \left(q_1 \left(\sum_{j=1}^{1} j + 1 \right) \right) + \left(q_2 \left(\sum_{j=1}^{2} j \left(\sum_{k=1}^{i-1} \left(\prod_{l=1}^{k-1} q_l \right) k \right) + 2 \right) \right) + \cdots + \left(q_{n-1} \left(\sum_{j=1}^{n-1} j \left(\sum_{k=1}^{n-2} \left(\prod_{l=1}^{k-1} q_l \right) k \right) + 2 \right) \right) + \frac{\lambda}{\mu} \left(\frac{n(n+1)}{2} \right) \right) = 1 \tag{21}$$

$$P_0 \left(1 + \sum_{i=1}^{n-1} \left(q_i \left(\sum_{j=1}^{i} j \left(\sum_{k=1}^{i-1} \left(\prod_{l=1}^{k-1} q_l \right) k \right) + i \right) \right) + \frac{\lambda}{\mu} \left(\frac{n(n+1)}{2} \right) \right) = 1 \tag{22}$$

$$P_0 = \frac{1}{\left(1 + \sum_{i=1}^{n-1} \left(q_i \left(\sum_{j=1}^{i} j \left(\sum_{k=1}^{i-1} \left(\prod_{l=1}^{k-1} q_l \right) k \right) + i \right) \right) + \frac{\lambda}{\mu} \left(\frac{n(n+1)}{2} \right) \right)} \tag{23}$$

Where,

$$P_n = \frac{\lambda}{\mu} \left(\frac{n(n+1)}{2} \right) P_0$$

$$P_0 \sum_{i=1}^{n-1} P_i = P_0 \sum_{i=1}^{n-1} \left(q_i \left[\sum_{j=1}^{i} j \left[\sum_{k=1}^{i-1} \left[\prod_{l=1}^{k-1} q_l \right] k \right] + i \right] \right)$$

From Equations (1), (8) and (9), all the remaining solutions for the balance equations for VBASBS (i.e., P_n from Equation (1) and P_i from Equation (8)) can be obtained).

The followings are a few baseline performance measurements of primary interests in VBASBS.

L_Q : the average number of customers (i.e., equivalently the average number of transactions) in the queue (i.e., the block currently being mined).

$$L_Q = \sum_{i=0}^{n} i P_i \tag{24}$$

Where,

$$\sum_{i=0}^{n} i P_i = \sum_{i=0}^{n} i \left(q_i P_0 \left(\sum_{j=1}^{i} j \left(\sum_{k=1}^{i-j} \left(\prod_{l=k}^{i-j} q_l \right) k \right) + i \right) \right)$$

W_Q : the average amount of time a customer (i.e., equivalently, a transaction) in the queue (i.e., the block currently being mined).

$$W_Q = \frac{L_Q}{\lambda} \tag{25}$$

W : the average amount of time a customer (i.e., equivalently, a transaction) in the system (i.e., the transaction pool in the blockchain).

$$W = W_Q + \frac{1}{\mu} \tag{26}$$

L : the average number of customers (i.e., equivalently, the average number of transactions) in the system (i.e., the transaction pool in the blockchain).

$$L = \lambda W \tag{27}$$

4 Numerical Analysis

The primary objective of the simulation is to reveal various preliminary performance of the blockchain system of interest such as L_Q, W_Q, W and L versus n (i.e., size of a block), λ (i.e., transaction arrival rate or speed), and $\frac{1}{\mu}$ (i.e., block posting time). Note that the block posting time, $\frac{1}{\mu}$, is fixed at 15 seconds (i.e., $\mu = 0.0667$) in order to conduct the analysis under a practical parametric condition as typical block delay is known to be about 15 seconds in Ethereum. Note that this section is not to conduct a simulation to reveal against a particular blockchain system but to demonstrate a valid and baseline simulation model in the context of a queueing system.

The following graph plots L_Q (i.e., from Equation (24)) versus n for given pairs of λ and μ.

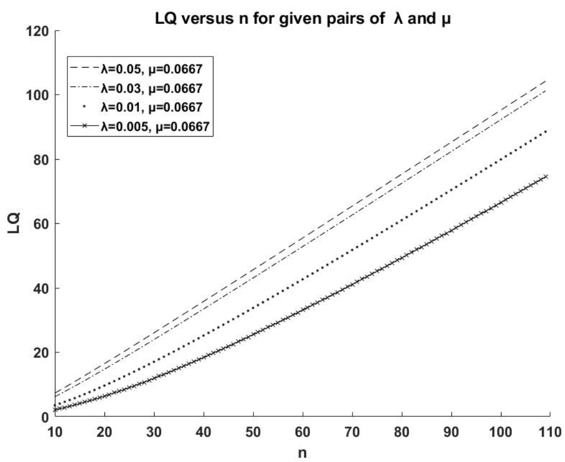

Figure 1: L_Q versus n for given pairs of λ and μ.

Figure 1 demonstrates the validity of the proposed VBASBS model. Under the assumptions on the arrival rates and service times, it shows quite a monotonically increasing trends of the average number of slots in a block as a representation of the number of transactions in a normalized manner.

Notice that the average number of slots ultimately represents the population in the block on average no matter how many transactions they belong to. In fact, each state P_i, $0 \leq i \leq n$, represents a transaction with i number of slots and its steady state probability represents the normalized likelihood of the number of transactions of the size of i. Thus, it is claimed that $L_Q = \sum_{i=0}^{n} iP_i$, has a valid representation of the average number of transactions in terms of the average number of slots. In fact, tracking the number of slots facilitates the process of tracking the number of transactions which otherwise would be complicated to track due to the variability of the sizes of the transactions. Also, note that the arrival rates of lots (cf. transactions) at the transaction pool (i.e., L) and at the block (i.e., L_Q) are assumed to be identical in this simulation, for simplicity purpose, which might have been assumed differently if mandated to do so for practicality purpose.

The following observations are drawn from the simulation results in Figure 1 : as the size of the block increases, the average number of slots in the mined transactions to be posted in the block increases slower as the arrival rate (i.e., λ) decreases as expected and intuitively as well; the unpopulated portion on average (i.e., $n - L_Q$) is narrowing as the size of the block grows, which is to do with the level of the arrival rate such that the higher the arrival rate goes, the narrower the unpopulated portion turns; further, notice that L_Q grows monotonically without a sign of saturation and it is speculated that the monotonicity is expected as the block is modeled to be purged as soon as the number of slots in the mined transactions to be posted on the block hits n, which does not lose any generality from the stand point of a queue of mined transactions to be posted on a block as is the underlying assumption of the proposed VBASBS model; and lastly, notice that as the arrival rate grows higher, the growth rate of L_Q slows.

Note that an assumption is made for simplicity such that the block is to be purged back to 0 slot status exactly when it hits n without any consideration of non-full block posting, yet in practice, it is not impossible to have a non-full block to be posted when, for instance, a huge transaction is mined and might span across two or more number of blocks, which is left in this work as a future work to be addressed and resolved.

The following graph plots W_Q (i.e., from Equation (26)) versus n for given pairs of λ and μ.

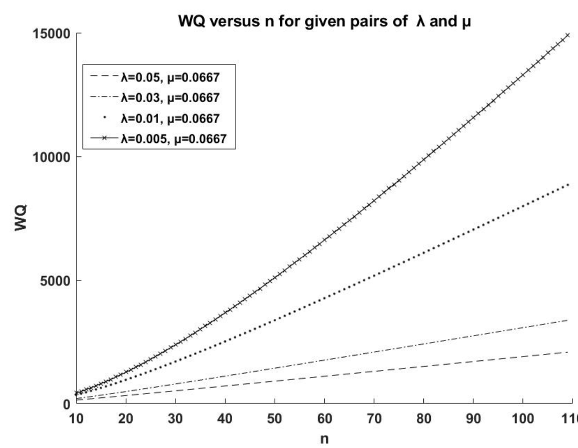

Figure 2: W_Q versus n for given pairs of λ and μ.

As the proposed VBASBS model has been validated in the simulation as shown in Figure 1 without loss of intuition, Figure 2 also demonstrates the average waiting time of the mined transactions (or slots) for the posting on the block is proportional to $\frac{L_Q}{\lambda}$ in a monotonic manner. It is observed that for a given size of the block, the waiting time picks up as the arrival rate λ decreases; and the growth rate of the waiting time steepens as the arrival rate λ decreases as well.

The following graph plots W (i.e., from Equation (25)) versus n for given pairs of λ and μ.

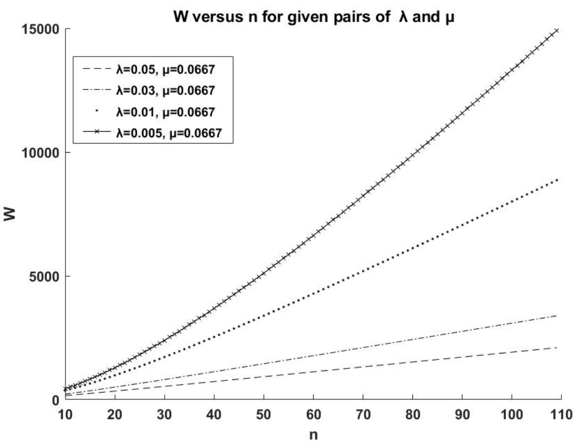

Figure 3: W versus n for given pairs of λ and μ.

Figure 3 demonstrates the waiting time of the pending transactions (in terms of slots) in the transactions pool for the mining selection for the block, is determined by $W_Q + \frac{1}{\mu}$ and it is observed to be just a matter as much as $\frac{1}{\mu}$ added to W_Q resulting in a slight increase in time.

The following graph plots L (i.e., from Equation (27)) versus n for given pairs of λ and μ.

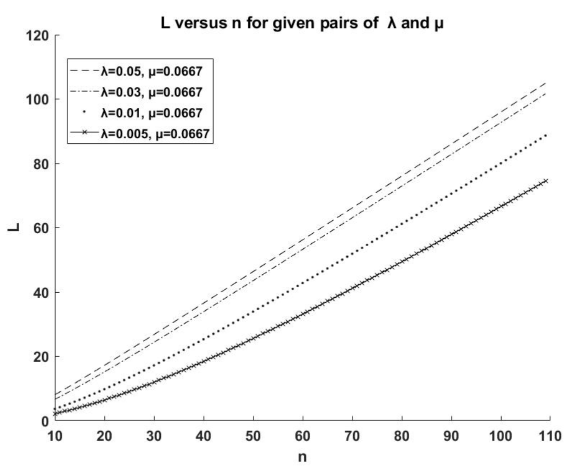

Figure 4: L versus n for given pairs of λ and μ.

Figure 4 demonstrates the average number of transactions in the transaction pool waiting for mining selection and is determined by $L = \lambda W$. It is observed that at the given arrival rates of λ, L is set to be slightly higher than L_Q at a given n, which is speculated such that the identical arrival rates of λ as assumed accounts for it.

The throughput per block in VBASBS model can be obtained as follows.

$$\gamma = \mu P_n = \mu \frac{\lambda}{\mu} \frac{n(n+1)}{2} P_0 = \lambda \frac{n(n+1)}{2} P_0 \tag{28}$$

The following graph plots γ versus n for given pairs of λ and μ. In Figure 5, it is observed that as the size of the block (i.e., n) grows, γ increases; γ is independent of μ and solely affected by λ; and the higher λ picks up, the higher throughput achieved for a given size of the block, n.

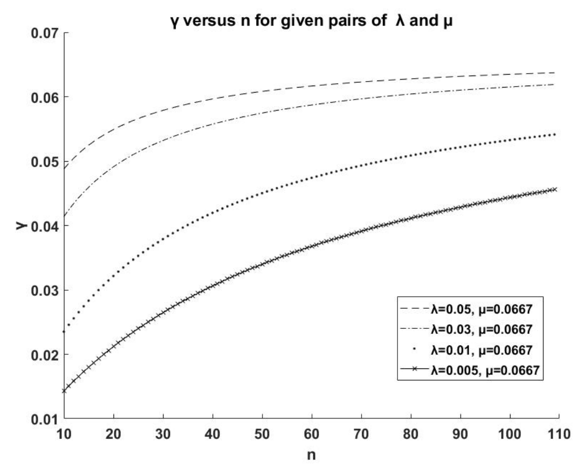

Figure 5: γ versus n for given pairs of λ and μ.

5 Conclusion and Discussion

This paper has presented an embedded Markovian queueing model of the $M^{1,n}/M^n/1$ type in order to establish a theoretical foundation to design a blockchain-based system with focus on the stochastic behavior of the mined transactions waiting to be posted for the block delay as the bulk synchronization point as is being exercised by Ethereum. The model assumes variable bulk arrivals of transactions in Poisson distribution, i.e., $M^{1,n}$, where n the number of slots across all the mined transactions and static bulk services of transactions in exponential time, i.e., M^n, for posting in the current block, namely, VBASBP. The primary performance measurements to be taken are the average number of slots no matter how many transactions are mined under the assumption of the maximum number of slots per block as specified by n; the average waiting time per slot; and the throughput in terms of the average number of slots to be processed per time. The variable bulk arrival rate is assumed to vary linearly proportional to the size of the transactions in a multiple of λ and it is assumed that there is only a single stage of queue of waiting transactions (in terms of slots) assumed for simplicity instead of assuming two independent arrival rates of the transactions, one for the transaction pool and another for the waiting queue for the block posting, thereby assuming only a single bulk arrival rate per slot λ, which might be to some extent different in practice. The static bulk service is assumed to take place when the number of slots in the mined transactions reaches n, i.e., a bulk processing of multiple transactions in multiple slots for posting in a block. Numerical simulations have been conducted to demonstrate the efficacy of the model with respect to the average number of slots waiting in the queue as is representation of the primary variable in VBASBS model, the average waiting time of the transactions as represented by the normalized size in terms of the slots, then the average waiting time of the transactions and the average number of slots in the system (i.e., the transaction pool). The simulation results confirm as expected to be able to claim the validity and efficacy of VBASBS model.

REFERENCES

[1] Vladimir Soloviev, "Fintech Ecosystem in Russia", *2018 Eleventh International Conference "Management of large-scale system development" (MLSD).* DOI: 10.1109/MLSD.2018.8551808

[2] Nir Kshetri, and Jeffrey Voas, "Blockchain in Developing Countries", *IT Professional* (Volume: 20, Issue: 2, Mar./Apr. 2018). DOI: 10.1109/MITP.2018.021921645

[3] Guido Perboli, Stefano Musso, and Mariangela Rosano, "Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases", *IEEE Access,* pp(99):1-1 · October 2018

[4] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", 2008, [online] Available: https://bitcoin.org/bitcoin.pdf.

[5] Chris Dannen. 2017. *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners* (1st. ed.). Apress

[6] Diego Vegros, and Jaime Saenz, 2010. *Peer-To-Peer Networks and Internet Policies*, Nova Science Publishers, Inc., NY, USA

[7] Vitalik Buterin, "A Next Generation Smart Contract & Decentralized Application Platform", *White Paper,* 2014

[8] Israel Koren and C. Mani Krishna. 2007. *Fault-Tolerant Systems*. San Francisco, CA: Morgan Kaufmann Publishers

[9] Pouria Pirzadeh, Michael Carey, and Till Westmann, "A performance study of big data analytics platforms" *2017 IEEE International Conference on Big Data (Big Data).* DOI: 10.1109/BigData.2017.8258260

[10] Morgen E. Peck. "Blockchains: How They Work and Why They'll Change the World", *IEEE Spectrum* (Volume: 54, Issue: 10, October 2017). DOI: 10.1109/MSPEC.2017.8048836

[11] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler, "The Hadoop Distributed File System", *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST).* DOI: 10.1109/MSST.2010.5496972

[12] Hemang Subramanian. "Decentralized Blockchain-Based Electronic Marketplaces", *Communications of the ACM*, Volume 61, Number 1, December 2017. https://doi.org/10.1145/3158333

[13] Dejan Vujii, Dijana Jagodi, and Sinia Rani, "Blockchain technology, bitcoin, and Ethereum: A brief overview", *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH).* DOI: 10.1109/INFOTEH.2018.834554

[14] Sara Rouhani, and Ralph Deters, "Performance Analysis of Ethereum Transactions in private blockchain" *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS).* DOI: 10.1109/ICSESS.2017.8342866

[15] Mrs. Anamika Chauhan, Om Prakash Malviya, Madhav Verma, and Tejinder Singh Mor, "Blockchain and Scalability", *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C).* DOI: 10.1109/QRS-C.2018.00034

[16] Gavin Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger", 2014. *Ethereum Project Yellow Paper*, http://gavwood.com/paper.pdf

[17] Ingo Weber, Vincent Gramoli, Alex Ponomarev, Mark Staples, Ralph Holz, An Binh Tran, and Paul Rimba, "On Availability for Blockchain-Based Systems", *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS).* DOI: 10.1109/SRDS.2017.15

[18] Morgen E. Peck (September 2017) "Blockchains: How They Work and Why They'll Change the World", *IEEE Spectrum* 54(10):26-35 · October 2017. DOI: 10.1109/MSPEC.2017.8048836

[19] Federico Lombardi, Leonardo Aniello, Stefano De Angelis, Andrea Margheri, and Vladimiro Sassone, "A Blockchain-based Infrastructure for Reliable and Cost-effective IoT-aided Smart Grids", *Living in the Internet of Things: Cybersecurity of the IoT – 2018.* DOI: 10.1049/cp.2018.0042

[20] Qi Zhang, Petr Novotný, Salman Baset, Donna N. Dillenberger, Artem Barger and Yacov Manevich, "LedgerGuard: Improving Blockchain Ledger Dependability", *ICBC* (2018).

[21] Jongho Seol, Abhilash Kancharla, Nicole Park, Nohpill Park, and Indy Nohjin Park, "The Dependability of Crypto Linked Off-chain File Systems in Backend Blockchain Analytics Engine", *International Journal of Networked and Distributed Computing* Vol 6, Dec 2018.

[22] How to Check the Reliability of a Blockchain Project, CoinJournal Press, [Online] https://coinjournal.net/sponsored-story-how-to-check-the-reliability-of-a-blockchain-project/

[23] Juan Benet, "IPFS – Content addressed, versioned, P2P file system", 2014. arXiv:1407.3561v1

[24] Abhilash Kancharla and N. Park, ''A Realtime Crypto Computing and Block-Dependability'', *IEEE SC2 2019*

[25] Abhilash Kancharla, Indy Park, Nicole Park and N. Park, ''Dependable Industrial Crypto Computing'', *IEEE ISIE 2019.* DOI: 10.1109/ISIE.2019.8781245

[26] Keke Gai, Yulu Wu, Liehuang Zhu, Zijian Zhang, and Meikang Qiu, "Differential Privacy-Based Blockchain for Industrial Internet-of-Things", *IEEE Transactions on Industrial Informatics* (Volume: 16, Issue: 6, June 2020). DOI: 10.1109/TII.2019.2948094

[27] Keke Gai, Yulu Wu, Liehuang Zhu, Meikang Qiu, and Meng Shen, ''Privacy-Preserving Energy Trading Using Consortium Blockchain in Smart Grid", *IEEE Transactions on Industrial Informatics* (Volume: 15, Issue: 6, June 2019). DOI: 10.1109/TII.2019.2893433

[28] Keke Gai, Yulu Wu, Liehuang Zhu, Lei Xu, and Yan Zhang, "Permissioned Blockchain and Edge Computing Empowered Privacy-Preserving Smart Grid Networks'', *IEEE Internet of Things Journal* (Vomue:6, Issue:5, Oct 2019). DOI: 10.1109/JIOT.2019.2904303

APPENDIX

The following MATLAB code computes L, W, W_Q, and L_Q.

```
iterations =100;
L = zeros(1,iterations)
W = zeros(1,iterations)
WQ = zeros(1,iterations)
LQ = zeros(1,iterations)
Gamma = zeros(1, iterations)
n=10;
Psum = 0;
lambda = 0.05; % 0.005 -x, 0.01 ., 0.03-., 0.05 --,
mu = 1/15; % set 1/15

for iterations = 1:iterations
  Q = zeros(1,n);
  a = zeros(1,n);
  P = zeros(1,n);
  temp = ones(1,n);
  for i = 1 : n-1
    q(i) = 2/((n-i)*(i+1));
    temp(i) = 2^(i-1)*(i*n-2*n+7*i-14-i^2)+8;
    a(i)= i*(i+1)/2)*(n/(sqrt(2*pi*n)*(n/exp(1))^n)^2)*temp(i)+i;
    Q(i) = a(i) * (2/((n-i)*(n-i+1)));
  end
    Psum = sum(Q)+1+(lambda/mu)*((n*(n+1))/2);
    Pzero = 1/Psum;

    for i = 1:n-1
      P(i) = Q(i)*Pzero ;
    end
    P(n) = (lambda/mu)*((n*(n+1))/2)*Pzero
```

```
 Psum = sum(P)+Pzero;
for i = 1:n
  L(iterations) = i * P(i) ;
end
for i = 1:n
  W(iterations) = L(iterations) / lambda;
end
for i = 1:n
  WQ(iterations) = W(iterations) - 1/mu;
end
for i = 1:n
  LQ(iterations) = WQ(iterations) * lambda;
end
for i = 1:n
   Gamma(iterations) = P(n)*mu;
end
n = n + 1;
```