An Asynchronous Chain and A Variable Bulk Arrival and Asynchronous Bulk Service Model

Jongho Seol*
Department of Information Technology, Middle Georgia State University
jongho.seol@mga.edu

Nohpill Park
Department of Computer Science, Oklahoma State University
npark@cs.okstate.edu

ABSTRACT

This paper proposes a, namely, asynchronous chain and presents a Variable Bulk Arrival and Asynchronous Bulk Service (VBAABS) model of the $M^{1,n}/M^{1,j,i,n}/1$ type in order to provide a quantitative method to design an asynchronous chain as a basis in its initial theoretical design stage. Note that the proposed chain is precisely and namely asynchronous along with adaptive-sized blocks in a proactive manner versus the conventional chain that controls the block posting in a strictly synchronous manner to the fixed-sized blocks. The model of the $M^{1,n}/M^{1,n}/1$ type (i.e., VBAVBS) is considered as the theoretical and quantitative baseline model for the proposed model, namely, an adaptive chain model, yet with a rather reactively dynamic size of blocks, while the proposed asynchronous model carries a proactively dynamic (or adaptive) size of blocks. The proposed asynchronous chain model assumes variable bulk arrivals of transactions in Poisson distribution, i.e., $M^{1,n}$, where n represents the number of slots across all the mined transactions, and variable asynchronous services of transactions, each of which applies to a block potentially of different capacity in terms of the number of slots in it, in exponential time, i.e., $M^{1,n}$, for being posted in the current block, namely, VBAABS. The major quantitative distinction between the adaptive model, i.e., VBAVBS, and the asynchronous model, i.e., VBAABS, is that in VBAVBS, every state P_i , where $0 < i \leq n$, transitions back into P_0 while in VBAABS, every state P_i , where $0 < i \leq n$, transitions back into P_j , where $0 < j < i$, as well as back into P_0, and is the state in which the transaction up in execution is to be discarded, in other words, that the current block capacity is in excess of the required capacity for the transaction in execution to result in a potentially excessive delay as much as $i - j$ than an otherwise ordinarily fully synchronous block posting delay that is as much as i. VBAABS will reveal the performance advantages of the asynchronous chain versus the baseline chain, i.e., VBASBS [30] and the adaptive chain, i.e., VBAVBS, with respect to the average time for a slot to wait in the block and the average spatial requirement by the slots in the block, in a quantitative manner. Extensive numerical simulations are conducted on Matlab. Further, for feasibility validation purpose, an asynchronous chain algorithm will be developed and implemented by redesigning the Ethereum open source and analyzed.

CCS CONCEPTS

• **Computer systems organization**; • **Architectures**; • **Distributed Architectures**;

KEYWORDS

Blockchain, Asynchronous chain, Adaptive chain, Baseline chain, Queueing model

ACM Reference Format:
Jongho Seol* and Nohpill Park. 2023. An Asynchronous Chain and A Variable Bulk Arrival and Asynchronous Bulk Service Model. In *The 5th ACM International Symposium on Blockchain and Secure Critical Infrastructure (BSCI '23), July 10–14, 2023, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3594556.3594628

1 INTRODUCTION

Blockchain technology [1–3, 8, 9] is a decentralized system that uses a web3 network, which is a peer-to-peer communication based on a distributed information network where all nodes store mutual data, as a reliable trust system that does not require mutual verification. Due to this mutually reliable network environment, there is an issue of block delay time until a contract is used to transmit or store data, i.e., as a transaction deploying, a mining process that is posting in a block as a confirmed transaction. The point is that the system used as a server-client based web2 network is relatively faster than a decentralized distributed network. Currently, decentralized applications in many industries are such as IoT [16–19] and MEC [24], and more notably, blockchain technology for digital currency is used in central banks. However, the issue of transaction delay time due to block processing speed must be resolved [15, 26]. While it has been reported that the blockchain consensus algorithm [2, 3, 26, 27] makes it a trusted system, it causes block time delay which leads to lower throughput of the system and increases waiting time. In this context, block size adjusting and its requirement, i.e., gas limit, will improve blockchain performance [20–23, 28]. The gas limit of the block affects the block time as well as the blockchain scalability [10–14, 29] and dependability [19].

In this paper, with the baseline models of VBASBS (i.e., a $M^{1,n}/M^{n}/1$ type) [30] and the adaptive model VBAVBS (i.e., a $M^{1,n}/M^{1,n}/1$type), the primary interest is to develop an embedded Markovian queueing model of the $M^{1,n}/M^{1,\mathrm{i},\mathrm{j},n}/1$ type in order to establish a quantitative foundation to design a blockchain-based system with a focus on the stochastic behavior of the mined transactions waiting to be posted for the block time as potentially purging at every state, which is possibly being from any state, P_i $(0 < i \leq n)$

BSCI '23, July 10–14, 2023, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0198-6/23/07…$15.00
https://doi.org/10.1145/3594556.3594628

back into the state, P_0. Unlike the existing baseline model, this model reports a case in which the state cannot be purged due to either block time delay or block size limit, and then the state transition position is changed by discarding which is different from the adaptive model. As in both the baseline models of VBASBS [30] and the adaptive model of VBAVBS, the proposed model assumes variable bulk arrivals of transactions in Poisson distribution, $M^{1,n}$, where n the number of slots across all the mined transactions, but, variable bulk service of transactions in exponential time, $M^{1,n}$, for posting into the current block and discarding when the block delaying or gas limit of block size at any state in a slot, VBAABS. The primary performance measurements are to be taken in comparison with the baseline model and the adaptive model, such as 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 a 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 variable asynchronous [4–7] service is assumed to take place when the number of slots in the mined transactions reaches any state, 1,2,. . .,$n-1$,n, i.e., a bulk processing of single or multiple transaction(s) in single or multiple slot(s) for posting in a block and the same manner single processing of single transaction in the single slot for discarding from at any state i into j ($j < i$) a slot.

The paper is organized as follows: a baseline model (i.e., VBASBS [30]) and an adaptive model (i.e., VBAVBS) are reviewed in the following section 2 as the preliminaries to the proposed approach; the new proposed VBAABS model is presented in the following section 3; a section 4 will follow to demonstrate numerical simulations versus various blockchain related parameters and results are shown to demonstrate the performance benefits of the proposed VBAABS; the implementation in a private blockchain network is presented and compared to the VBAVBS model to validate the efficacy and benefits of the model in the following section 5. Conclusion is in the final section 6.

2 PRELIMINARIES AND LITERATURE REVIEW

In the VBASBS model [30] and VBAVBS model, which are the basis for the proposed VBAABS model in this paper, the embedded Markovian models (i.e., $M^{1,n}/M^{n}/1$ and $M^{1,n}/M^{1,\ n}/1$ respectively) have been proposed. Figure 1 shows the all the state transitions diagram of the baseline chain model and Figure 2 shows all the states transitions diagram of the adaptive chain model respectively.

The VBASBS [30] and VBAVBS models define the states commonly as follows.

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, and is onheld in the proposed VBAVBS model in this paper.

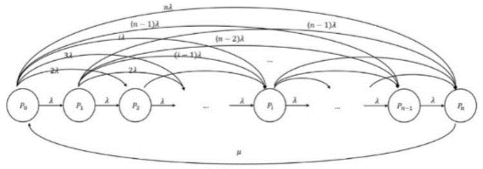

Figure 1: State transition diagram of the baseline chain model, VBASBS [30]

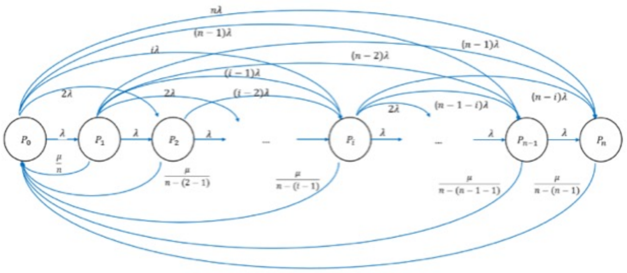

Figure 2: State transition diagram of the adaptive chain model, VBAVBS

P_n : the state in which there is 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, and is onheld in the proposed VBAVBS model in this paper.

P_i : the state in which there is i number of slots (where $0 < i < n$) arrived in the queue for the posting in the block, currently, and is onheld in the proposed VBAVBS model in this paper.

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 $j\lambda$, without loss of generality and practicality as well, which will be onheld in this paper as well.

μ: the rate for the slots of the transactions in the entire block to be posted and purged. Notice that this is a single and unique state transition precisely from P_n back to P_0, which will be onheld in this paper as well regardless of the potentially varying number of slots in a block in a normalized manner.

The state for the adaptive chain model, VBAVBS, is identical to and only the state transition probabilities differ from the ones in the baseline model, VBASBS [30].

μ: the rate for the slots of the transactions in the entire queue to be purged and posted into a block. It is only noted in the adaptive model that it is an every state transition precisely from P_n back to P_0, P_{n-1} back to P_0,. . ., P_2 back to P_0, and P_1 back to P_0 with $\frac{\mu}{n-(n-1)}$, $\frac{\mu}{n-((n-1)-1)}$, . . . , $\frac{\mu}{n-1}$, $\frac{\mu}{n}$ respectively.

The balance equations for VBASBS [30] can be generalized as follows.

$$\lambda\left(\frac{(n-i)(n-i+1)}{2}+\frac{\mu}{\lambda(n-i+1)}\right)P_i$$
$$=\lambda P_{i-1}+2\lambda P_{i-2}+3\lambda P_{i-3}+\ldots+(i-1)\lambda \mathrm{P}_{\mathrm{i}-(\mathrm{i}-1)}+i\lambda \mathrm{P}_{\mathrm{i}-\mathrm{i}}$$

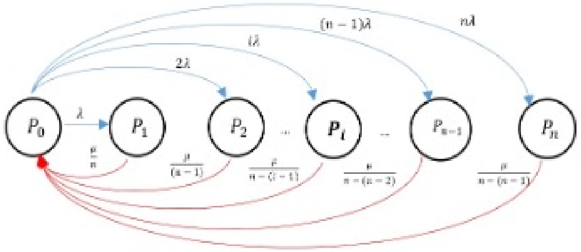

Figure 3: State transition diagram of the asynchronous chain model, VBAABS

Where,

$$0 < i \leq n,$$

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

3 PROPOSED VARIABLE BULK ARRIVAL AND ASYNCHRONOUS BULK SERVICE (VBAABS) MODEL FOR ASYNCHRONOUS CHAIN

The state for the asynchronous chain is identical to and only the state transition probabilities differ from the ones in VBAVAS. The random variables employed to employed to express the state transition rates and defined as follows. Figure 3 shows all states and their transition diagrams of the asynchronous chain model.

The VBAABS model uses the definitions of the states commonly used in both VBASBS [30] and VBAVBS models in the previous Section 2. The VBAABS model proposed utilizes the definitions of states commonly used in the VBASBS [30] and VBAVBS models. Among them, lambda, λ, is newly defined as follows and applied to this model.

λ: refer to the definition in the previous section as in section 2.

$\frac{\mu}{i-(i-1-j)}$ (μ transition definition): $P_i \rightarrow P_j$

1) the rate for the slots of the transactions in the entire queue to be purged and posted into a block. Note that this is an every state transition precisely from P_n back to P_0, P_i back to P_0, P_2 back to P_0, and P_1 back to P_0, when $i-(i-1-j)=1$ only.

2) the rate for the specific slots of the transactions in the queue to be discarded into another slot down when the mining process takes longer or a current block size limit is exceeded. Note that this is every state from P_i, where $1 < i \leq n$, transition back to P_j, where $0 < j < i$. After discarding the state, the state will be purged or discarded by the condition of transition definition.

State transition definition of $P_i \longrightarrow P_j \; with \; \mu$:

$$P_i \longrightarrow P_j : \; \frac{\mu}{i-(i-1-j)}$$

Where,

$j < i \; \leq n$ and $j \geq 0$

n: max number of slots

μ: purging at $j=0$ and discarding at $0 < j < i$

The following each transition of purging and/or discarding from P_i to P_j with μ.

$$i = 1,$$

$$P_1 \longrightarrow P_0 : \; \frac{\mu}{1-(1-1-0)} = \frac{\mu}{1} \;\; (purging)$$

$$i = 2,$$

$$P_2 \longrightarrow P_1 : \; \frac{\mu}{2-(2-1-1)} = \frac{\mu}{2} \;\; (discarding)$$

$$P_2 \longrightarrow P_0 : \; \frac{\mu}{1-(1-1-0)} = \frac{\mu}{1} \;\; (purging)$$

$$i = n,$$

$$P_n \longrightarrow P_{n-1} : \; \frac{\mu}{n-(n-1-(n-1))} = \frac{\mu}{n} \; (discarding)$$

$$P_n \longrightarrow P_{n-2} : \; \frac{\mu}{n-(n-1-(n-2))} = \frac{\mu}{n-1} \;\; (discarding)$$

$$P_n \longrightarrow P_j : \; \frac{\mu}{n-(n-1-j)} = \frac{\mu}{1+j} \,, \;\; 0 < j \leq n \;\; (discarding)$$

$$P_n \longrightarrow P_1 : \; \frac{\mu}{n-(n-1-1)} = \frac{\mu}{2}, \;\; j = 1 \;\; (discarding)$$

$$P_n \longrightarrow P_0 : \; \frac{\mu}{n-(n-1-0)} = \frac{\mu}{1}, \;\; j = 0 \;\; (purging)$$

The generalized equation 1) of the state transition of $P_i \longrightarrow P_j \; with \; \mu$ is following.

$$\sum_{i=1}^{n} \sum_{j=0}^{i-1} \frac{\mu}{i-(i-1-j)} \tag{1}$$

Where,

$\sum_{j=0}^{i-1} \frac{\mu}{i-(i-1-j)}$: discarding $0 < j < i-1$, but purging at $j=0$

The balance equations 2) and (3) for the asynchronous chain model are as follows.

$$(\lambda + 2\lambda + 3\lambda + \;\; \cdots + \; n\lambda)\, P_0 = \lambda \frac{n\,(n+1)}{2} P_0 \tag{2}$$

$$\begin{aligned} &\lambda \frac{n\,(n+1)}{2} P_0 \\ &= \left(\frac{\mu}{1}\right) P_1 + \left(\frac{\mu}{2} + \frac{\mu}{1}\right) P_2 + \left(\frac{\mu}{3} + \frac{\mu}{2} + \frac{\mu}{1}\right) P_3 + \\ &\quad \ldots + \left(\frac{\mu}{n-1} + \frac{\mu}{n-2} + \ldots + \frac{\mu}{2} + \frac{\mu}{1}\right) P_{n-1} \\ &\quad + \left(\frac{\mu}{n} + \frac{\mu}{n-1} + \frac{\mu}{n-2} + \ldots + \frac{\mu}{2} + \frac{\mu}{1}\right) P_n \\ &= \end{aligned} \tag{3}$$

It shows the balance equations 4) as follows.

$$\lambda \frac{n\,(n+1)}{2} P_0 = \sum_{i=1}^{n} \sum_{j=0}^{i-1} \frac{\mu}{i-(i-1-j)} P_i \tag{4}$$

Where,

$\lambda \frac{n(n+1)}{2} P_0$: all outgoing from P_0

$(\frac{\mu}{1})P_1$: incoming from $P_1$to $P_0$with $\frac{\mu}{1}$

$(\frac{1}{i} + \frac{1}{i-1} + \ldots + \frac{1}{2} + \frac{1}{1})\mu P_i$: incoming from P_i to P_{i-1} with $\frac{\mu}{i}$, ..., and from P_i to $P_0$with $\frac{\mu}{1}$

It shows the outgoings from P_i, the equation 5) as follows.

$$\begin{aligned} &(\lambda P_i + 2\lambda P_i + \ldots + (i-1)\,\lambda P_i + i\lambda P_i) \\ &+ \left(\left(\frac{\mu}{i} + \frac{\mu}{i-1} + \ldots + \frac{\mu}{i-(i-2)} + \frac{\mu}{i-(i-1)}\right) P_i\right) \\ &= \lambda \left(\frac{i\,(i+1)}{2} + \frac{1}{\lambda} \sum_{j=0}^{i-1} \frac{\mu}{i-(i-1-j)}\right) P_i \end{aligned} \tag{5}$$

The outgoings and the incomings of the balance equation 6) of P_i is following.

$$\lambda\left(\frac{(n-i)\,(n-i+1)}{2}+\frac{1}{\lambda}\sum_{j=0}^{i-1}\frac{\mu}{i-(i-1-j)}\right)P_i$$
$$=\lambda P_{i-1}+2\lambda P_{i-2}+\ldots+(i-1)\,\lambda P_1+i\lambda P_0 \tag{6}$$

$$i=1,$$

$$\lambda\left(\frac{(n-1)\,n}{2}+\frac{1}{\lambda}\sum_{j=0}^{0}\frac{\mu}{1+j}\right)P_1=\lambda P_0$$

$$P_1=\left(\frac{(n-1)\,n}{2}+\frac{1}{\lambda}\sum_{j=0}^{0}\frac{\mu}{1+j}\right)^{-1}P_0$$

$$P_1=q_1^{-1}P_0$$

$$q_1=\frac{(n-1)\,n}{2}+\frac{1}{\lambda}\sum_{j=0}^{0}\frac{\mu}{1+j}$$

$$i=2,$$

$$\lambda\left(\frac{(n-2)\,(n-1)}{2}+\frac{1}{\lambda}\sum_{j=0}^{1}\frac{\mu}{1+j}\right)P_2=\lambda\left(P_1+2P_0\right)$$

$$P_2=\left(\frac{(n-2)\,(n-1)}{2}+\frac{1}{\lambda}\sum_{j=0}^{1}\frac{\mu}{1+j}\right)^{-1}\left(P_1+2P_0\right)$$

$$P_2=q_2^{-1}\left(q_1^{-1}+2\right)P_0$$

$$q_2=\frac{(n-2)\,(n-1)}{2}+\frac{1}{\lambda}\sum_{j=0}^{1}\frac{\mu}{1+j}$$

$$i=3,$$

$$\lambda\left(\frac{(n-3)\,(n-2)}{2}+\frac{1}{\lambda}\sum_{j=0}^{2}\frac{\mu}{1+j}\right)P_3=\lambda\left(P_2+2P_1+3P_0\right)$$

$$P_3=\left(\frac{(n-3)\,(n-2)}{2}+\frac{1}{\lambda}\sum_{j=0}^{2}\frac{\mu}{1+j}\right)^{-1}\left(P_2+2P_1+3P_0\right)$$

$$P_3=q_3^{-1}\left(q_2^{-1}\left(q_1^{-1}+2\right)+2q_1^{-1}+3\right)P_0$$

$$q_3=\left(\frac{(n-3)\,(n-2)}{2}+\frac{1}{\lambda}\sum_{j=0}^{2}\frac{\mu}{1+j}\right)$$

$$i=n-1,$$

$$\lambda\left(\frac{(n-1-i)\,(n-i)}{2}+\frac{1}{\lambda}\sum_{j=0}^{\mathrm{n}-2}\frac{\mu}{1+j}\right)P_{n-1}$$
$$=\lambda\left(P_{\mathrm{n}-2}+2P_{\mathrm{n}-3}+\ldots+(\mathrm{n}-1)\,P_0\right)$$

$$P_{n-1}=\left(\frac{(n-1-i)\,(n-i)}{2}+\frac{1}{\lambda}\sum_{j=0}^{n-2}\frac{\mu}{1+j}\right)^{-1}$$
$$\left(P_{n-2}+2P_{\mathrm{n}-3}+\ldots+(\mathrm{n}-1)\,P_0\right)$$

$$P_{\mathrm{n}-1}=q_{\mathrm{n}-1}^{-1}\left(\begin{array}{l}q_{n-2}^{-1}\left(q_{n-3}^{-1}(\ldots\left(q_1^{-1}+2\right)+2q_1^{-1}+3\right))\\+\ldots+(\mathrm{n}-1)\end{array}\right)P_0$$

$$q_{\mathrm{n}-1}=\left(\frac{(n-1-i)\,(n-i)}{2}+\frac{1}{\lambda}\sum_{j=0}^{n-2}\frac{\mu}{1+j}\right)$$

$$i=n,$$

$$\lambda\left(\frac{(0)\,(1)}{2}+\frac{1}{\lambda}\sum_{j=0}^{n-1}\frac{\mu}{1+j}\right)P_n=\lambda P_{n-1}+2\lambda P_{i-2}$$
$$+\ldots+(n-1)\,\lambda P_1+n\lambda P_0$$

$$P_n=\left(0+\frac{1}{\lambda}\sum_{j=0}^{n-1}\frac{\mu}{1+j}\right)^{-1}\left(\begin{array}{l}P_{n-1}+2P_{i-2}\\+\ldots+(n-1)\,P_1+nP_0\end{array}\right)$$

$$P_n=q_n^{-1}\left(q_{n-1}^{-1}\left(q_{n-2}^{-1}\left(\begin{array}{l}q_{n-3}^{-1}(\ldots\left(q_1^{-1}+2\right)\\+2q_1^{-1}+3\end{array}\right)\right)+\ldots+n\right)P_0\right)$$

Where,

$$q_n=\left(\frac{1}{\lambda}\sum_{j=0}^{n-1}\frac{\mu}{1+j}\right)$$

A generalized form of P_i, equation 7) is expressed as follows.

$$P_i=\ q_i^{-1}P_0\left[\sum_{j=1}^{i}j\left[\sum_{k=1}^{i-1}k\left[\prod_{l=1}^{k-1}q_l^{-1}\right]\right]+i\right] \tag{7}$$

Where,

$$q_i^{-1}=\left(\frac{(n-i)\,(n-i+1)}{2}+\frac{1}{\lambda}\sum_{j=0}^{i-1}\frac{\mu}{1+j}\right)^{-1} \tag{8}$$

From equation 8), the sigma form can be solved as follows.

$$\sum_{j=0}^{i-1}\frac{1}{1+j}=\frac{1}{1+0}+\frac{1}{1+1}+\frac{1}{1+2}+\ldots+\frac{1}{1+(i-2)}+\frac{1}{1+(i-1)}$$

$$\sum_{j=0}^{i-1}\frac{1}{1+j}=\sum_{j=1}^{i}\frac{1}{j}$$

$$\sum_{j=1}^{i}\frac{1}{j}=\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{i-1}+\frac{1}{i}$$

It is partial sums of the harmonic series as the following equation 9).

$$\sum_{k=1}^{\infty}\frac{1}{k}=1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{n}+\ldots+\frac{1}{\infty} \tag{9}$$

The partial sums are written as following equation 10).

$$H_n=\sum_{k=1}^{n}\frac{1}{k}=1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{n} \tag{10}$$

$$P_0+\sum_{i=1}^{n}P_i=1$$

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

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

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

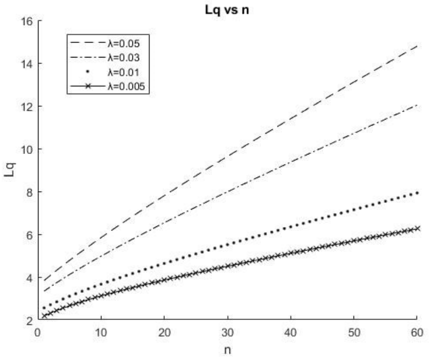

Figure 4: L_Q vs. n for given pairs of λ and μ = 0.0667

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

Where,

$$q_i^{-1} = \left(\frac{(n-i)\,(n-i+1)}{2} + \frac{1}{\lambda} H_n\right)^{-1}$$

The performance measurements of primary interests in the asynchronous model are expressed in L_Q, W_Q, W, and L as following Equations 11), (12), (13), and Equation 14) respectively.

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).

$$L_Q = \sum_{i=0}^{n} iP_i \tag{11}$$

Where,

$$\sum_{i=0}^{n} iP_i = \sum_{i=0}^{n} \left(iq_i \left[\sum_{j=1}^{i} j \left[\sum_{k=1}^{i-1} \left[\prod_{l=1}^{k-1} q_l\right] k\right] + i\right] \right)$$

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

$$W_Q = \frac{L_Q}{\lambda} \tag{12}$$

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

$$W = W_Q + \frac{1}{\mu} \tag{13}$$

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).

$$L = \lambda W \tag{14}$$

4 NUMERICAL SIMULATION OF PROPOSED VBAVAS MODEL

Figure 4 plots L_Q based on Equation 11) with respect to n for given pairs of λ and μ. Observe that as λ goes 0.005 to 0.05, at μ=0.0667, L_Q grows in a monotonic manner as expected.

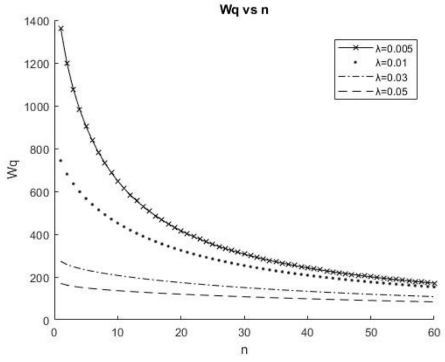

Figure 5: W_Q vs. n for given pairs of λ and μ = 0.0667

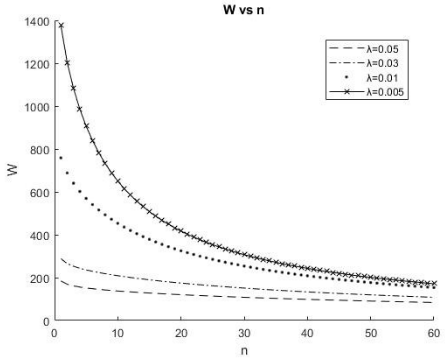

Figure 6: W vs. n for given pairs of λ and μ = 0.0667

Figure 5 plots W_Q based on Equation 12) with respect to n for given pairs of λ and μ. Observe that as λ goes 0.005 to 0.05, at μ=0.0667, W_Q drops in a monotonic manner yet rather exponentially at lower λ, which indicates that W_Q is more sensitive to n at lower λ.

Figure 6 plots W based on Equation 13) with respect to n for given pairs of λ and μ. Observe that as λ goes 0.005 to 0.05, at μ=0.0667, W trends more or less the same patterns.

Figure 7 plots L based on Equation 14) with respect to n for given pairs of λ and μ. Observe that as λ goes 0.005 to 0.05, at μ=0.0667, L grows in a monotonic manner as expected and as in the case of L_Q.

The throughput per block (γ) in the asynchronous chain model can be obtained from Equation 15). Figure 8 plots γ versus n for given pairs of λ and μ. Observe that as λ goes 0.005 to 0.05, at μ=0.0667, γ grows in rather a logarithmic trends with steeper slopes at lower λ.

$$\gamma = \ \mu P_n = \mu \, \frac{\lambda}{\mu} \frac{n\,(n+1)}{2} P_0 = \ \lambda \frac{n\,(n+1)}{2} P_0 \tag{15}$$

5 IMPLEMENTATION OF AN ASYNCHRONOUS CHAIN

An algorithm is developed to realize an asynchronous chain and the algorithm is used to modify the Ethereum source code to demonstrate the effect of the asynchronous control of the posting of the

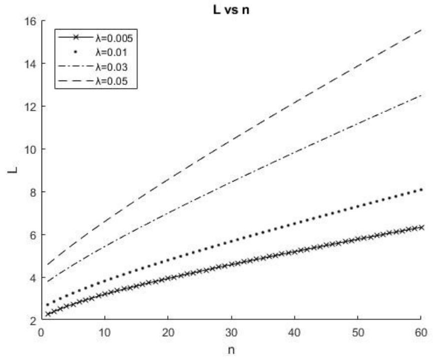

Figure 7: L vs. n for given pairs of λ and μ = 0.0667

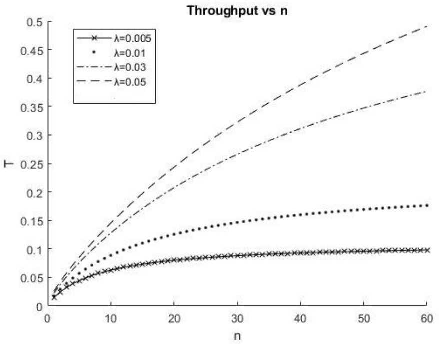

Figure 8: Throughput vs. n for given pairs of λ and μ = 0.0667

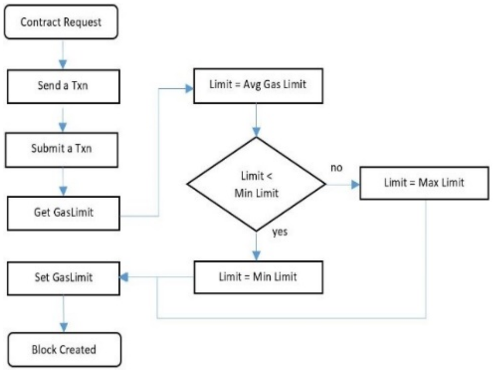

Figure 9: Experiment procedure and gas limit algorithm

transactions versus the conventional fixed block size-based strict synchronous control of transactions posting.

In the algorithm, the gas fee of each transaction is used as an estimate of the size of the transactions as the gas fee is proportional to the amount of computation both spatially and temporally without loss of generality and practicality. In fact, the blocks in the original Ethereum source code refer to the amount of gas fees of the transactions to determine the timing of block posting.

Figure 9 presents the proposed asynchronous chain algorithm.

The algorithm reads the gas fee in a transaction as an estimate of the size being executed and dispatched off of a contract to be posted in the current block. Then, the variable Limit is set to the average value across the gas limits including the active and current one, namely in a proactive manner, thus far, and then compares the new Limit against Min Limit and if the new Limit is less than Min Limit, then Limit is set to Min Limit, otherwise, Limit is set to Max Limit, thereby determines whether the current transaction is to be posted or not with respect to the new Limit. Notice that the computational complexity is determined by the cost to maintain the Max and Min Limits in a matter of linear time and the average gas limit can be computed in constant time, resulting in an overall linear time, which won't slow the block posting process as the conventional block posting will take linear time as well to check the total gas fees against the gas fee limit per block.

Precisely, in an interest to simplify and mitigate the complexity of the proposed asynchronous control of the block posting time, the block size (or gas limit) is adapted to the extent of 100% (note that this option indicates that there is no adaptation on the size of the block whatsoever), 70%, 50%, 30%, and 15% of the current block size, and W, L, and γ are computed, accordingly and respectively. Per block, about 20~24 transactions are executed in an arbitrary manner with respect to the gas fee. Note that the experimental setup is on a private net; a single blockchain account is created in the private per block; a fixed-size (with 158050 gas fee) contract in solidity code is used; 4 threads are created for a mining; and the hardware specification is such that macOS Catalina version 10.15.7, 3.2 GHz Quad-Core Intel Core i5, Memory 16 GB 1600 Mhz DDR3, and Graphics NVIDIA GeForce GTX 675MX 1GB.

The performance of the proposed asynchronous chain is empirically demonstrated versus each adaptive block size as follows.

100% of block size

The time transactions arrived is plotted along with their arrival rate measured at approximately 0.332 μs for total 20 transactions as shown in Figure 10, where the joint trend provides a transactions traffic pattern. Those transactions are posted split into two adjacent blocks, 15 into the first block with total 2377575 gas fee and 5 into the second block with total 792525 gas fee as a result of the proposed asynchronous control of the block posting times of the transactions with respect to the gas fees of the transactions, respectively. The average waiting time of 20 transactions is measured approximately 18.219s, 10.978s in the first block and 39.946s in the second block, respectively. Note that the waiting time on each transaction is supposed to be affected by the mining process in the corresponding block resulting in different waiting times block-wise as shown in Figure 11. The average block time is measured 25.5s across those two adjacent blocks involved, split into 15s into the first block and 40s the second block, respectively. Each block time determined by the average waiting time of transactions at a constant arrival rate, and hence the throughput of the first block gets higher than the one of the second as shown in Figure 1.

70% block size

The time transactions arrived is plotted along with their arrival rate measured at approximately 0.333 μs for total 24 transactions as shown in Figure 13. Those transactions are posted split into two adjacent blocks, 12 evenly into each block with total 1902060 gas fee as a result of the proposed asynchronous control of the

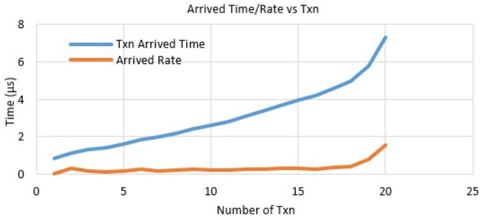

Figure 10: Arrived Time/Rate vs. Txn with 100% gas limit per block

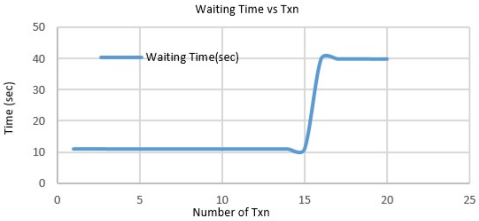

Figure 11: Waiting Time vs. Txn with 100% gas limit per block

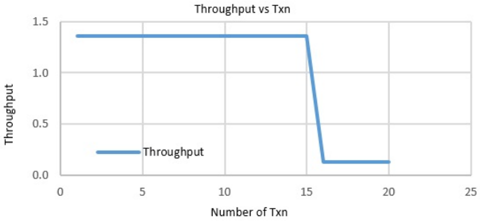

Figure 12: Throughput vs. Txn with 100% gas limit per block

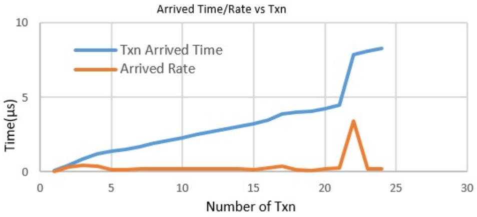

Figure 13: Arrived Time/Rate vs. Txn with 70% gas limit per block

block posting times of the transactions with respect to the gas fees of the transactions, respectively. The average waiting time of 24 transactions is measured approximately 12.468s, 9.984s in the first block and 14.952s in the second block, respectively. The block-wise waiting times are shown in Figure 14. The average block time is measured 12.5s across those two adjacent blocks involved, split into 10s into the first block and 15s the second block, respectively. The first block exhibits higher throughput than the one of the second as shown in Figure 15, however in a narrower gap than in the case of 100% block size due to the narrow gap of the block sizes across two adjacent blocks.

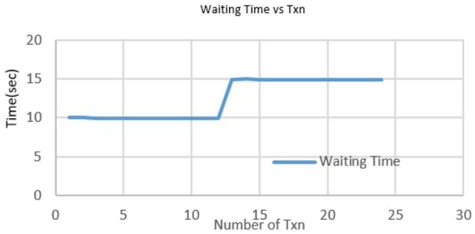

Figure 14: Waiting Time vs. Txn with 70% gas limit per block

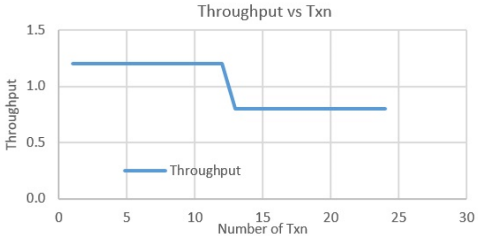

Figure 15: Throughput vs. Txn with 70% gas limit per block

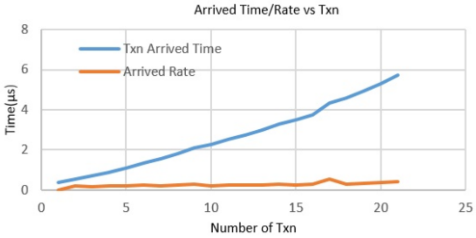

Figure 16: Arrived Time/Rate vs Txn with 50% gas limit per block

50% of block size

The time transactions arrived is plotted along with their arrival rate measured at approximately 0.255 μs for total 21 transactions as shown in Figure 16. Those transactions are posted split into three adjacent blocks due to the smaller sizes of the blocks, 7 evenly into each block with total 1109535 gas fee as a result of the proposed asynchronous control of the block posting times of the transactions with respect to the gas fees of the transactions, respectively. The average waiting time of 21 transactions is measured approximately 6.306s, 0.99s in the first block, 12.974s in the second block, and 4.954s in the third block, respectively. The block-wise waiting times are shown in Figure 17. The average block time is measured 6.333s across those three adjacent blocks involved, split into 1s, 13s, and 5s, respectively. The first block exhibits higher throughput than the one of the second one, however, the third one turn slightly higher than the second one as shown in Figure 18, due to the reduced waiting time in the third block.

30% of block size

The time transactions arrived is plotted along with their arrival rate measured at approximately 0.235 μs for total 22 transactions as shown in Figure 19. Those transactions are posted split into six

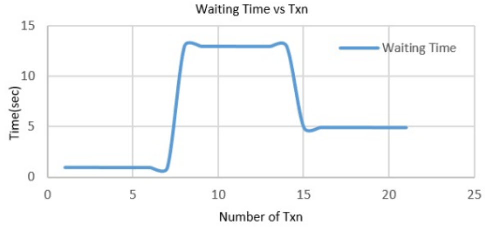

Figure 17: Waiting Time vs. Txn with 50% gas limit per block

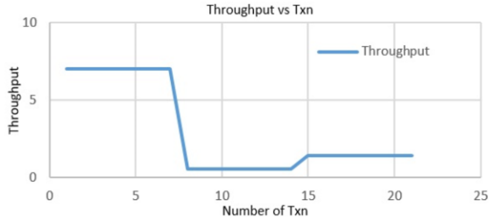

Figure 18: Throughput vs. Txn with 50% gas limit per block

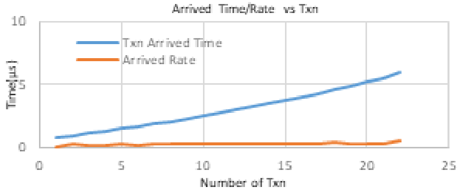

Figure 19: Arrived Time/Rate vs Txn with 30% gas limit per block

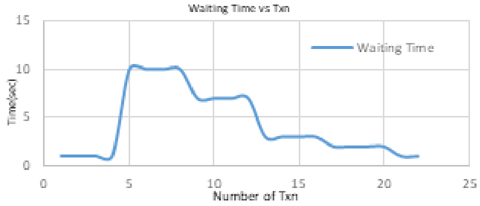

Figure 20: Waiting Time vs Txn with 30% gas limit per block

adjacent blocks due to the smaller sizes of the blocks, 4 evenly into the first five blocks with total 634020 gas fee and 2 into the last block with total 317010 gas fee as a result of the proposed asynchronous control of the block posting times of the transactions with respect to the gas fees of the transactions, respectively. The average waiting time of 22 transactions is measured approximately 4.242s, 0.992s, 9.9982s, 6.973s, 2.964s, 1.952s, and 0.943s, respectively into each block from the first through the last block. The block-wise waiting times are shown in Figure 20. As dictated by the waiting time in each block, the block-wise throughput is presented as shown in Figure 21, due to the reduced waiting time in the third block.

15% of block size

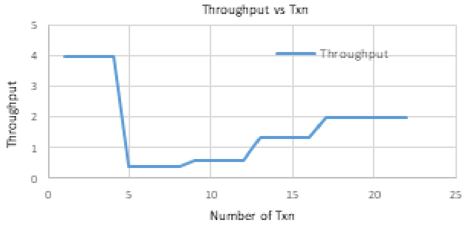

Figure 21: Throughput vs Txn with 30% gas limit per block

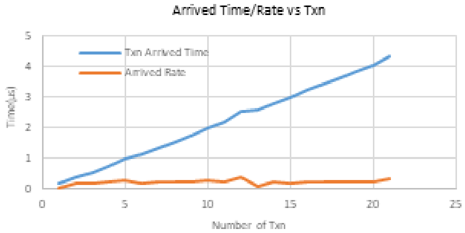

Figure 22: Arrived Time/Rate vs Txn with 15% gas limit per block

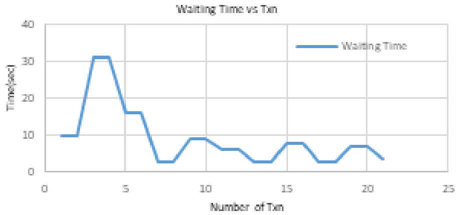

Figure 23: Waiting Time vs Txn with 15% gas limit per block

This case supposed to exhibit the most dramatic impact of block size adaption as follows. The time transactions arrived is plotted along with their arrival rate measured at approximately 0.199 μs for total 21 transactions as shown in Figure 22. Those transactions are posted split into eleven adjacent blocks due to the further smaller sizes of the blocks, 2 evenly into the first ten blocks with total 317010 gas fee and 1 into the last block with total 158505 gas fee as a result of the proposed asynchronous control of the block posting times of the transactions with respect to the gas fees of the transactions, respectively. The average waiting time of 21 transactions is measured approximately 9.298s, 9.997s, 30.993s, 15.989s, 2.985s, 8.981s, 5.976s, 2.972s, 7.969s, 2.964s, 6.960s, and 3.693s, respectively into each block from the first through the last block. The block-wise waiting times are shown in Figure 23. Differently from the previous throughput results and as shown in Figure 24, the throughputs revealed a rather inconsistent trends partly believed to be due to possibly higher sensitivity of other variables than the waiting times.

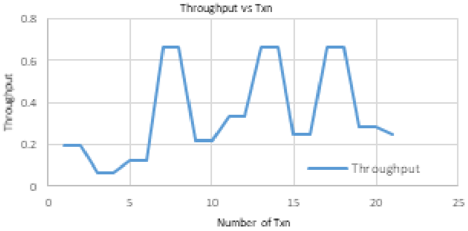

Figure 24: Throughput vs Txn with 15% gas limit per block

6 CONCLUSION

This paper has proposed an asynchronous chain along with a VBAABS model of the $M^{1,n}/M^{1,j,i,n}/1$ type to establish a quantitative method to assess the performance of the proposed asynchronous chain. The model assumes variable bulk arrivals of transactions in Poisson distribution, i.e., $M^{1,n}$, and variable bulk services of transactions, each of which applies to a block potentially of different capacity in terms of the number of slots in it, and every state transition is possibly discarding from i at any slot to j (0<j<i) slot down with μ transition definition $P_i \longrightarrow P_j$ when the mining process is delayed and/or the current block size down, in exponential time. The asynchronous chain is distinguished from the both baseline model and the adaptive model, in comparing the waiting time and the number of transactions in a queue to the block size variation. It has been observed that the block size is proactively reduced as commanded by the proposed asynchronous chain algorithm across 100%, 70%, 50%, 30%, 15% of the block sizes and as expected theoretically, and the waiting times and throughputs of the transactions involved have confirmed and validated the theoretical results to a reasonable extent such that increasing waiting time and decreasing throughput versus the number of transactions. Overall, the implementation has demonstrated that the adaptive block size under the proposed asynchronous control confirms to reduce the waiting time and increase the throughput with reasonable consistency.

REFERENCES

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

[2] Gavin Wood, “Ethereum: A Secure Decentralised Generalised Transaction Ledger, Petersburg Version 4ea7b96 – 2020-06-08”, Yellow Paper, Jun 8, 2020

[3] Vitalik Buterin, “A Next Generation Smart Contract & Decentralized Application Platform”, White Paper, 2014.

[4] Songpu Ai, Diankai Hu, Tong Zhang, Yunpeng Jiang, Chunming Rong, Junwei Cao, “Blockchain based Power Transaction Asynchronous Settlement System”, 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring). DOI: 10.1109/VTC2020-Spring48590.2020.9129593

[5] J. Lind, O. Naor, I. Eyal, F. Kelbert, P. Pietzuch, E.G. Sirer, “Teechain: A Secure Payment Network with Asynchronous Blockchain Access”, arXiv:1707.05454

[6] Rafael Pass, Lior Seeman, Abhi Shelat, “Analysis of the Blockchain Protocol in Asynchronous Networks”, Published in: Advances in Cryptology – EUROCRYPT 2017

[7] Wei Chih Hong, Ying Chin Chen, Ren Kai Yang, Bo Li, Jung San Lee, “Efficient peer-to-peer E-payment based on asynchronous dual blockchain”, Journal of Internet Technology, vol 21. Issue no. 5. Published 2020

[8] Fabian Knirsch, Andreas Unterweger and Dominik Engel, “Implementing a blockchain from scratch: why, how, and what we learned”, EURASIP Journal on Information Security volume 2019, Article number: 2 (2019)

[9] Di Yang, Chengnian Long, Han Xu, Shaoliang Peng, “A Review on Scalability of Blockchain”, ACM ICBCT'20: Proceedings of the 2020 The 2nd International Conference on Blockchain Technology, March 2020. Pages 1–6. https://doi.org/10.1145/3390566.3391665

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

[11] 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

[12] 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

[13] 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

[14] 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

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

[16] 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

[17] 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

[18] 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

[19] 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

[20] Quan-Lin Li, Jing-Yu Ma, and Yan-Xia Chaing, “Blockchain Queue Theory”, arXiv:1808.01795v1 [cs.CE], 6 Aug 2018

[21] Parth Thakkar, Senthil Nathan N, Balaji Viswanathan, “Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform”, 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), 08 Nov 2018. DOI: 10.1109/MASCOTS.2018.00034

[22] Suporn Pongnumkul, Chaiyaphum Siripanpornchana, and Suttipong Thajchayapong, “Performance Analysis of Private Blockchain Platforms in Varying Workloads”, 2017 26th International Conference on Computer Communication and Networks (ICCCN), Sep 2017. DOI: 10.1109/ICCCN.2017.8038517

[23] Peilin Zheng, Zibin Zheng, Xiapu Luo, Xiangping Chen, and Xuanzhe Liu, “A detailed and real-time performance monitoring framework for blockchain systems”, ACM ICSE-SEIP '18: Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice, May 2018. Pages 134–143. https://doi.org/10.1145/3183519.3183546

[24] Mengting Liu, F. Richard Yu, Yinglei Teng, Victor C. M. Leung, Mei Song, “Distributed Resource Allocation in Blockchain-Based Video Streaming Systems With Mobile Edge Computing”, IEEE Transactions on Wireless Communications (Volume: 18, Issue: 1, Jan. 2019). DOI: 10.1109/TWC.2018.2885266

[25] Shishir Rai, Kendric Hood, Mikhail Nesterenko, and Gokarna Sharma, “Blockguard: Adaptive Blockchain Security”, Distributed, Parallel, and Cluster Computing, Jul 2019. arXiv:1907.13232

[26] Jie Xu, Cong Wang, and Xiaohua Jia. 2023. “A Survey of Blockchain Consensus Protocols.”, ACM Comput. Surv. Just Accepted (January 2023). https://doi.org/10.1145/3579845

[27] Gilad, Yossi, *et al.* "Algorand: Scaling byzantine agreements for cryptocurrencies." Proceedings of the 26th symposium on operating systems principles. 2017. https://doi.org/10.1145/3132747.3132757

[28] S. Rouhani and R. Deters, "Performance analysis of ethereum transactions in private blockchain," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2017, pp. 70-74, doi: 10.1109/ICSESS.2017.8342866.

[29] [29] Doshi, Dhruv, and Satvik Khara. "Blockchain-Based Decentralized Cloud Storage." International Conference on Mobile Computing and Sustainable Informatics: ICMCSI 2020. Springer International Publishing, 2021.

[30] [30] Jongho Seol, Abhilash Kancharla, Zuqiang Ke, Hyeyoung Kim, and Nohpill Park. 2020. “A Variable Bulk Arrival and Static Bulk Service Queueing Model for Blockchain.”, In Proceedings of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure (BSCI '20). Association for Computing Machinery, New York, NY, USA, 63–72. https://doi.org/10.1145/3384943.3409423