Quantitative Study on the Performance of An Asynchronous Chain Model

JONGHO SEOL
Department of Computer Science, Middle Georgia State University, jongho.seol@mga.edu

CONG PU
Department of Computer Science, Oklahoma State University, cong.pu@outlook.com

NOHPILL PARK
Department of Computer Science, Oklahoma State University, npark@cs.okstate.edu

This paper presents a quantitative study on the performance of an asynchronous chain model [32] during its theoretical design stage. The asynchronous chain [32] in this study is asynchronous along with adaptively-sized blocks in a proactive manner, whereas the conventional chain controls the block posting in a strictly synchronous manner to the fixed-sized blocks. The model of the adaptive chain, with a rather "reactively" dynamic size of blocks as shown in [31], can be compared versus the proposed model which is asynchronous with a "proactively" dynamic size of the blocks. It is assumed in this paper that variable bulk arrivals (VBA) of transactions in Poisson distribution and asynchronous bulk posting (ABS) of transactions off a block potentially in different capacity in exponential time, referred to as VBAABS. Basic numerical simulations results have been reported in [32] primarily for feasibility validation purpose. In our earlier conference version [32] we have presented the work with focus on development and validation of the performance model in a quantitative manner with extensive numerical simulations to demonstrate the efficacy of the proposed asynchronous chain model, and in this paper we extend the work to also include substantially new works such as an extensive comparative study in performance versus other blockchain models such as the baseline chain model [30] (see Section 5.1) and the adaptive chain model [31] (5.2). Then, a summary by each chain model in consideration, is provided for clarity (5.3). Furthermore, a new simulation is conducted with a focus on the performance of microtransactions as a type of transactions to be commonly expected in the gaming decentralized applications to demonstrate the benefit from the proactive asynchrony of block postings (5.4). Lastly, the implementation results and analysis are shown based on the Ethereum open source as reported in [32].

CCS CONCEPTS • Computer systems organization • Architectures • Distributed architectures • Peer-to-peer architectures

Additional Keywords and Phrases: Blockchain, Asynchronous chain, Adaptive chain, Baseline chain, Queueing model

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

ACM 2769-6480/2025/1-ART
https://doi.org/10.1145/3712619

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 $M^{1,n}/M^{1,j,i,n}/1$ 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 $(0 < i \leq n)$ 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.

In this study, the focus is given on further extensive simulations to be conducted to observe and demonstrate the benefits of the asynchronous chain model versus the baseline chain and adaptive chain models, respectively, with respect to the average transaction waiting time, the average block space requirement, transactions throughput, and network traffic. Furthermore, an extensive simulation and analysis is conducted with a focus on the performance of microtransactions as a type of transactions to be commonly expected in the gaming dApps (decentralized Apps) to ultimately benefit from the proactive asynchrony of block postings.

The objectives in this research are to introduce the VBAABS model, a novel asynchronous chain characterized by adaptively sized blocks in a proactive manner. This model stands in contrast to traditional synchronous chains, such as the $M^{1,n}/M^{1,n}/1$ type (VBAVBS) model, which controls block posting strictly synchronously with fixed-sized blocks; To make a quantitative distinction between the adaptive model (VBAVBS) and the asynchronous model (VBAABS). Unlike VBAVBS, where every state P_i transitions back to P_0, in VBAABS, every state P_i, where $0 < i \leq n$, transitions not only back into P_0 but also into P_j, where $0 < j < i$. This distinction has implications for transaction delays, and the paper rigorously evaluates and quantifies these differences; To conduct basic numerical simulations to validate the feasibility of the proposed model (VBAABS) and presents results primarily for feasibility validation. Extensive simulations are then performed to assess the performance advantages of the asynchronous chain model against both baseline chain models (VBASBS) and adaptive chain models (VBAVBS); To develop and implement the asynchronous chain model within the Ethereum open source. This implementation facilitates a more practical assessment of the proposed model's benefits; To case study the prevalence of microtransactions in gaming decentralized applications (dApps by conducting a dedicated simulation and analysis focusing on the performance of microtransactions within the asynchronous chain model. This aspect provides insights into the potential benefits of proactive asynchrony in block postings for specific transaction types; To broaden the understanding of designing asynchronous chains during their initial theoretical design stage. Through quantitative analysis, the paper sheds light on average transaction waiting times, block space requirements, transactions throughput with respect to transaction arrival rate, block posting rate and network traffic.

The paper is organized as follows: a baseline model (i.e., VBASBS [30]) and an adaptive model (i.e., VBAVBS [31]) are reviewed in the following section II as the preliminaries to the proposed approach; the VBAABS model [32] is presented in the following section III; a section IV will follow to demonstrate numerical simulations versus various blockchain-related parameters and results are shown to demonstrate the performance benefits of the VBAABS; section V extends the simulations and analyses on performance of the asynchronous chain model in a manner of comparative study against the baseline chain and the adaptive chain models; a simulation and analysis is further presented with focus on the expected performance of microtransactions as a common type of transactions in the gaming dApps; then 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 VI; Then conclusions and discussion are drawn in section VII.

2 PRELIMINARIES AND LITERATURE REVIEW

Dynamic block size adjustment, as an underlying central technology to a Dynamic Blockchain-based Decentralized Network Computing, has been considered as an alternative solution to address the block delay with respect to the gas limit per block and to improve the performance [20,21,22,23]. The gas limit per block proportionally influences block delay, scalability [10,11,12,13,14] and dependability [19]. This research proposes an asynchronous chain and an analytical model, namely, a Variable Bulk Arrival and Asynchronous Bulk Service (VBAABS) Model, to be developed to study and demonstrate the theoretical performance of an asynchronous chain as an ultimate solution to the proposed dynamic blockchain-based decentralized network computing. The proposed asynchronous chain is to address and respond to the compelling speed needs by such transactions as being exigently mandated to be executed in an asynchronous (i.e., dynamic) manner (c.f., otherwise, all the transactions pending for a posting in the current block are synchronized by the block posting delay which is primarily determined by the gas limit on the block and the total gas used by transactions pending for the block in a static manner determined by the worst case performance). The analytical models will be simulated extensively to compare the basic performances of the proposed models such as the average transaction waiting time, the average number of slots per block and throughput. Further, extensive experiments will be conducted in order to validate the analytical results by redesigning the source code of Ethereum to implement and demonstrate each of the proposed chains including the baseline chain as the base case, e.g., the adaptive, the fully asynchronous and the staged asynchronous chains. The analytical results and the experimental results will be compared and analyzed extensively in order to tune the design models and flows.

The baseline chain [30] considered in this research is the conventional chain (e.g., Ethereum) with a fixed size for the blocks in a static manner throughout, and note that, in this research, the size of a block is defined (represented) by the block gas limit which is fixed throughout unless otherwise designed, and also note that in fact the physical size of each block in the baseline chain is practically fixed at 20~30KB [26] per block primarily determined by the number of transactions to fill in the block multiplied by the size of a node to represent a transaction hash which is in fact nothing to do with the size of the block to be adjusted as is concerned in this research. A variable bulk arrival and static bulk service (VBASBS) of the $M^{1,n}/M^n/1$ type can be considered to analyze the performance of the baseline chain and can serve as the basis of analysis of such a variety of variations of chains as adaptive, fully asynchronous and staged asynchronous chains as proposed in this research. VBASBS is an embedded markovian model and defines the state of the chain by a single variable i, i.e., the number of slots pending in the current block for being posted, where note that a slot is approximated to 100,000 wei in this research and note that a typical gas limit per block is set to 10 gwei [26], and state transitions triggered from state i to j, where $0 \leq i, j \leq n$ and $i < j$, and n is the total possible number of slots to be posted in the current block, and a state transition from n back to 0 takes place when the current block is topped off and posted in the chain.

In the VBASBS model [30] and VBAVBS model [31], which are the basis for the VBAABS model [32] in this study, embedded Markovian models have been proposed as shown in Figure 1 for the state transition diagram of the baseline chain model and Figure 2 shows the states transition diagram of the adaptive chain model, respectively.

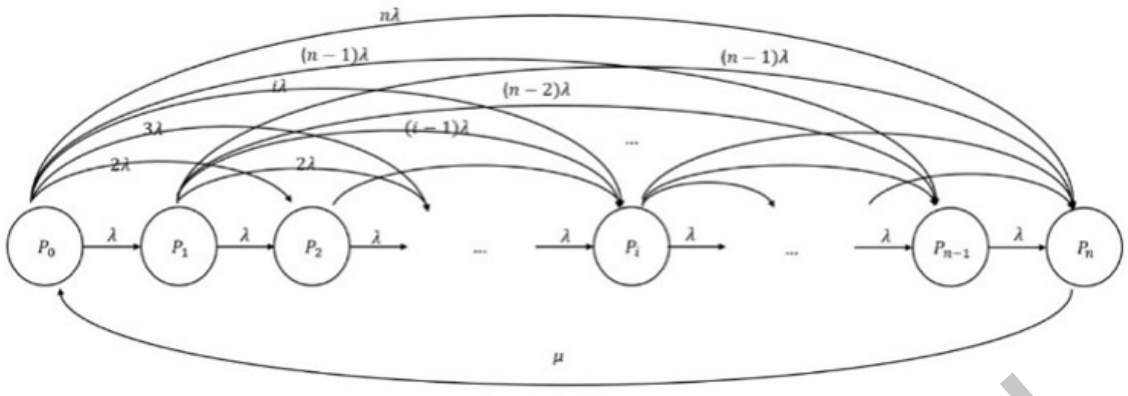


Figure 1. State transition diagram of the baseline chain model, VBASBS [30].

The adaptive chain [31] is defined as a chain in which the block size (i.e., the block gas limit as defined in this research) is adaptive to the total of the gas fees paid by the transactions in the previous block with a certain variation (or the average of a certain number of previous blocks) in a reactive manner (i.e., the size of the block is adapted afterwards in other words). Precisely, the size of a block stays static throughout the time period of block delay. A variable bulk arrival and static bulk service (VBASBS) of the $M^{1,n}/M^n/1$ type can be considered to analyze the performance of the baseline chain and can serve as the basis of analysis of such a variety of variations of chains as adaptive, fully asynchronous and staged asynchronous chains as proposed in this research. VBASBS is an embedded markovian model and defines the state of the chain by a single variable i, i.e., the number of slots pending in the current block for being posted, where note that a slot is approximated to 100,000 wei in this research and note that a typical gas limit per block is set to 10 gwei [26], and state transitions triggered from state i to j, where $0 \leq i, j \leq n$ and $i < j$, and n is the total possible number of slots to be posted in the current block, and a state transition from n back to 0 takes place when the current block is topped off and posted in the chain. As it turns to the adaptive chain, VBASBS model can be extended to allow a state transition from each state i back to state 0 in addition to the state transition from state n back to 0, in other words, a block could be of any size and posted (i.e., flushed from any state $0 < i \leq n$) with the size that is adaptive to a certain condition such as (the average of) the total of the sizes of the transactions in the prior block(s), namely, a variable bulk arrival and variable bulk service (VBAVBS) of the $M^{1,n}/M^{1,n}/1$ type under the assumption that the block size adaptation takes place in a reactive manner after each round of block posting process and the size of the block (the block gas limit) stays static throughout the block posting process.

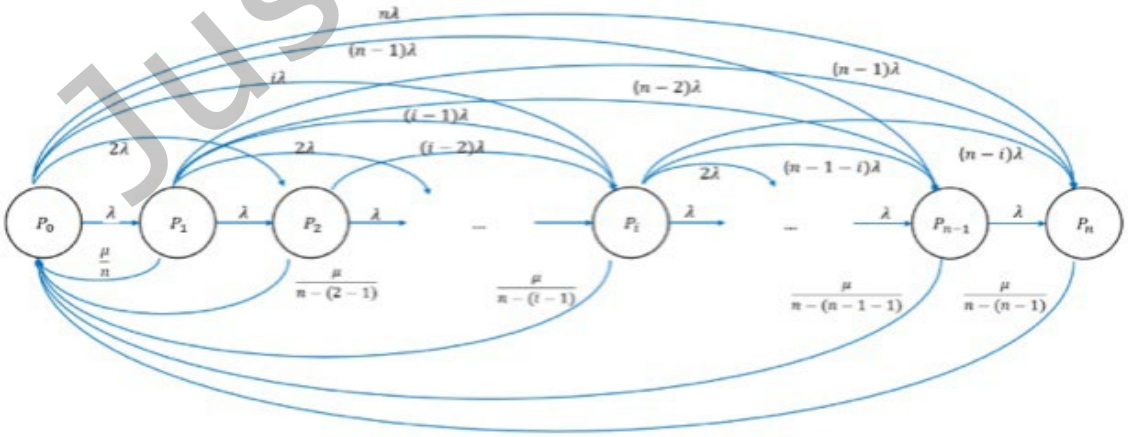


Figure 2. This State transition diagram of the adaptive chain model, VBAVBS [31].

The asynchronous chain [32] to be studied in this research is defined by an adaptive chain yet in a proactive manner, in other words, the size of the block is adapted on the fly. Two different types of asynchronous chains

are proposed such that the fully asynchronous chain adapts the size of the block to the size of an arriving transaction on the fly immediately when a transaction arrives, which seemingly is an overly-costly, stringent and slow solution, yet a fully asynchronous and adaptive solution adapts to the temporal requirement for a transaction being posted in the block; and the staged asynchronous chain moderates the stringent proactive asynchrony requirement of the posting of a transaction such that the adaptive posting is staged (or grouped) by the ranges of the sizes of the transactions, e.g., instead of immediately posting each and every individual transaction in the block, each block posting is delayed for a stage delay to accommodate potentially more transactions within the range of the stage in an effort to relax the stringent asynchronous posting requirement of the fully asynchronous chain. In order to allow a block size to be adaptive in a proactive manner, namely, the asynchronous chain, the block size adaptation should take place on the fly such that the waiting time for a transaction to be posted be strictly asynchronous to its arrival time (namely, the fully asynchronous chain), or, at the most, be shorter than the possible maximum allowed block posting delay (e.g., block gas limit) (namely, a non-fully asynchronous chain such as the staged asynchronous chain as proposed in this research). A variable bulk arrival and asynchronous bulk service (VBAABS) of the $M^{1,n}/M^{ik,n}/1$ type can be considered, where ik indicates the staged group of states such that in the fully asynchronous chain each state i has only an arriving state transition from state 0 and only an outgoing state transition back to state 0 in order to model the strict asynchrony; and in the staged asynchronous chain the strict asynchrony is moderated to some extent by letting states in the range from $ik+1$ to $(i+1)k$ trigger state transitions from l to m, where $ik+1 \leq l$, $m \leq (i+1)k$ and $l < m$ and $(i+1)k \leq n$.

In VBASBS [30] and VBAVBS [31] models, the states are defined 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 [30,31].

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 held on in the proposed VBAVBS model in this paper [30,31].

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 [30,31].

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 jλ, without loss of generality and practicality as well, which will be held on in this paper as well [30,31].

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 held on in this paper as well regardless of the potentially varying number of slots in a block in a normalized manner [30,31].

The state space for the adaptive chain model, VBAVBS, is identical to the baseline model [31] 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 $\frac{\mu}{n-(n-1)}, \frac{\mu}{n-((n-1)-1)}, \ldots, \frac{\mu}{n-1}, \frac{\mu}{n}$, respectively.

The expression for the baseline chain model for P_i at steady state [30] can be generalized as follows.

$$P_i = \left(\lambda\left(\frac{(n-i)(n-i+1)}{2} + \frac{\mu}{\lambda(n-i+1)}\right) + i\lambda P_{i-i}\right)^{-1} \left(\lambda P_{i-1} + 2\lambda P_{i-2} + 3\lambda P_{i-3} + \cdots + (i-1)\lambda P_{i-(i-1)}\right.$$

where,

$$0 < i \leq n,$$

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

The expression for the adaptive chain model for P_i at steady state [31] can be generalized as follows.

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

$$P_0\left[\frac{i^4 - i^2}{4}\left(\left(\frac{\sqrt{2\pi(1-r_1)}\left(\frac{1-r_1}{e}\right)^{(1-r_1)}}{\sqrt{2\pi(-n+1-r_1)}\left(\frac{(-n+1-r_1)}{e}\right)^{(-n+1-r_1)}}\right)\left(\frac{\sqrt{2\pi(1-r_2)}\left(\frac{1-r_2}{e}\right)^{(1-r_2)}}{\sqrt{2\pi(-n+1-r_2)}\left(\frac{(-n+1-r_2)}{e}\right)^{(-n+1-r_2)}}\right)\left(\frac{\sqrt{2\pi(1-r_3)}\left(\frac{1-r_3}{e}\right)^{(1-r_3)}}{\sqrt{2\pi(-n+1-r_3)}\left(\frac{(-n+1-r_3)}{e}\right)^{(-n+1-r_3)}}\right)^{-1}\right) + i\right]$$

where,

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

3 VARIABLE BULK ARRIVAL AND ASYNCHRONOUS BULK SERVICE (VBAABS) MODEL FOR ASYNCHRONOUS CHAIN [32]

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.

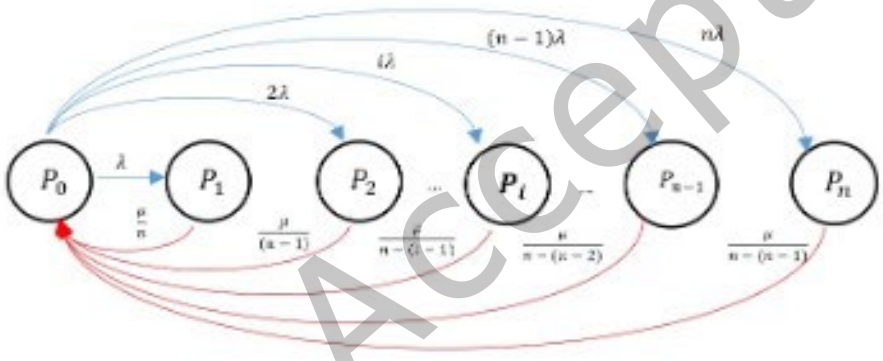


Figure 3. State transition diagram of the asynchronous chain model, VBAABS [32].

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): $P_i \to P_j$

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 $P_i \to P_j$ *with* μ:

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

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 generalized equation (1) of the state transition of $P_i \rightarrow P_j \; with \; \mu$ 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.

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

$$\begin{aligned} &\lambda \frac{n(n+1)}{2} P_0 \\ &= \left(\frac{\mu}{1}\right) P_1 + \left(\frac{\mu}{2} + \frac{\mu}{1}\right) P_2 + \left(\frac{\mu}{3} + \frac{\mu}{2} + \frac{\mu}{1}\right) P_3 + \cdots + \left(\frac{\mu}{n-1} + \frac{\mu}{n-2} + \cdots + \frac{\mu}{2} + \frac{\mu}{1}\right) P_{n-1} \\ &\quad + \left(\frac{\mu}{n} + \frac{\mu}{n-1} + \frac{\mu}{n-2} + \cdots + \frac{\mu}{2} + \frac{\mu}{1}\right) P_n \\ &= \sum_{i=1}^{n} \sum_{j=0}^{i-1} \frac{\mu}{i-(i-1-j)} P_i \end{aligned} \tag{3}$$

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
$\left(\frac{\mu}{1}\right) P_1$: incoming from P_1 to P_0 with $\frac{\mu}{1}$
$\left(\frac{1}{i} + \frac{1}{i-1} + \cdots + \frac{1}{2} + \frac{1}{1}\right) \mu P_i$: 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.

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

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

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

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,

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

P_0 can be solved as follows.

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

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 (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).

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

where,

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

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 SIMULATION AND ANALYSIS OF THE ASYNCHRONOUS CHAIN MODEL [32]

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. This trend aligns with expectations, suggesting that as the arrival rate (λ) experiences a gradual rise within the specified range, the length of the queue L_Q proportionally grows. The impact of this increase on the system, particularly in scenarios where the service rate (μ) remains steady at 0.0667, underscores the sensitivity of the queue length to variations in arrival rates. Such insights are crucial for optimizing system performance and can inform decision-making in resource allocation and capacity planning. Furthermore, this observed monotonic behavior provides a valuable basis for predicting and understanding system dynamics under changing workloads. The analysis of L_Q in response to varying arrival rates contributes to a more comprehensive comprehension of system behavior, facilitating informed adjustments and improvements in scenarios where managing queue length is a critical aspect of operational efficiency.

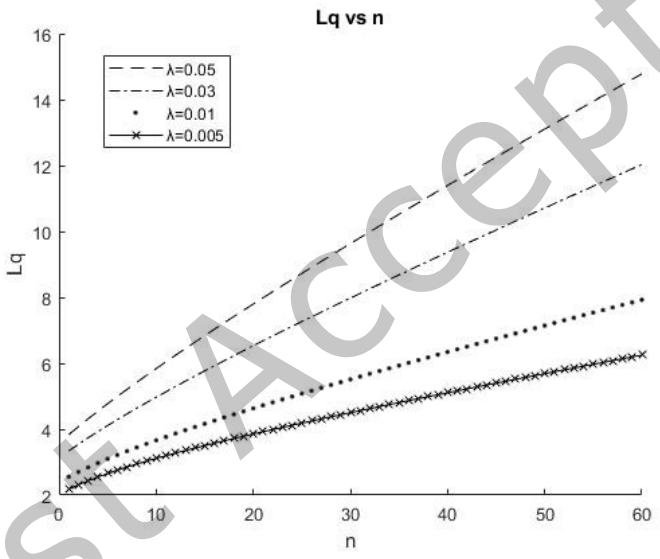


Figure 4. L_Q vs. n for given pairs of λ and $\mu = 0.0667$.

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. Moving beyond the analysis of queue length, Figure 5 investigates into the examination of W_Q, as determined by Equation (12), in relation to n across different combinations of λ and μ. Notably, with μ fixed at 0.0667, as λ transitions from 0.005 to 0.05, a distinct pattern emerges in the behavior of W_Q. The plot reveals a striking monotonic decline in W_Q as λ increases, consistent with expectations. However, the character of this decline exhibits an exponential trend, particularly at lower values of λ. This nuanced observation suggests a heightened sensitivity of W_Q to changes in n when λ is on the lower end of the spectrum. The exponential decrease in W_Q as λ decreases underscores the intricate relationship between the arrival rate and queue wait times. This finding implies that, under conditions of lower arrival rates, the impact on wait times is more pronounced for varying queue lengths. Understanding this sensitivity is pivotal for system optimization, as it highlights the potential for disproportionate effects on waiting times in scenarios with lower demand. In practical terms, this insight could guide decision-making in resource allocation, capacity planning, and service level agreements. The differential sensitivity of W_Q to n under varying

arrival rates provides a nuanced perspective crucial for implementing effective strategies to manage and minimized wait times in dynamic operational environments.

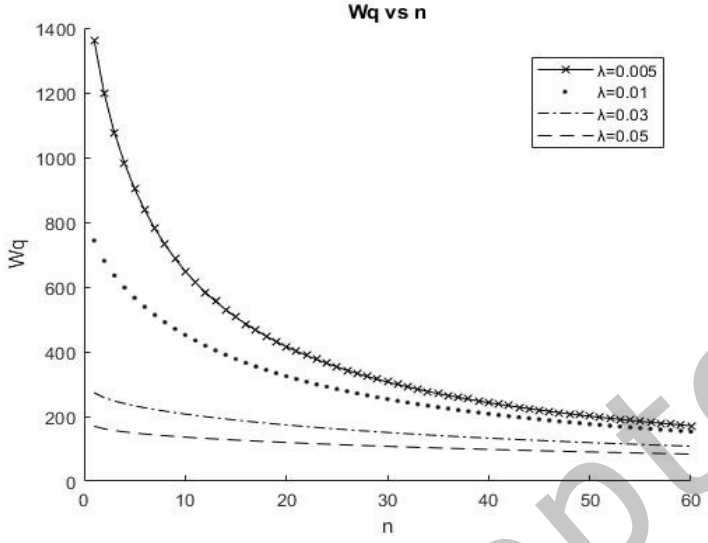


Figure 5. W_Q vs. n for given pairs of λ and μ = 0.0667.

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 6 extends our analysis to the overall system wait time, W, as depicted by Equation (13), in relation to n for diverse pairings of λ and μ. Notably, with a fixed μ value of 0.0667, the plot showcases the behavior of W as λ traverses the range from 0.005 to 0.05. Upon observation, it becomes apparent that the trends in W exhibit a consistency across the spectrum of λ values. Unlike the distinct exponential decline observed in W_Q at lower λ, W presents a more stable pattern as λ varies. This consistency suggests that the overall system wait time is less sensitive to fluctuations in n when compared to the individual queue wait time. The relatively stable behavior of W across different queue lengths emphasize the robustness of the system in managing overall wait times, irrespective of the variations in the arrival rate. Such resilience is valuable in practical scenarios where maintaining consistent service levels is a priority. This finding has implications for system design and optimization, suggesting that while individual queue lengths (W_Q) may exhibit heightened sensitivity to changes in n at lower λ, the impact on the overall system wait time (W) remains more predictable. This insight can guide decision-makers in designing systems that balance the efficiency of individual queues with the overall stability of the system in the face of fluctuating demand.

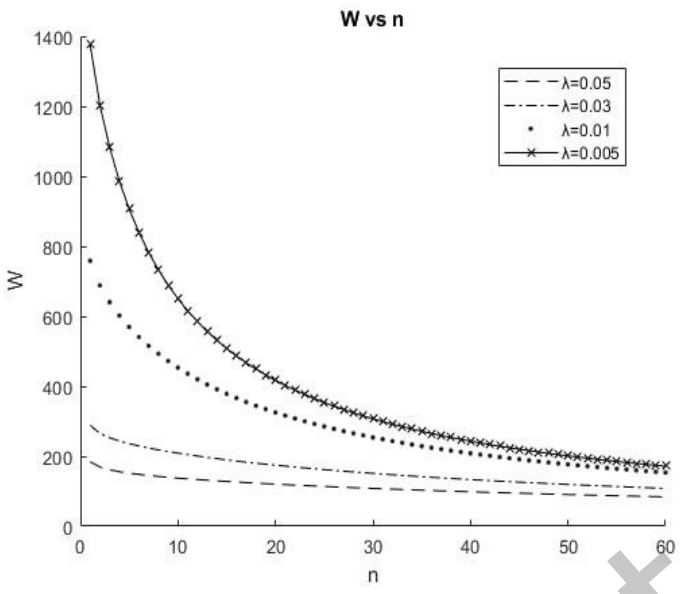


Figure 6: W vs. n for given pairs of λ and μ = 0.0667.

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. Turning our attention to Figure 7, we explore the behavior of the system size, L, as modeled by Equation (14), concerning variations in n across various combinations of λ and μ. Consistent with the patterns observed in both L_Q and W_Q, the analysis here reveals a monotonic increase in L as λ spans from 0.005 to 0.05, maintaining a constant μ of 0.0667. This observed monotonic growth aligns with theoretical expectations and mirrors the behavior noted in the queue length (L_Q) analysis. The steady increase in the system size implies a proportional expansion in the cumulative number of entities within the system, reflecting the responsiveness of the system to changes in arrival rates. Understanding the relationship between the system size and queue lengths is crucial for anticipating the dynamics of a system under varying workloads. The consistent monotonic growth in both L and L_Q further emphasizes the interdependence of these metrics, underscoring the systemic impact of arrival rate variations on overall system behavior. This insight is pivotal for system administrators and designers, providing a basis for predicting system sizes and ensuring that resource allocation and capacity planning efforts are aligned with the expected growth in demand. Additionally, it offers valuable input for the design of scalable systems capable of accommodating increased workloads while maintaining optimal performance.

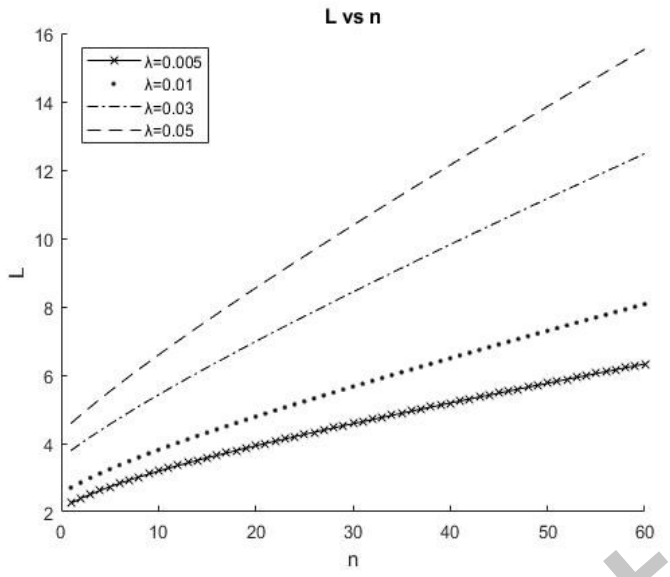


Figure 7: L vs. n for given pairs of λ and $\mu = 0.0667$.

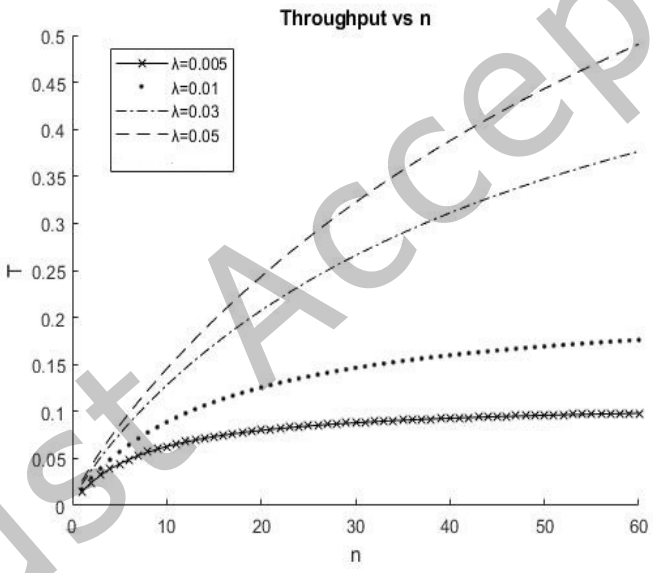


Figure 8: Throughput vs. n for given pairs of λ and $\mu = 0.0667$.

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 trend with steeper slopes at lower λ.

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

Expanding our analysis to the throughput per block (γ) in the asynchronous chain model, as derived from Equation (15), Figure 8 provides a comprehensive view of γ with respect to n for specific combinations of λ and μ. The examination, particularly as λ traverses from 0.005 to 0.05 at a constant μ of 0.0667, reveals intriguing trends in the behavior of γ. A distinctive feature in the plot is the emergence of a logarithmic growth pattern in γ. As λ increases within the specified range, γ exhibits a trend characterized by steeper slopes at lower values of λ. This logarithmic behavior signifies a non-linear relationship between the throughput per block and the system size, suggesting that the impact of changes in n on throughput is more pronounced at lower arrival

rates. The steeper slopes at lower λ underscore the heightened sensitivity of γ to variations in the number of entities in the system under conditions of lower demand. This observation carries implications for system scalability and efficiency, as it implies that throughput experiences more significant fluctuations in response to changes in system size when the arrival rate is lower. Understanding the logarithmic trend in γ provides valuable insights for optimizing system performance under varying workloads. This insight can guide decision-makers in adapting resource allocation and capacity planning strategies to effectively manage throughput in scenarios where demand fluctuates, thereby enhancing the overall responsiveness of the asynchronous chain model.

5 COMPARATIVE STUDY ON ASYNCHRONOUS CHAIN MODEL VERSUS BASELINE CHAIN AND ADAPTIVE CHAIN MODELS

Further simulations are conducted to observe and demonstrate the benefits of the asynchronous chain model in comparison against the baseline chain model as a comparison against a conventional Proof-of-Work blockchain model and adaptive chain models as a comparison against a reactively dynamic chain model while the asynchronous chain model takes a proactive approach in dynamic block posting management under various network-traffic, respectively. The simulations and analyses are conducted with specific respect to the average transaction waiting time, the average block space requirement, and transactions throughput. Particularly, extensive comparative study is conducted in this section across baseline, adaptive, and asynchronous models. It is expected in this study to reveal the benefit of the dynamic management of the size of blocks as exercised by the asynchronous chain model by evaluating its performance in comparison with that of the baseline chain model. Specifically, the comparison with the baseline chain model will demonstrate the performance benefit of dynamic size of blocks over static. The primary contribution of this study is to provide a comprehensive analysis of the performance improvements achieved through the asynchronous chain model, highlighting its advantages in managing block sizes dynamically. This contribution is significant as it ultimately offers insights into optimizing blockchain systems for enhanced efficiency and scalability.

5.1 Comparison between Asynchronous Chain model and Baseline Chain Model

The observations provided below investigate W trends within asynchronous and baseline chain models across various λ values, at 0.05, 0.03, 0.01, and 0.005. The key focus of this investigation is the difference in identifiable behaviors presented by these distinct models. To further explain, the adjustment of the W value within the asynchronous model is embodied at a gradual rate, indicating an almost consistent trend of slowly descending, such as homeostasis. In contrast, the baseline model emphasizes a linear correlation with W and n, where the range of changes appears to be inherently proportional. In particular, it is worth emphasizing that as the λ value decreases, the change in slope is noticeable in the baseline model in Figure 9.

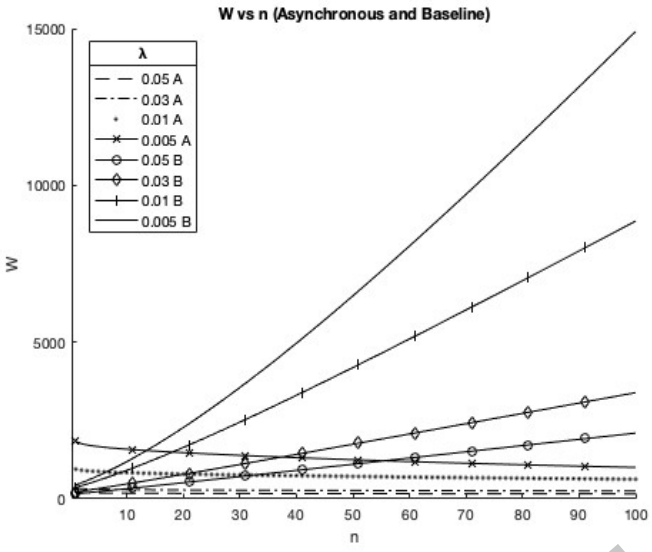


Figure 9: W (vs. n) of asynchronous chain model versus baseline chain model.

Figure 10 mirrors the identical conditions observed in the preceding Figure 9. The ensuing observations delve into the trends of L within asynchronous and baseline chain models, spanning an array of lambda values at 0.05, 0.03, 0.01, and 0.005. This inquiry is fundamentally anchored in the disparities discernible in behavior and inherent model attributes that delineate these queueing models. To elaborate, the fluctuation in L values within the asynchronous model depicts an almost linear trajectory, exhibiting a notably restrained ascent in comparison to the baseline model across all lambda values. In contrast, the baseline model assumes a functional configuration characterized by direct proportionality, with a markedly elevated slope pattern relative to the asynchronous model. A salient feature shared by both the asynchronous and baseline models is their congruent elevation in direct proportion. Furthermore, when the lambda value is elevated, this proportionality is reflected in an escalation of the slope in both models. Consequently, an intuitive correspondence emerges between the augmenting values of L and n in both model paradigms.

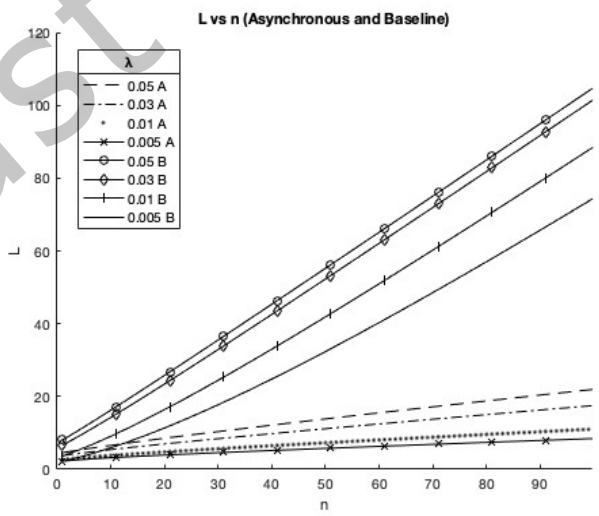


Figure 10: L (vs. n) of asynchronous chain model versus baseline chain model.

The observations provided below systematically investigate the trends of W and L within asynchronous and baseline chain models, dissecting their behavior across a spectrum of lambda (λ) values—namely, 0.05, 0.03,

0.01, and 0.005. The paramount objective underlying this analysis is to discern and articulate the contrasting characteristics inherent in these modeling paradigms.

In terms of W trends, the asynchronous model demonstrates a gradual adjustment in W values, indicative of a nearly constant and gradual descent that draws parallels to a state of homeostasis. Conversely, the baseline model depicts a linear interrelation between W and n, suggesting that the extent of changes remains inherently proportional. Particularly noteworthy is the distinctive shift in the baseline model's slope as the λ value diminishes, prominently displayed in Figure 25.

The following examination, as encapsulated in Figure 26, upholds the identical contextual premises as those of Figure 25. Here, the scrutiny shifts to the trends exhibited by L within both asynchronous and baseline chain models across the same range of lambda values. This analysis pivots around the divergences in behavior and intrinsic model attributes that hallmark these respective queueing models.

Expounding upon the nuances of L, the asynchronous model traces an almost linear trajectory, depicting a notably subdued ascent when juxtaposed with the baseline model, irrespective of the lambda values considered. This contrast becomes particularly pronounced in the baseline model, wherein a functional structure underscores a direct proportionality between L and n, signified by a substantially augmented slope in comparison to the asynchronous model. Notably, a shared facet between the asynchronous and baseline models is their synchronous elevation in direct proportion, further augmented as the lambda value escalates.

In essence, the discernible patterns in both W and L values within these models reflect the interplay between varying lambda (λ) values and the associated changes in slope. This intricate relationship aligns with the intuitive notion of increasing L and n values in both asynchronous and baseline models, emphasizing the model's responsiveness to parameter shifts.

5.2 Comparison between Asynchronous Chain Model and Adaptive Chain Model under Various Network Traffic

The following pair of graphs mirrors the W and L patterns depicted in the preceding figures 11 and 12: the asynchronous and adaptive models.

The present observations delve profoundly into the intricate trajectory of W within both asynchronous and baseline chain models, traversing a gamut of lambda (λ) values: 0.05, 0.03, 0.01, and 0.005. Imbued with a central objective, this inquiry is primed to unveil and elucidate the distinctive behaviors inherent in these contrasting models.

Of particular significance is the dynamic interplay within the asynchronous model, as the adjustment of W unfolds akin to the visual representation previously portrayed. It unveils an almost unwavering, measured descent—a progression redolent of a deliberate and gradual downward trend reminiscent of physiological homeostasis. Contrarily, the adaptive model unveils its own trajectory, one characterized by the gradual diminishment of W values—a departure that markedly distinguishes it from the baseline model. Notably, a nuanced revelation unfurls: the asynchronous model's descent can be perceived as slightly more pronounced than the decreasing slope encountered in the adaptive model.

Significantly, a shared behavior surfaces as λ values decrease in both models: an initial peak in W value followed by a near-linear trajectory approaching a declining curve, as n ascends. It's a noteworthy discovery that the W pattern within the adaptive model at the lowest λ value (0.005) initiates midway between the W patterns of asynchronous models with λ values of 0.005 and 0.01. Moreover, as n progresses, the adaptive model's W pattern dips beneath the W patterns associated with λ values of 0.03 and 0.01 when n is set at 50. Intriguingly, parallels emerge between the patterns at λ values of 0.03 and 0.05 in both models, further intensifying the intrigue of these findings.

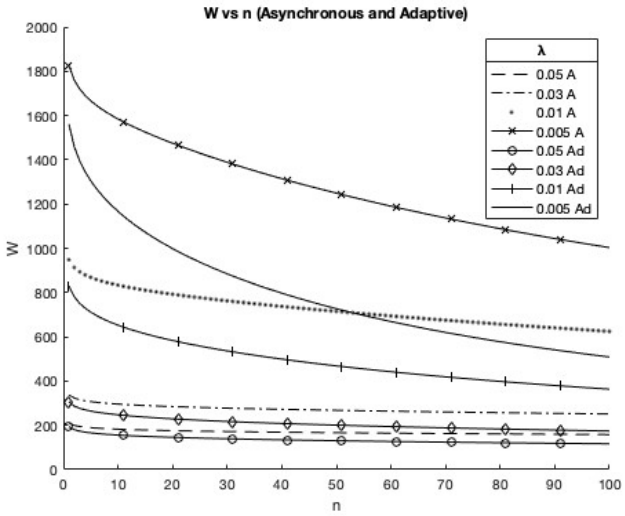


Figure 11: W (vs. n) of asynchronous chain model versus adaptive chain model.

Figure 12 reflects the same conditions observed in previous Figure 11. Subsequent observations examine in detail the trend of L within asynchronous and baseline chain models across arrays of λ values. 0.05, 0.03, 0.01, and 0.005. This investigation is fundamentally fixed on identifiable differences and unique model properties in behaviors depicting these queue models. To be more specific, the variation in L values within the asynchronous model exhibits almost a linear trajectory and shows a more significantly higher rise in the adaptive model across all λ values. Unlike the baseline model, the L value in the adaptive model exceeds the L value of asynchronous at all λ values, but all L values intersect the asynchronous rising as n passes 60 points with the slowly descending adaptive.

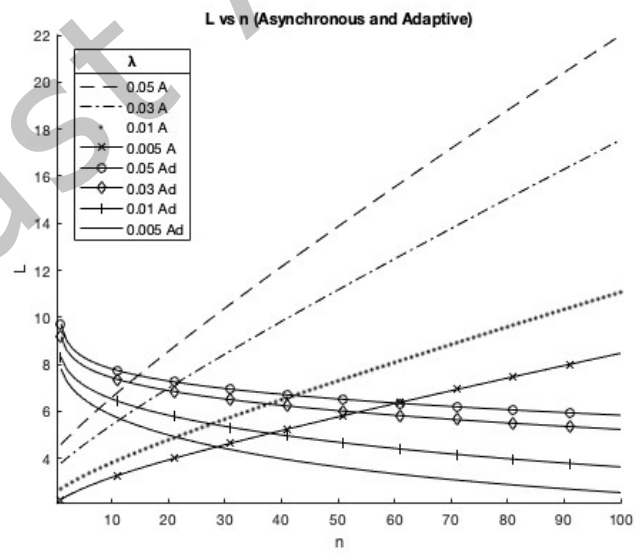


Figure 12: L (vs. n) of asynchronous chain model versus adaptive chain model.

The initial part highlights how Figures 11 and 12 present W and L patterns, respectively, in the asynchronous and adaptive models. The inquiry dives into the trend of W within both models, revealing distinct behaviors.

The asynchronous model displays a consistent, gradual decrease reminiscent of homeostasis, while the adaptive model exhibits its own diminishing pattern, setting it apart from the baseline model. Notably, the descent in the asynchronous model seems slightly more pronounced than in the adaptive model. Furthermore, as λ values decrease, both models experience an initial peak in W followed by a nearly linear trajectory transitioning into a declining curve as n increases. The adaptive model's W pattern at the lowest λ value (0.005) initiates midway between the W patterns of asynchronous models with λ values of 0.005 and 0.01. Additionally, as n increases, the adaptive model's W pattern dips below the W patterns associated with λ values of 0.03 and 0.01, particularly when n reaches 50. This finding is intriguing, particularly the convergence of patterns at λ values of 0.03 and 0.05 in both models. Shifting focus to Figure 28 mirrors the conditions in Figure 27.

The subsequent analysis centers on the trend of L within asynchronous and baseline chain models, spanning various λ values. This examination is geared toward identifying differences and unique model behaviors. Notably, the fluctuation in L values within the asynchronous model follows an almost linear trajectory, displaying a considerably higher ascent in the adaptive model across all λ values. In contrast to the baseline model, the adaptive model consistently surpasses the L value of the asynchronous model across all λ values. However, both L values intersect as n surpasses 60 points, with the adaptive model exhibiting a gradual descent. The combined analysis underscores the distinct characteristics in the behaviors of W and L across both asynchronous and adaptive models. It explores the influence of λ values on slope patterns and establishes intriguing parallels and divergences between the models, offering a comprehensive insight into their dynamics.

In summation, this analysis navigates the complex tapestry of W and L behaviors, underscoring the interdependence of λ values and their concomitant slopes. This intricate synergy aligns seamlessly with the intuitive notion of heightened L and n values, encapsulating the models' responsiveness to the dynamic interplay of parameter permutations.

The following Figures 13, 14, and 15 compare and observe the W value, figures 16, 17, and 18 compare and monitor the L value, and figures 19, 20, and 21 compare and observe the Gamma value to three models: asynchronous, baseline, and adaptive chains. At this time, the network traffic, λ/μ values can be divided into high (0.749), medium (0.083), and low (0.05) to see how they affect changes between models.

In Figures 19, 20, and 21, under conditions of high, medium, and low network traffic, we systematically vary the λ value across high, medium, and low levels. This intricate exploration allows for a comprehensive analysis of the W value within the context of three distinct models. Notably, this investigation reveals consistent patterns across all instances, including LH-B (Low λ /High Network Traffic-baseline), MH-B (Medium λ/High Network Traffic-baseline), and the elevated "HH-B" (High λ /High Network Traffic-baseline). The latter is characterized by a low initial slope that continuously keeps, and all the baseline model stands out prominently W value increasing across these scenarios than other models.

Conversely, both the asynchronous and adaptive chain models showcase a descending slope. Particularly noteworthy is the striking resemblance between the slope and pattern of the graphs representing the HH-A (High λ / High Network Traffic-Asynchronous) and HH-Ad (High λ/ High Network Traffic-Adaptive) models, with both graphs indicating the lowest W value in the observed range.

The trio of graphs showcased below illustrates the variation in gamma resulting from shifts in λ values in conjunction with alterations in network traffic. Of notable significance are the three dominant graphs, prominently featuring the asynchronous chain model, with the peak performer being "HH-A," trailed by "MH-A" and "LH-A." Among the remaining models, their slopes and values exhibit comparably subdued characteristics. It's worth highlighting that "HH-Ad" distinguishes itself with a discernible pattern that sets it apart from the rest of the models.

In particular, Figure 21 warrants attention, revealing a shared pattern between the "HL-B" and "HL-Ad" models, except in the asynchronous case.

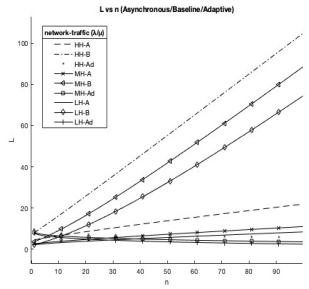


Figure 16: W (vs. n) of asynchronous versus baseline versus adaptive chain model in terms of high-network traffic with λ.

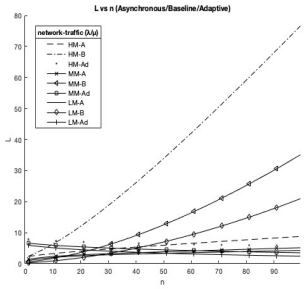


Figure 17: W (vs. n) of asynchronous versus baseline versus adaptive chain model in terms of medium-network traffic with λ.

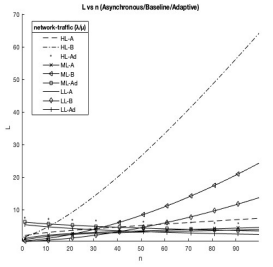


Figure 18: L (vs. n) of asynchronous versus baseline versus adaptive chain model in terms of low-network traffic with λ.

In a comprehensive analysis, the asynchronous chain model emerges as the optimal performer across varying network traffic and λ values, demonstrating consistent superiority.

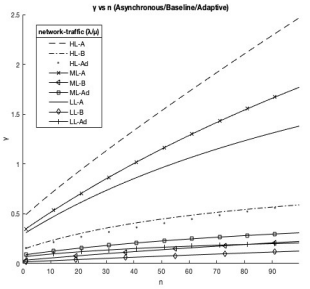


Figure 19: γ (vs. n) of asynchronous versus baseline versus adaptive chain model in terms of high-network traffic with λ.

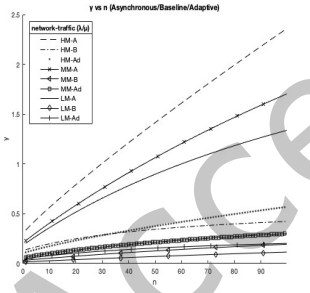


Figure 20: γ (vs. n) of asynchronous versus baseline versus adaptive chain model in terms of medium-network traffic with λ.

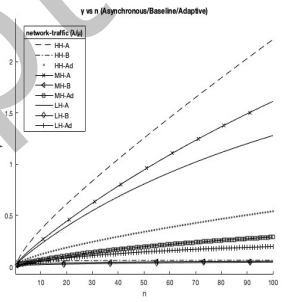


Figure 21: γ (vs. n) of asynchronous versus baseline versus adaptive chain model in terms of low-network traffic with λ.

5.3 Chain Model Insights

For clarity and reference purposes, the simulation results are separated and shown for each chain model, the asynchronous chain model, the baseline chain model, and the adaptive chain model, respectively, as in the following. Each model provides insights into the behavior of $L, W,$ and γ values through the permutations of high, medium, and low network traffic and λ combinations.

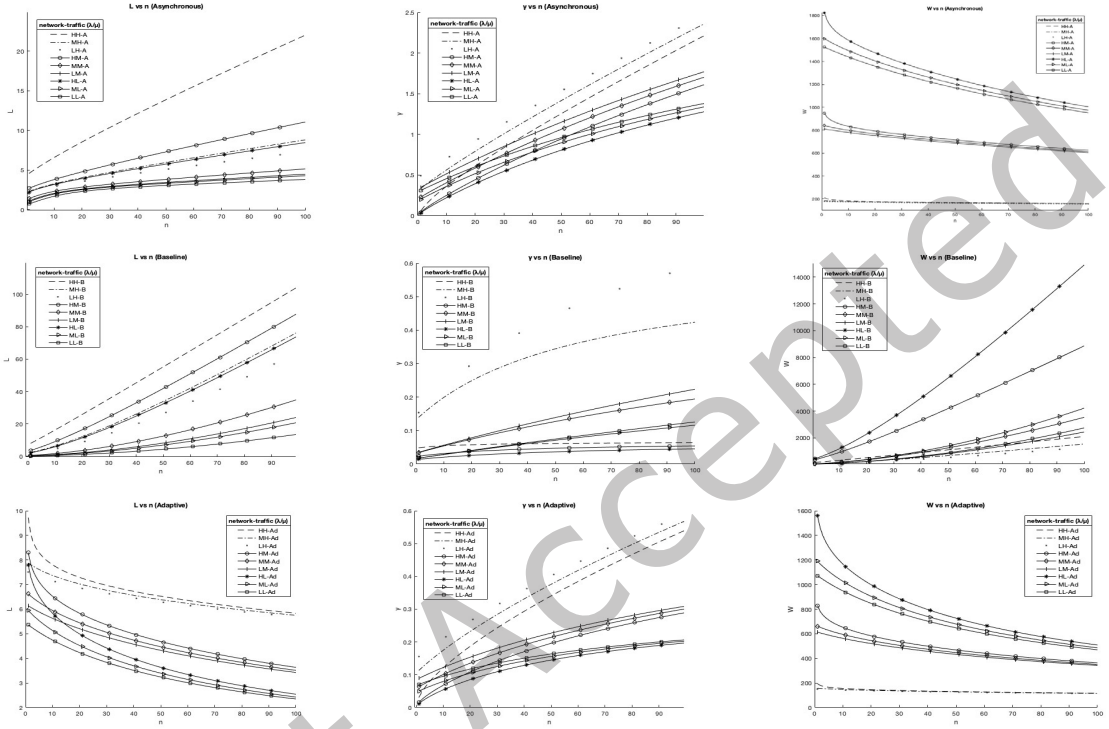
The simulation results have been presented separately for distinct chain models: the asynchronous chain model, the baseline chain model, and the adaptive chain model. This separation facilitates clarity and serves as a point of reference. Each of these models offers insights into the behavior of variables $L, W,$ and γ as they are subjected to permutations and combinations of upper, middle, and lower network traffic scenarios.

In the methods illustrated in Figures 13 to 21, the variables $L, W,$ and γ correspond to various combinations of network traffic intensities, namely high, medium, and low. This arrangement simplifies the comparison and analysis of the three models as they undergo changes. By considering the impact of high, medium, and low network traffic in combination, this approach provides an opportunity to comprehensively grasp the nuances of changes within each chain model.

Consequently, it becomes feasible to discern the optimal combination for the asynchronous model's relatively superior performance, as depicted in Figures 13-21. Similarly, it becomes possible to identify combinations wherein delayed waiting times, along with relatively weaker blockchain characteristics, could be

enhanced. Furthermore, it is worth noting that the adaptive model, while slightly inferior to the asynchronous model, exhibits improved characteristics compared to the baseline model. This suggests that deeper insights into model characteristics and potential enhancement strategies can be derived from the adaptive model.

In summary, the method of segregating simulation results by chain model, combined with the consideration of different network traffic scenarios, enables a thorough exploration of performance nuances. This approach aids in identifying optimal combinations for improved asynchronous performance and highlighting areas for enhancing blockchain characteristics and performance, particularly through the adaptive model's more refined attributes.



5.4 Simulation and Analysis of the Impact of Asynchronous Chain Model on the Performance of Microtransactions

As validated in Section 4, the asynchronous chain model emerges as highly adept in handling microtransactions—such as gaming or minor payments—characterized by their relatively diminutive scale and demand for exceptional efficiency and minimal delays. Consequently, in the forthcoming Section 5, we delve into a comparative analysis of the three models employing notably diminutive λ values (0.001, 0.0005, and 0.0001) meticulously tailored for microtransactions and micropayments. This analysis aims to simulate and scrutinize the performance of these models in this specific context. Furthermore, an extensive simulation and analysis is conducted with a focus on the performance, throughput in particular, of microtransactions as a type of transactions to be commonly expected in the gaming dApps (decentralized Apps) to ultimately benefit from the proactive asynchrony of block postings.

Figure 22 encapsulates the distinctive traits and patterns of the three previously illustrated models. In the context of microtransactions, where input values are injected at an exceptionally rapid rate—signified by a minute λ value of 0.001, 0.0005 and 0.0001—the graph unveils the throughput pattern of these microtransactions. As discerned from the graph's outcomes, the asynchronous pattern significantly outperforms both the baseline and adaptive models in terms of throughput. The ensuing investigation delves sequentially into the adaptive and baseline models to finalize the examination of results.

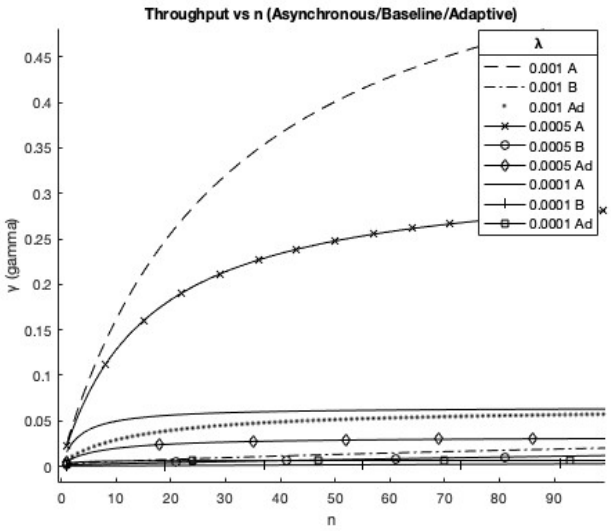


Figure 22: Throughput, γ, (vs. n) of asynchronous chain model versus baseline chain model and adaptive chain model for the micro λ = 0.001, 0.0005, and 0.0001. (A: Asynchronous, B: Baseline, Ad: Adaptive)

From the analysis perspective, Figure 22 presented herewith, serves as an encapsulation of the distinctive attributes and intricate patterns exhibited by the three antecedently showcased models. In the context of microtransactions, a scenario characterized by the swift infusion of values, epitomized by the minute λ value of 0.001, 0.0005, and 0.0001 this graphical representation allows for an insightful exploration of the ensuing throughput patterns of these microtransactions. The discernible outcomes, vividly manifested within this graph, unequivocally affirm the asynchronous pattern's unparalleled prowess in terms of throughput performance, thereby surpassing the baseline and adaptive models by a significant margin. This analytical trajectory culminates in a comprehensive comprehension of the distinctive attributes underpinning these models, thereby contributing substantively to the burgeoning realm of scholarly inquiry as well as practical industry.

Hence, based on the outcomes of this simulation, a conclusive inference can be drawn. The asynchronous chain model not only proves exceptionally fitting for scenarios involving gaming payments or modest-scale personal transactions but also emerges as a compelling option for addressing the persistent issue of sluggish blockchain speeds.

5.5 Implementation of An Asynchronous Chain [32]

The results of an experimental study [32] is presented in this section to demonstrate the feasibility of the asynchronous chain technology by implementing and demonstrating that the asynchronous block size effectively and efficiently facilitates the reduction of the transaction response time (i.e., waiting time) and increase of the throughput as expected.

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 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 23 presents the proposed asynchronous chain algorithm.

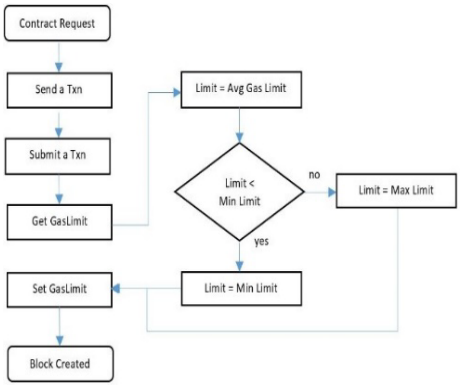


Figure 23: Experiment procedure and gas limit algorithm.

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.

5.5.1 100% 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 24, 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 25. 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 26.

5.5.2 70% of 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 27. 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 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 28. 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 29, 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.

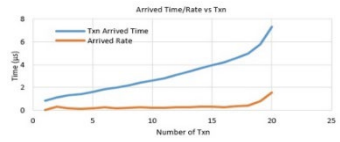


Figure 24: Arrival time/Rate vs. Txn with 100% gas limit per block.

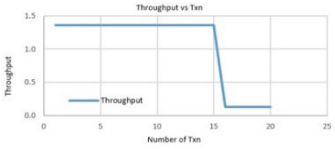


Figure 25: Waiting Time vs. Txn with 100% gas limit per block.

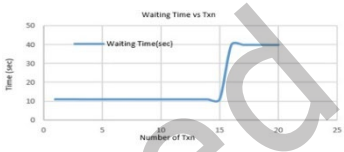


Figure 26: Throughput vs. Txn with 100% gas limit per block.

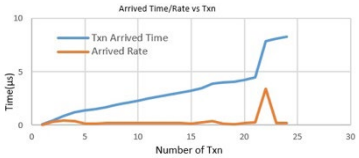


Figure 27: Arrival Time/Rate vs. Txn with 70% gas limit per block.

Figure 28: Waiting Time vs. Txn with 70% gas limit per block.

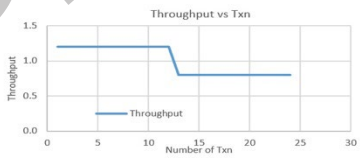


Figure 29: Throughput vs. Txn with 70% gas limit per block.

5.5.3 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 30. 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 31. 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 32, due to the reduced waiting time in the third block.

5.5.4 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 33. Those transactions are posted split into six 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 34. As dictated by the waiting time in each block, the block-wise throughput is presented as shown in Figure 35, due to the reduced waiting time in the third block.

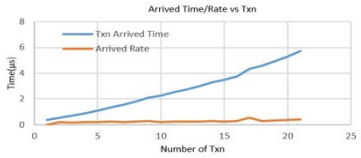


Figure 30: Arrival Time/Rate vs. Txn with 50% gas limit per block.

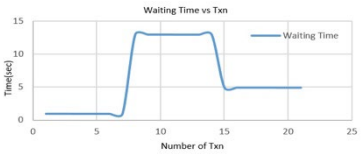


Figure 31: Waiting Time vs. Txn with 50% gas limit per block.

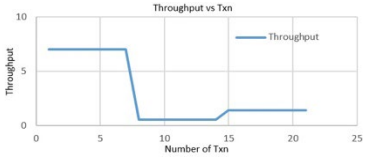


Figure 32: Throughput vs. Txn with 50% gas limit per block.

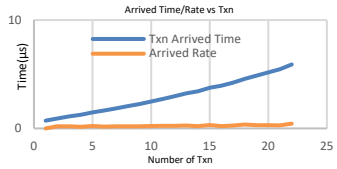


Figure 33: Arrival Time/Rate vs transactions with 30% gas limit per block.

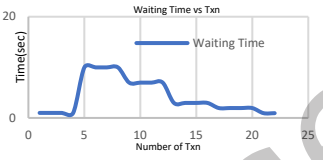


Figure 34: Waiting Time vs transactions with 30% gas limit per block.

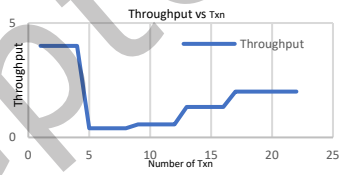


Figure 35: Arrival Time/Rate vs transactions with 30% gas limit per block.

5.5.5 15% of block size

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 36. 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 37. Differently from the previous throughput results and as shown in Figure 38, the throughputs revealed a rather inconsistent trends partly believed to be due to possibly higher sensitivity of other variables than the waiting times.

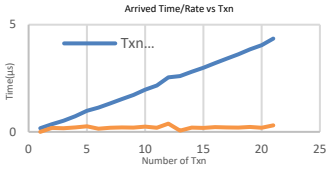


Figure 36: Arrival Time/Rate vs trans-actions with 15% gas limit per block.

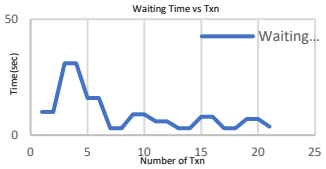


Figure 37: Waiting Time vs transactions with 15% gas limit per block.

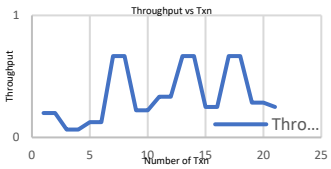


Figure 38: Throughput vs trans-actions with 15% gas limit per block.

6 CONCLUSIONS AND DISCUSSION

This paper has presented an asynchronous chain model of the $M^{1,n}/M^{1,j,i,n}/1$ type to demonstrate a quantitative study the performance of an asynchronous chain as a means to realize a proactively dynamic approach to managing the block posting delays in an effort to improve the scalability of the blockchain [32]. The model assumes variable bulk arrivals of transactions in Poisson distribution, i.e., $M^{1,n}$, and variable bulk services of transactions, resulting in a block potentially of different capacity in terms of the number of slots in it, and each state transition is potentially switching from state i down to state j $(0 < j < i)$ with a transition rate μ, i.e., $P_i \rightarrow P_j$, in exponential time [32]. The asynchronous chain is distinguished from the baseline model and the adaptive such that the asynchronous chain is dynamic in managing the size of the block while the baseline is static; and the asynchronous chain is proactively dynamic to the response time desired by the transactions such as microtransactions as commonly expected in gaming dApps while the adaptive chain is reactively dynamic to the recent block utilization either temporally or spatially. It has been observed that the block size is proactively reduced as commanded by the asynchronous chain throughout 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 in terms of the average transaction waiting time and the average throughput versus the number of transactions with respect to λ, μ, network traffic (λ/μ) to mention a few primary variables [32]. Further, in this work, extensive simulations have been conducted based on the quantitative model as proposed in [32], and the benefits of the asynchronous chain model versus the baseline chain and adaptive chain models, have been observed and demonstrated, respectively. The simulations and analyses were with respect to the average transaction waiting time, the average block space requirement, transactions throughput, and network traffic. Also, a simulation and analysis has been conducted with a specific focus on the performance of microtransactions as a type of transactions to be commonly expected in the gaming dApps (decentralized Apps) to ultimately benefit from the proactive asynchrony of block postings. Lastly, the results of an experimental study [32] has been presented to demonstrate the feasibility of the asynchronous chain technology by implementing and demonstrating that the asynchronous block size effectively and efficiently facilitates the reduction of the transaction response time (i.e., waiting time) and increase of the throughput as expected.

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 (Volume: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] 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] 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

[31] Jongho Seol, Kyle Kentner, Indy N. Park, Shital Joshi, and Nohpill Park, “An Adaptive Blockchain-based Decentralized Network Computing”, Accepted on The Fifth International Conference on Blockchain Computing and Applications (BCCA2023), 24-26 October, Kuwait City, KuwaitAdaptive chain model BCCA 2023

[32] Jongho Seol, and Nohpill Park, “An Asynchronous Chain and A Variable Bulk Arrival and Asynchronous Bulk Service Model”, Accepted on ACM BSCI 2023 (The 5th ACM International Symposium on Blockchain and Secure Critical Infrastructure), July 10th – July 14th 2023, Melbourne, Australia