A Real-Time Performance Model for NFT Chains

K. Kentner
Computer Science Dept.
Oklahoma State University
Stillwater, OK, USA
kyle.kentner@okstate.edu

J. Seol
Information Technology Dept.
Middle Georgia State University
Macon, GA, USA
jongho.seol@mga.edu

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

2023 Fifth International Conference on Blockchain Computing and Applications (BCCA) | 979-8-3503-3923-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/BCCA58897.2023.10338939

***Abstract* - This paper sets forth a bivariate, real-time NFT (nonfungible token) model based on a Markovian queueing process. This model is proposed with the goal of simulating and observing the performance of an NFT system as it behaves under real-time constraints, as well as to see how various on-/off-chain transaction rates, service rates, blocksizes, and deadline-miss rates affect the overall performance of the system. The system is assumed to operate at a steady state, thus allowing for each probabilistic state (which at a given time consists of i on-chain transactions and j off-chain transactions that have arrived) to be calculated using balance equations (all incoming transactions to a state equal all outgoing from the state when the relevant probabilities are considered). The Markovian assumption necessitates that the full state space must sum to 1.0, and because all states derive from $P_{0,0}$, $P_{0,0}$ may be factored out and solved using this constraint. This in turn facilitates the computation of the remaining probabilities. The on-/off-chain transaction rates λ_{on} and λ_{off}, service rate μ, and deadline-miss rate λ_F are the primary parameters within the model, and λ_F is the key parameter for modeling the real-time behavior of the system (λ_F represents a failure for a transaction to meet the deadline, and it causes the current state to self-loop). An analysis of the system via metrics including average wait time, average number of transactions, and throughput is presented, and the implications of the simulation on the overall performance of the system are discussed.**

Keywords—NFT, NFT chain, real time, on-/off- chain, blockchain, queueing model, stochastic process

I. INTRODUCTION

The actions of buying, selling, and tracking assets are fundamental to all business practices. As such, there is a great demand for systems which can facilitate these interactions while maintaining the integrity and security of the assets being bought/sold. Non-fungible tokens (NFTs) are a technology which strives to satisfy this demand using a cryptographically secure means of tracking the ownership of assets [1, 2]. This security is ultimately based on the principles of blockchain technology, which utilizes immutable, peer-to-peer, append-only recordkeeping along with complex hash signatures to validate every transaction across the decentralized system—without the need for third-party reconciliation [3]. NFT technology expands upon these principles with the addition of nonfungibility, which adds the ability to link unique (nonfungible) assets (art, legal documents, video game purchases, etc.) to a digitally signed entry on a decentralized blockchain network [1]. A major cost of this technology is that performing operations on the blockchain is computationally demanding due to the consensus which must be shared across the distributed network [1, 2, 4, 7]. Thus, a "lightweight" or "slim" system in which some digital assets and transactional computations may be handled in an off-chain location has gained relevancy to address this cost [4, 7].

When considering the implementation of NFT technology in the context of real-world business transactions and movement of assets, it naturally becomes necessary to consider the real-time performance of the technology. Broadly speaking, real-time models of a particular system seek to evaluate the risks of operating within the system at each step and account probabilistically for associated degradation and failures [5]. By incorporating these considerations, a more realistic view of the system may be ascertained since any technological application will be subject to imperfections that will ultimately impede the overall performance. With respect to blockchain/NFT systems, a real-time system should account for the potential for a transaction to be rejected by the network. One method for handling errors is simply to reject the transaction outright and notify the user [6], while other implementations could simply maintain the uncommitted state and retry the transaction [8].

This suite of real-world issues and open research problems constitutes a niche which could be satisfied by a real-time NFT system and a corresponding performance model. This paper will describe a Markovian-queueing-based approach to a real-time, NFT performance model in which transactions can take place both on- and off-chain and (to simulate the real-time nature of the system) each state in the system has a potential for failure when adding a transaction.

The paper will begin with a summary of preliminary research, followed by a description of the new real-time NFT model. Then, results from a MATLAB simulation of the system will be presented alongside the related analysis/conclusions.

II. PRELIMINARIES AND REVIEW

Due to the ability of blockchain networks to record transactions in a decentralized, immutable manner, blockchain has been proposed as an attractive solution for industrial problems in which the movement of assets occurs in a peer-to-peer manner. One such application is recording the movement of real-world items within a supply chain, as proposed by P. Helo and A.H.M. Shamsuzzoha [11]. Because supply chains involve many different parties cooperating in complex, multi-faceted ways, there is no clear central point at which accounting should take place, and a decentralized method of tracking assets is readily applicable. For blockchain to be a viable solution in such a system, however, the technology naturally must be able to meet the real-time performance standards set by the other frameworks used in the system, such as Internet of Things (IoT). Thus, creating a real-time performance model for a blockchain

Authorized licensed use limited to: Oklahoma State University. Downloaded on February 25,2024 at 22:26:03 UTC from IEEE Xplore. Restrictions apply.

network is a necessary step in the refinement, expansion, and widespread implementation of blockchain technology.

A persistent issue faced by blockchain is that of scalability. In Kawase and Kasahara [12], the scalability problem was expressed in analytical terms via a system of differential equations which was generated and then approximately solved by numerical methods to calculate the mean transaction-confirmation time at a variety of block sizes. It was demonstrated that even as the block size increased (thereby increasing the processing capabilities for a given transaction rate), the mean time interval between the incoming and outgoing of a transaction could only remain stable up to a point; the transaction-confirmation time would invariably experience a sharp increase after some critical transaction rate was reached.

Li, et al. expanded upon the work of Kawase and Kasahara by creating a new, continuous-time Markovian model [13]. This model was structured such that it considered the service rates for both the block-generation and blockchain-building steps. The model assumed a Poisson distribution for the arriving transactions, and it utilized a two-dimensional state-transition diagram to account for transactions in the block and in the queue. Ultimately, this work leveraged a matrix representation of the state-transition diagram to provide an analytical solution for the performance of the system (based on the simplifying assumptions). The performance measurements of primary interest were the average number of transactions in the queue and block, and the mean transaction-confirmation time. However, the performance at-scale was not investigated to the same extent as in Kawase and Kasahara.

In Chaparala, et al., a system dubbed "LiftChain" was proposed as a means of managing NFT transactions in a real-time environment [6]. This system utilized an off-chain node that received requests from clients and forwarded them to the on-chain portion of the system upon successful validation of the user's ownership of the asset. If the transaction were to fail on-chain, the process would be terminated, and the user would be notified. This proposed system paid special attention to the safety and security of the NFT system and put forth a novel algorithm for the concurrency of transactions; the on-chain transactions would not necessarily be ordered, so a means of serialization was proposed.

The model presented here primarily has the Variable Bulk Arrival and Static Bulk Service (VBASBS) [9], Real-Time Blockchain [8], and Bivariate NFT Chain [10] models as its preliminaries. Many of the concepts and assumptions from these models are synthesized and expanded upon in this work.

The VBASBS model proposed by Seol, et al. [9] employed an embedded Markovian queueing system to track the arrival of transactions with variable sizes into a block which is to be posted. The transactions were modeled as having a basic (per-slot) arrival rate of λ, which would then be scaled proportionally based on the size of the arriving transaction. The block was assumed to have a maximum size (i.e., number of transactions), and after this size was reached, the block would be posted and purged at a constant service rate, thereby returning the system to its initial state of having zero transactions. The transactions were assumed to arrive according to a Poisson distribution, and the system as a whole was assumed to operate in a steady state. The model was simulated using metrics including average wait time, average number of transactions, and throughput to evaluate how changing the size of the block and the arrival/service rates affected the performance.

The Real-Time Blockchain model presented by Park and Kancharla, et al. [8] expanded upon the variable bulk arrival portion of the VBASBS model, but it changed the posting/purging portion of the system to be variable bulk service (VBS). As a result, the model allowed for the block to be serviced at any number of transactions instead of assuming that the block must be full prior to posting/purging. Additionally, a rate associated with the failure of a transaction to meet its deadline (λ_F) was implemented to reflect the real-time nature of the system. This failure was implemented as a self-loop as opposed to an outright rejection of the transaction or block.

Finally, in the Bivariate NFT Chain model proposed by Seol, et al. [10], a combined on- and off-chain system based on the VBASBS model [9] was described and simulated. This system assumed static service whenever the number of on-chain and off-chain transactions summed to the maximum blocksize, and the system would transition to the first state in whichever on-chain row the service was taking place on [10]. Because both on-chain and off-chain portions of the system were being considered, separate rates for the on-chain and off-chain transactions (λ_{on} and λ_{off}, respectively) were implemented, and simulations were performed varying both the blocksize and the relative proportions of these rates.

III. Proposed Real-Time NFT Chain Model

The proposed Real-Time NFT Chain model combines elements of the previously described models [8, 9, 10] in a novel way to account for the considerations endemic to a real-time NFT system. To elaborate, the maximum-block-size and static-bulk-service assumptions from the VBASBS model [9] are incorporated alongside the bivariate (on- and off-chain) structure of the NFT Chain model [10], and the self-looping failure scenario presented the Real-Time Blockchain model [8] is applied to each internal state. As in these base models, transactions are assumed to arrive in a Poisson distribution, with the rate (on- and off-chain) varying proportionally according to the size of the transaction. As in the NFT Chain model [10], bulk service at a constant rate of μ occurs when the number of on-chain transactions (i) and off-chain transactions (j) in the block sum to the maximum blocksize (n). However, rather than return to the i=0 state for the given j, the block is assumed to be posted and purged all the way back to $P_{0,0}$, thereby simulating a full reset of the block.

A state-transition diagram of the model is presented in Figure 1. By leveraging a steady-state assumption regarding the operation of the system (i.e., that the probability-weighted sum of outgoing transactions equals the probability-weighted sum of incoming transactions), a stochastic analysis via a series of balance equations may be carried out on the model to evaluate its performance.

As with the NFT Chain model, the analysis must account for the possibility of both on-chain and off-chain outgoing transactions at each interior state [10]. Furthermore, per the

Authorized licensed use limited to: Oklahoma State University. Downloaded on February 25,2024 at 22:26:03 UTC from IEEE Xplore. Restrictions apply.

Markovian queueing assumption, all states within the space must sum to 1.0 [9]. Finally, the analysis considers each interior state to have a deadline-miss rate of λ_F in a similar manner to the Real-Time Blockchain model [8]. While the λ_F in the newly proposed NFT model has on- and off-chain components that technically differentiate it from the Real-Time Blockchain model, the constant pairing of these components and the strictly outgoing nature of their occurrence allow a more simplistic consideration of just λ_F rather than its decomposed form.

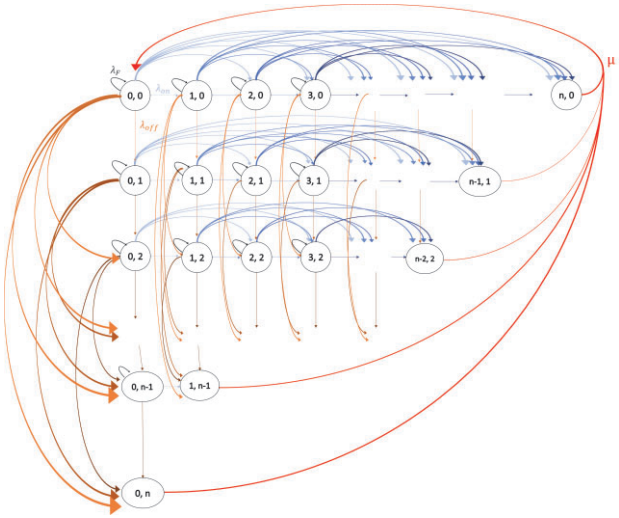

Fig. 1. State-transition diagram for the real-time NFT chain; on-chain transactions occur from left to right, and off-chain transactions occur from top to bottom.

Given the previous discussion regarding the assumptions and transactional flow of the system, the balance equations will be presented accordingly.

$$P_{0,0} = \frac{\mu \sum_{a=0}^{n} P_{n-a,a}}{\left(\frac{n(n+1)}{2}\right)(\lambda_{on} + \lambda_{off}) + \lambda_F} \quad \textbf{(0)}$$

This balance equation (solved in terms of $P_{0,0}$) is marked as Equation (0) since it will not actually factor into the solution for $P_{0,0}$ and the chain as a whole; $P_{0,0}$ will be solved via the Markovian assumption that state space sums to 1.0, as performed in previous work [9, 10]. However, this equation may be used for validation.

$$P_{i,0} = q_{i,0} P_{0,0} \left[\sum_{a=1}^{i} a \left[\sum_{b=1}^{i-a} \left[\prod_{c=b}^{i-a} q_{c,0} \right] b \right] + i \right] \quad \textbf{(1)}$$

Equation (1) models the strictly on-chain (j=0) transactions which correspond to the top row of the state-transition diagram. The $q_{i,0}$ coefficient captures the inverse of the outgoing rates from $P_{i,0}$, and the $q_{c,0}$ coefficient represents the inverse of the outgoing rates at each preceding state. The summation reflects the influence of preceding states on the current state and is directly based on the summation presented in the VBASBS model [9].

$$P_{0,j} = q_{0,j} P_{0,0} \left[\sum_{a=1}^{j} a \left[\sum_{b=1}^{j-a} \left[\prod_{c=b}^{j-a} q_{0,c} \right] b \right] + j \right] \quad \textbf{(2)}$$

Equation (2) models the strictly off-chain (i=0) transactions which correspond to the leftmost column of the state-transition diagram. The $q_{0,j}$ coefficient captures the inverse of the outgoing rates from $P_{0,j}$, and the $q_{0,c}$ coefficient represents the coefficient at each preceding state.

$$P_{i,j} = (q_{i,j})_{on} \left[\sum_{a=1}^{i} a \left[\sum_{b=1}^{i-a} \left[\prod_{c=b}^{i-a} q_{c,j} \right] b \right] + i \right] P_{0,j} + (q_{i,j})_{off} \left[\sum_{a=1}^{j} a \left[\sum_{b=1}^{j-a} \left[\prod_{c=b}^{j-a} q_{i,c} \right] b \right] + j \right] P_{i,0} \quad \textbf{(3)}$$

Equation (3) is the general form by which an interior state may be computed. In fact, Equations (1) and (2) follow directly from Equation (3) when j=0 and i=0, respectively, are used.

$$P_{n-j,j} = (q_{n-j,j})_{on} P_{0,j} \left[\sum_{a=1}^{n-j} a \left[\sum_{b=1}^{n-j-a} \left[\prod_{c=b}^{n-j-a} q_{c,j} \right] b \right] + n - j \right] + (q_{n-j,j})_{off} P_{n-j,0} \left[\sum_{a=1}^{j} a \left[\sum_{b=1}^{j-a} \left[\prod_{c=b}^{j-a} q_{i,c} \right] b \right] + j \right] \quad \textbf{(4)}$$

Equation (4) is also a specialized version of Equation (3) with i=n-j for some j. However, as an additional change, the outgoing coefficient utilizes the service rate μ due to the final states all transitioning back to $P_{0,0}$.

Thus, from Equations (1) and (2), one may observe that all strictly on-/off-chain states may be computed in terms of $P_{0,0}$. Then, all interior and final states ultimately derive from these basis states (i.e., $P_{i,0}$ and $P_{0,j}$). Therefore, these states may be solved in terms of $P_{0,0}$ by extension. Using Equations (1) through (4) and the Markovian queueing constraint, $P_{0,0}$ may be isolated.

$$\sum_{j=0}^{n} \sum_{i=0}^{n-j} P_{i,j} = 1.0$$

$$P_{0,0} + \sum_{i=1}^{n} P_{i,0} + \sum_{j=1}^{n} P_{0,j} + \sum_{j=1}^{n} \sum_{i=1}^{n-j} P_{i,j} = 1.0$$

Factoring out $P_{0,0}$ and dividing both sides by the summations yields the following equation.

$$P_{0,0} = \left[1 + \sum_{i=1}^{n} q_{i,0} \left[\sum_{a=1}^{i} a \left[\sum_{b=1}^{i-a} \left[\prod_{c=b}^{i-a} q_{c,0} \right] b \right] + i \right] + \sum_{j=1}^{n} q_{0,j} \left[\sum_{a=1}^{j} a \left[\sum_{b=1}^{j-a} \left[\prod_{c=b}^{j-a} q_{0,c} \right] b \right] + j \right] + \sum_{j=1}^{n} \sum_{i=1}^{n-j} \left[(q_{i,j})_{on} \left[\sum_{a=1}^{i} a \left[\sum_{b=1}^{i-a} \left[\prod_{c=b}^{i-a} q_{c,j} \right] b \right] \right] + i \right] q_{0,j} \left[\sum_{a=1}^{j} a \left[\sum_{b=1}^{j-a} \left[\prod_{c=b}^{j-a} q_{0,c} \right] b \right] + j \right] + (q_{i,j})_{off} \left[\sum_{a=1}^{j} a \left[\sum_{b=1}^{j-a} \left[\prod_{c=b}^{j-a} q_{i,c} \right] b \right] \right] + j \right] q_{i,0} \left[\sum_{a=1}^{i} a \left[\sum_{b=1}^{i-a} \left[\prod_{c=b}^{i-a} q_{c,0} \right] b \right] + i \right] \right]^{-1} \quad \textbf{(5)}$$

Authorized licensed use limited to: Oklahoma State University. Downloaded on February 25,2024 at 22:26:03 UTC from IEEE Xplore. Restrictions apply.

It should be noted that $(q_{i,j})_{on}$ and $(q_{i,j})_{off}$ will vary according to whether an interior state (which uses Equation (3)) or a final state (which uses Equation (4)) is being computed, as will $q_{i,0}$ and $q_{0,j}$ based on whether i or j equals n.

Via Equation (5), $P_{0,0}$ can be solved using the coefficients of all the successive states. The successive states, in turn, can then be solved using this value of $P_{0,0}$ in their respective equations.

IV. Simulation and Results

The equations presented here were simulated in MATLAB while adjusting a variety of parameters to gain insights into the average system wait time (W), average number of transactions (L), and throughput (γ), with these metrics defined via the following equations and based on the metrics presented in previous models [8, 9, 10].

Block Metrics

The block metrics pertain to the performance of the system prior to the process of posting a block. The average numbers of on-chain and off-chain transactions are calculated by multiplying the stepwise number of on-chain or off-chain transactions by the probability associated with that step and summing over the state space.

$$(L_Q)_{on} = \sum_{j=0}^{n}\sum_{i=0}^{n-j} iP_{i,j}$$

$$(L_Q)_{off} = \sum_{j=0}^{n}\sum_{i=0}^{n-j} jP_{i,j}$$

The average block wait times are then calculated by dividing the average number of on-chain/off-chain transactions by the respective transaction rates.

$$(W_Q)_{on} = \frac{(L_Q)_{on}}{\lambda_{on}}$$

$$(W_Q)_{off} = \frac{(L_Q)_{off}}{\lambda_{off}}$$

System Metrics

The system metrics are calculated by combining the block metrics with the bulk service rate (μ). For the average system wait time, this entails summing the average on-chain and off-chain wait times and adding the reciprocal of μ.

$$W = (W_Q)_{on} + (W_Q)_{off} + \frac{1}{\mu}$$

Next, the throughput γ may be calculated by multiplying the probability associated with each final state $P_{n-j,j}$ by μ and summing these values.

$$\gamma = \sum_{j=0}^{n} \mu P_{n-j,j}$$

Finally, the average number of transactions in the system may be calculated by multiplying the throughput by the average system wait time.

$$L = \gamma W$$

Simulation 1: Varying $\lambda_{on}/\lambda_{off}$ Ratios

First, the effect of varying λ_{on} with a static λ_{off} and a static $\lambda_F = 0$ was investigated. This was carried out at $\lambda_{on}/\lambda_{off}$ ratios from 0.0001 up to 10.0 with low (*L*), medium (*M*), and high (*H*) static λ_{off} values.

For W, L, and γ:

Low……..λ_{off} = 0.005
Medium…λ_{off} = 0.01
High…….λ_{off} = 0.05

The results for the average system wait time (W) are shown in Figure 2. It may be observed that as λ_{on} outpaces λ_{off}, the average wait time for the system decreases, and the larger the static λ_{off} is, the lower the average wait time is throughout, as would be expected.

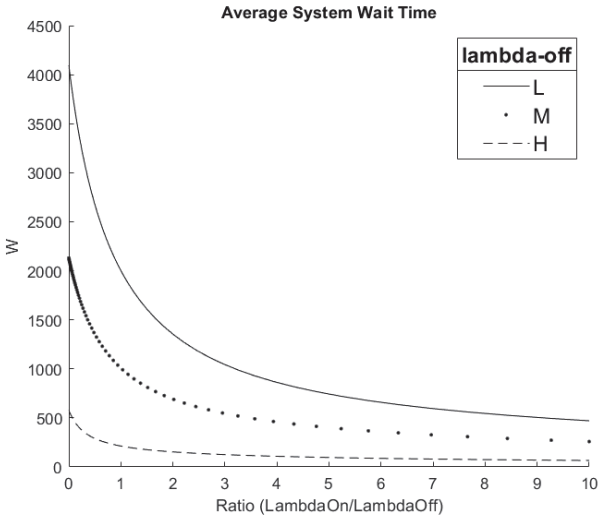

Fig. 2. Average System Wait Time (W) with increasing $\frac{\lambda_{on}}{\lambda_{off}}$ ratio.

The results for the average number of transactions (L) are shown in Figure 3. As with W, L decreases as λ_{on} grows and as the static λ_{off} increases.

Authorized licensed use limited to: Oklahoma State University. Downloaded on February 25,2024 at 22:26:03 UTC from IEEE Xplore. Restrictions apply.

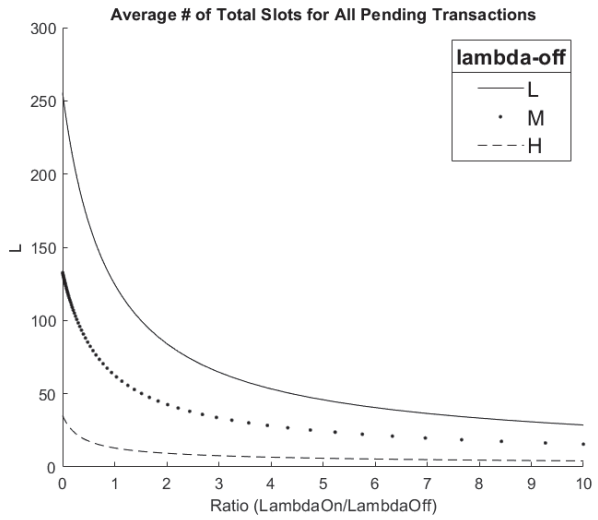

Fig. 3. Average Number of Transactions (L) with increasing $\frac{\lambda_{on}}{\lambda_{off}}$ ratio.

The results for the throughput (γ) are shown in Figure 4. This metric behaves in a less intuitive manner, as it sharply decreases as λ_{on} initially increases, then sharply increases after a certain point. This is a reflection of the probability distribution across the state space with changing $\frac{\lambda_{on}}{\lambda_{off}}$ ratios; increasing λ_{on} makes transactions more likely to be in intermediate states at first, which decreases the overall throughput. Then, once λ_{on} becomes sufficiently large, the probability of being in a bulk-service state grows again.

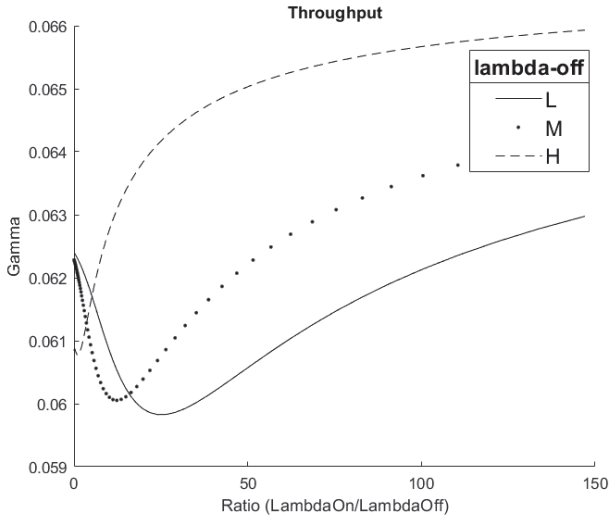

Fig. 4. Throughput (γ) with increasing $\frac{\lambda_{on}}{\lambda_{off}}$ ratio.

Simulation 2: Varying μ

Next, the effect of varying μ with a static λ_{on} and a static $\lambda_F = 0$ was investigated using μ from 1/15 up to 1/5 with low (L), medium (M), and high (H) static λ_{on} values.

For W and L:

Low……..λ_{on} = 0.005
Medium…λ_{on} = 0.00625
High…….λ_{on} = 0.0075

For γ:

Low……..λ_{on} = 0.005
Medium…λ_{on} = 0.1
High…….λ_{on} = 2.0

Separate rates were used for the throughput to accommodate the lower sensitivity of this metric to λ_{on}.

The results for the average system wait time (W) are shown in Figure 5. As μ increases, W decreases gradually for a given λ_{on}.

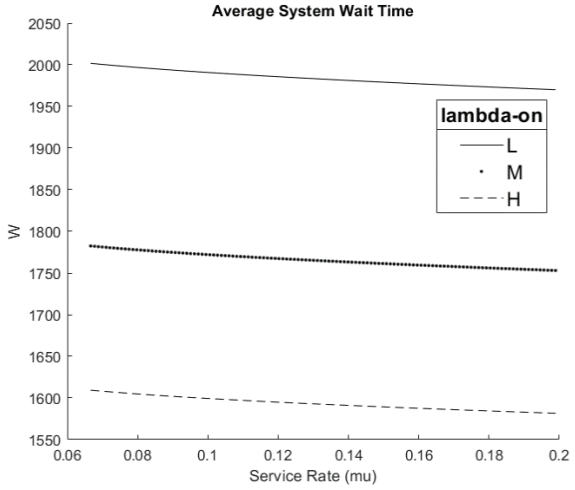

Fig. 5. Average System Wait Time (W) with increasing μ.

The results for the average number of transactions (L) are shown in Figure 6. L increases considerably as μ grows since this allows more transactions to be in the system as a whole.

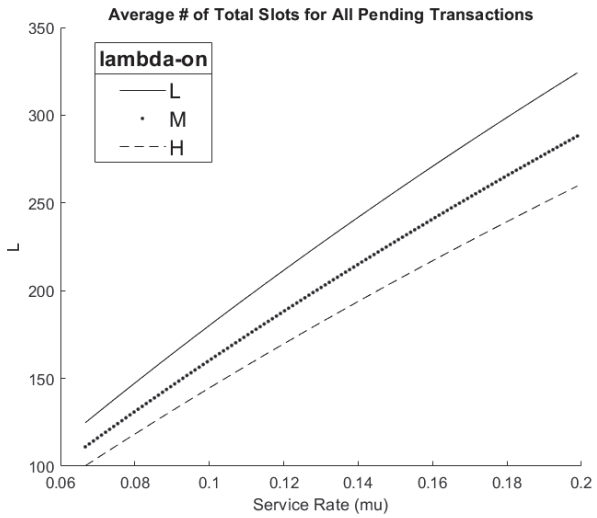

Fig. 6. Average Number of Transactions (L) with increasing μ.

Authorized licensed use limited to: Oklahoma State University. Downloaded on February 25,2024 at 22:26:03 UTC from IEEE Xplore. Restrictions apply.

The results for the throughput (γ) are shown in Figure 7. Again, this metric behaves in a less intuitive manner; although the throughput increases with increasing μ as anticipated, the medium λ_{on} value has lower γ than the low λ_{on} for all μ. Then, the high λ_{on} value has higher γ than both the low and medium λ_{on}. As in the first simulation, this is likely due to the medium λ_{on} reducing the off-chain traffic and the high λ_{on} re-establishing the higher bulk-service-state traffic.

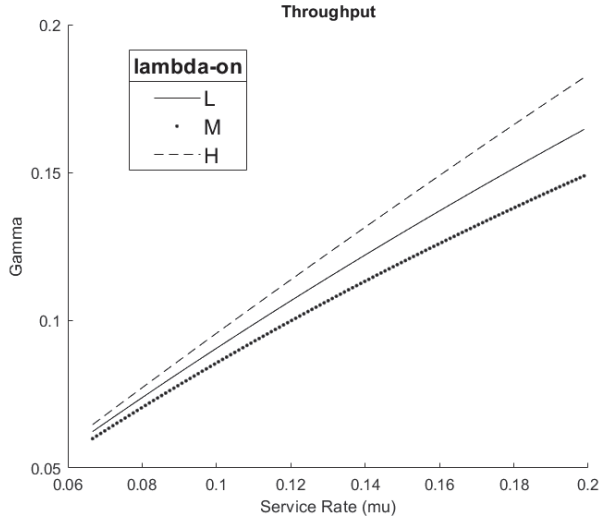

Fig. 7. Throughput (γ) with increasing μ.

Simulation 3: Varying Blocksize

Next, the effect of varying n with a static λ_{off} and a static $\lambda_F = 0$ was investigated using n from 10 up to 109 with low (L), medium (M), and high (H) static λ_{off} values.

For W and L:
Low……..λ_{off} = 0.005
Medium…λ_{off} = 0.01
High…….λ_{off} = 0.05

For γ:
Low……..λ_{on} = 0.005
Medium…λ_{on} = 0.1
High…….λ_{on} = 2.0

Separate rates were used for the throughput to illustrate the same trends observed in Simulation 2.

The results for the average system wait time (W) are shown in Figure 8. As n increases, W increases. The rate of increase decreases with a larger given λ_{off}.

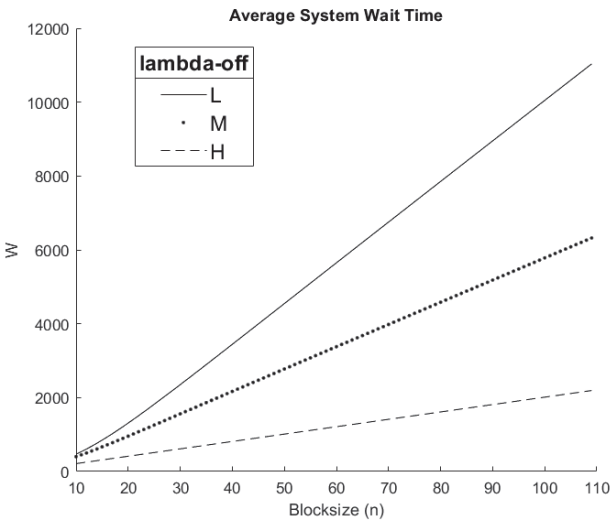

Fig. 8. Average System Wait Time (W) with increasing n.

The results for the average number of transactions (L) are shown in Figure 9. L increases as n grows, and the rate of increase decreases with a larger given λ_{off}, as with W.

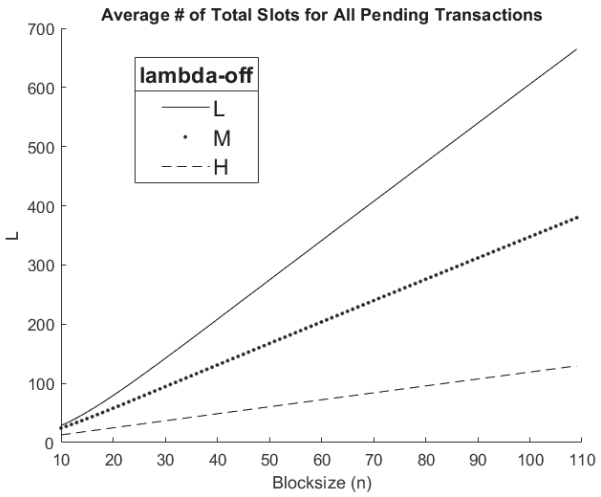

Fig. 9. Average Number of Transactions (L) with increasing n.

The results for the throughput (γ) are shown in Figure 10. The series plot in the same order as in Simulation 2; the overall throughput is lower for the medium λ_{off} case compared to the low λ_{off} case, and the overall throughput is higher in the high λ_{off} case than for the low λ_{off} case.

Authorized licensed use limited to: Oklahoma State University. Downloaded on February 25,2024 at 22:26:03 UTC from IEEE Xplore. Restrictions apply.

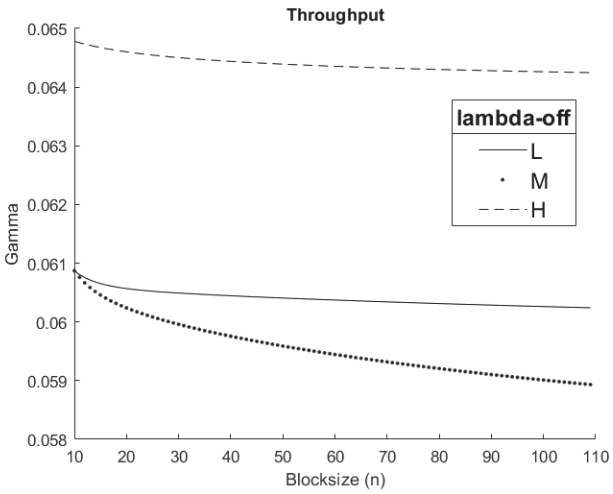

Fig. 10. Throughput (γ) with increasing n.

Simulation 4: Varying On-Chain Transactions (i)

The following simulation investigates the effect of which column in the state space is being analyzed on the individual throughput associated with the column. This was done using a maximum blocksize n = 109, λ_{off} = 0.005, and λ_F = 0.

Low……..λ_{on} = 0.005
Medium…λ_{on} = 0.01
High…….λ_{on} = 0.05

The results for the column-specific throughput (γ_i) are shown in Figure 11. The throughput increases sharply at i = 0 and i = n and decreases for the middle states, with local maxima near the upper and lower bounds. The sharp increase at these bounds is likely an edge effect caused by the direct proximity of $P_{0,0}$ for the strictly on- and off-chain states.

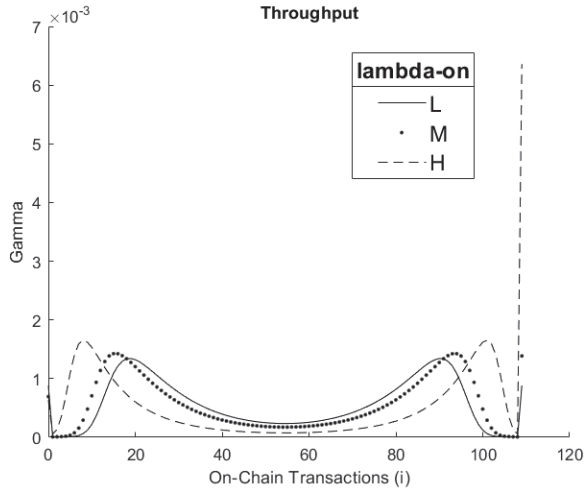

Fig. 11. Throughput (γ_i) versus the number of off-chain transactions (i).

Simulation 5: Varying Deadline-Miss Rate (λ_F)

The following simulation investigates the effect of changing λ_F on W, L, and γ for a static λ_{off} = 0.005, n = 10, and various static λ_{on}.

Low……..λ_{on} = 0.005
Medium…λ_{on} = 0.01
High…….λ_{on} = 0.05

The results for the average system wait time (W) are shown in Figure 12. It may be observed that the average wait time is not significantly affected by λ_F up to 0.09.

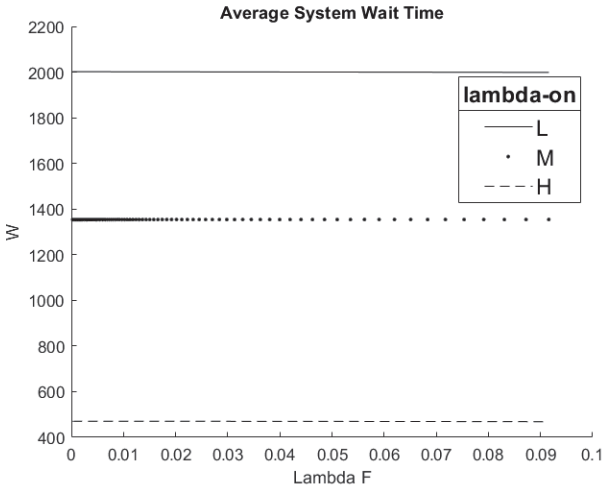

Fig. 12. Average System Wait Time (W) with increasing λ_F.

The results for the average number of transactions in the system (L) are shown in Figure 13. It may be observed that this metric is also not significantly affected by λ_F up to 0.09.

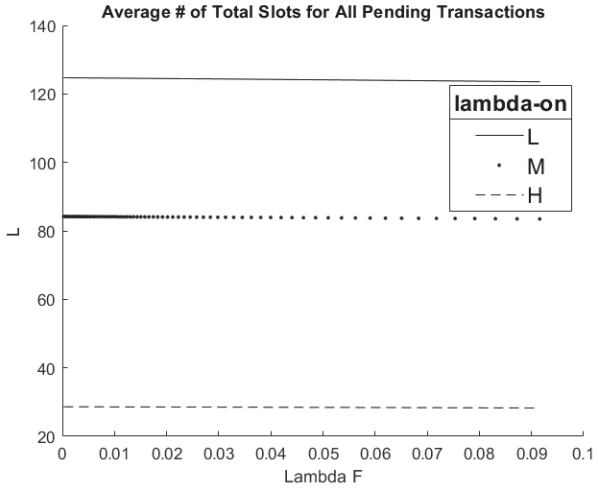

Fig. 13. Average Number of Transactions (L) with increasing λ_F.

The results for the throughput (γ) are shown in Figure 14. Throughput decreases as λ_F increases up to 0.09.

Authorized licensed use limited to: Oklahoma State University. Downloaded on February 25,2024 at 22:26:03 UTC from IEEE Xplore. Restrictions apply.

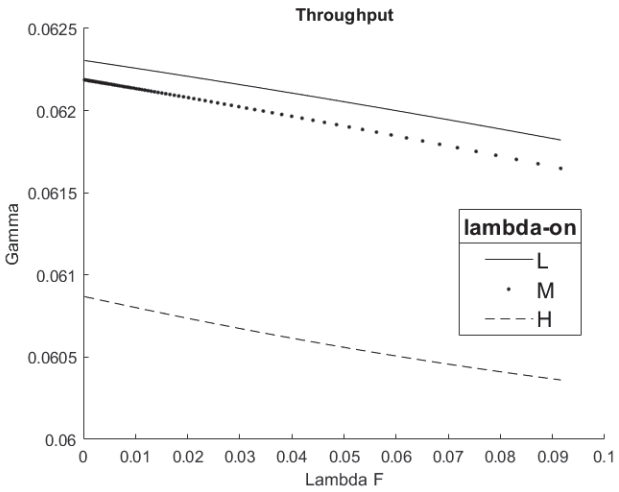

Fig. 14. Throughput (γ) with increasing λ_F.

V. Conclusion and Discussion

This paper has presented a Markovian-queueing-based theoretical model for modeling the real-time operation of an NFT system. The model expands upon previous work done on blockchain [9], real-time blockchain [8], and bivariate on- and off-chain NFT [10] systems and proposes a novel formulation for the probabilities across the associated state space that considers on-chain transaction rate, off-chain transaction rate, and deadline-miss rate. The performance of the theoretical model was tested using a variety of metrics, and it was demonstrated that throughput generally performed better when the on-chain and off-chain transaction rates were sufficiently different such that traffic was more likely to be routed towards a final processing state. The deadline-miss rate, meanwhile, was shown not to have a significant impact on the performance given the range of miss rates tested. The proposed model performed as expected for average wait time and average number of transactions across different testing methodologies. Overall, this model serves as a reasonable baseline upon which more extensive, real-world testing may be performed to evaluate the real-time performance of a bivariate NFT system.

References

[1] W. Rehman, H. E. Zainab, J. Imran, and N. Bawany, "NFTs: Applications and Challenges", 22nd International Arab Conference on Information Technology 2021, https://doi.org/10.1109/ACIT53391.2021.9677260.

[2] Q. Wang, R. Li, Qi Wang, and S. Chen, "Non-Fungible Token (NFT): Overview, Evaluation, Opportunities and Challenges", CoRR 2021, https://arxiv.org/abs/2105.07447.

[3] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", Satoshi Nakamoto Institute, 2008, https://nakamotoinstitute.org/bitcoin/.

[4] M. Allouche, T. Frikha, M. Mitrea, G. Memmi, and F. Chaabane, "Lightweight Blockchain Processing. Case Study: Scanned Document Tracking on Tezos Blockchain", MDPI Appl. Sci. 2021, https://doi.org/10.3390/app11157169.

[5] H. Lu, W. J. Kolarik, S. S. Lu, "Real-Time Performance Reliability Prediction", IEEE Transactions on Reliability, Vol. 50, No. 4, December 2001, https://doi.org/10.1109/24.983393.

[6] H. K. Chaparala, S. V. Doddala, A. Showail, A. Singh, S. Gazzaz, and F. Nawab, "LiftChain: A Scalable Multi-Stage NFT Transaction Protocol", 2022 IEEE International Conference on Blockchain, https://doi.org/10.1109/Blockchain55522.2022.00057.

[7] A. Kancharla, J. Seol, and N. Park , "Slim Chain and Dependability'', ACM BSCI 2020.

[8] N. Park, A. Kancharla, and H.-Y. Kim, "A Real-Time Chain and Variable Bulk Arrival and Variable Bulk Service (VBAVBS) Model with λ_F", MDPI Appl. Sci. 2020.

[9] J. Seol, A. Kancharla, J. Ke, H. -Y. Kim and N. Park, "A Variable Bulk Arrival and Static Bulk Service Queueing Model for Blockchain", ACM BSCI 2020.

[10] J. Seol, J. Ke, A. Kancharla, S. Joshi and N. Park, "A Bivariate Performance Model across On- and Off-Chain in A NFT (Non-Fungible Token) Chain", BCCA 2022.

[11] P. Helo, A.H.M. Shamsuzzoha, "Real-time supply chain—A blockchain architecture for project deliveries," Robotics and Computer-Integrated Manufacturing, Vol. 63 (2020) 101909, https://doi.org/10.1016/j.rcim.2019.101909.

[12] Y. Kawase, S. Kasahara, "Transaction-Confirmation Time for Bitcoin: A Queueing Analytical Approach to Blockchain Mechanism," (2017) 75-88, https://doi.org/10.1007/978-3-319-68520-5_5.

[13] Q. Li, J. Ma, Y. Chang, "Blockchain Queue Theory," 7th International Conference, CSoNet 2018, Shanghai, China, December 18–20, 2018, Proceedings, https://doi.org/10.1007/978-3-030-04648-4.

Authorized licensed use limited to: Oklahoma State University. Downloaded on February 25,2024 at 22:26:03 UTC from IEEE Xplore. Restrictions apply.