A Performance Analysis of Priority Queuing on Distributed Decentralized Chain Systems with k-Queue Variable Bulk Arrival and Static Bulk Service model

CS-5113: Computer Organization and Architecture
Assignment 1 – Group 3

Raghu Medarametla, Sai Vishwas Pathem, Kalyan Kumar Danda, Rakesh Chowdary Polina, Hareesh Namani, Divya Sravani Bandi
(Code for Matlab simulations: GitHub - Kalyankumarreddy369/COA: COA group-3)
Department of Computer Science, Oklahoma State University, Stillwater, OK 74078, USA.
(Everyone Contributed Equally)

ABSTRACT:

This report proposes a novel approach to address the queuing model present in the k - Variable Bulk Arrival of Transactions and Static Bulk Service (k -VBASBS) system, which is associated with contemporary blockchain technology of Embedded Markovian single server exponential queueing system and provides various system performance evaluations. Our new method proposes a priority for the transactions using a decaying function that considers transactions arriving in bulk, with the queueing being dependent on the transaction arrival size and transaction fees. Queuing process is performed in a static manner once a certain number of slots are occupied. Traditionally, the First in First Out (FIFO) approach is assumed in the K-queue model. However, our mathematical implementation of equations on existing literature and thorough analysis led to the development of a priority model that assumes to enhance the performance of a K-VBASAS. Priority-based transaction processing systems have been widely studied in the literature, and their primary goal is to efficiently process transactions based on their priorities. This system aims to optimize various performance metrics, such as minimizing response time or maximizing throughput, while preventing starvation of lower-priority transactions.

INTRODUCTION:

Blockchain technology is a newly adaptive technology which has many benefits and gives the solution for long wait time for most of the applications, including cryptocurrencies, supply chain management, and more. With the decentralization of digital ledgers, it makes it possible for transparent recording of transactions and security enhancement. In simple terms decentralization is nothing but a network of computers that works together to verify each transaction by adding it to blocks as proof of-work.

The Distributed Decentralized Chain (DDC) is a specific type of Distributed Decentralized Ledger (DLT) in which the entire blockchain data is stored across a limited number of nodes (k nodes) rather than each node within the blockchain network. In this setup, the total storage requirement is represented by N*B, where B signifies the overall number of blocks in the entire blockchain, and this equals the combined storage slots among the selected k nodes. Within this context, we consider two different transaction posting rates, $\mu_{inter\text{-node}}$ and $\mu_{intra\text{-node}}$, which respectively represent the communication rate within a single node and between two separate nodes in the k distributed nodes. It is important to note that the $\mu_{\text{intra-node}}$ rate is always higher than the $\mu_{inter\text{-node}}$ rate.

This k-VBASBS is based on embedded Markovian queueing model of type $\boldsymbol{M^{1,\,n}/M^{n}/1}$. In k-VBASBS model the block of transaction arrives at a rate of λ, in which the state is represented by (i,k), where i is the number of slots ranging from 0 to n in a block on the present node, and k is the number of distributed nodes to store a entire copy of a blockchain. Without loss of generality and practicality, it is assumed that there are two different transaction posting rates assumed to consider the overhead of inter-node (i.e., $\mu_{inter\text{-node}}$) control hopping versus the one of the original intra-node posting rate (i.e., $\mu_{intra\text{-node}}$), and $\mu_{inter\text{-node}} \ll \mu_{intra\text{-node}}$.

In this given model on Distributed Decentralized Chain Systems with $\boldsymbol{k}$-Queue Variable Bulk Arrival and Static Bulk Service, each queue is independent of each other. The proposed priority sorts the transactions in the queue accordingly with the priority to be sent for next stage of verification. The report is structured as follows: Preliminaries and review based on our work on model discussed in the original seed paper. Followed by the newly proposed model concern with k-VBASBS model. A section with numerical stimulations for k-VBASBS with priority queueing for transaction processing and then conclusions are discussed in the later sections.

PRELIMINARIES AND REVIEW:

In the proposed k- Queue Variable Bulk and Static Bulk Service (k-VBASAS) model, the service time for a bulk arrival of transactions in a queue is a key factor in determining the performance and optimization of the system. Even though the service time also depends on the specific implementation and properties of the underlying DDC system, this service time decides the throughput of the blockchain system.

Even though the DDC implementation for different solutions of crypto would be benefited from the on-off, slim, real-time and hybrid features and performance and dependability, the dependability or performance models cannot adequately address the stochastic nature of the transactions flow in the queue or bulk posting it for the verification. This transaction flows affect the throughput of the derived k-VBASAS system by increasing the average time a transaction spent in the system E[T] or W and increases the average number of transactions in the system E[N]. E[N] is defined as the average number of transactions that are present in the system at any given time, including both those waiting in the queue and those being serviced. E[N] is a measure of the system's ability to handle incoming transactions and is another important performance metric. E[T] is defined as the average time that a transaction spends in the system, from the moment it arrives to the moment it is confirmed. This includes both the time spent waiting in the queue and the time spent being serviced. E[T] is a measure of the system's responsiveness to incoming transactions and is an important performance metric in many applications.

It is assumed that the service time is exponential at $1/\mu$ when the server is serving the entire queue (e.g., equivalently, posting and purging the entire queue). Without loss of generality, it is assumed that customers arrive at an exponential rate of λ. The service time distribution for a DDC chain model is given as

$$f(s) = \mu * e^{(-\mu * s)}$$

Since service time is exponential at $1/\mu$ at i^{th} slot the mean service time is given as:

$$\frac{1}{\mu_i} = \int^{s} f(s)ds = \frac{1}{\mu} * \left(1 - e^{(-\mu t_i)}\right)$$

Where t_i is expected time spent in the system.

$$E[T] \; or \; W \; = \sum_{i=0}^{n} \frac{t_i}{\mu_i} * P_i$$

where Pi is the steady-state probability of a transaction being in queue i.

$$E[N] \; or \; L \; = \sum_{i=1}^{N} \lambda_i * t_i$$

A longer mean transaction-confirmation time E[T] typically indicates that transactions are spending more time in the system, which can reduce the system's throughput. Similarly, a larger average number of transactions in the system E[N] can indicate that the system is becoming congested, which can also reduce throughput. In this Embedded Markovian single server exponential queueing system, it is assumed that the transactions are arriving in FIFO. If a transaction with highest importance arrives at the last in a queue, it is being starved all the time until the execution of all those before. So, we proposed a new method to address this issue which is assigning priority to the queue based on the transaction age which will calculate exponentially with a decaying function.

Implementation is as follows:

1. Assume an initial state where all priority queues are empty, i.e., P(i,j)=0 for every i, j.

2. Let a transaction T arrive with initial priority $p(T)$ and age $t_a = 0$. Without loss of generality, we can assume that the transaction belongs to a queue with d = 1.

3. The transaction is inserted into priority queue 1, and the corresponding priority value p' and γ(i,j) are set based on its age, which is 0 at arrival.

4. As time progresses, the age of the transaction T increases. Periodically, we can recalculate p' (p,t_a) for each transaction in the queue and accordingly update the priority queues and the variables γ(i,j) in the model.

5. When the age of transaction T increases, its effective priority p' increases, leading to a higher chance of T getting processed by the system.

6. Since the priority values are adjusted to consider the age of the transaction, the chance of a higher-priority transaction starving a lower-priority transaction decreases, ensuring that all transactions are served in a linear mean time.

A priority queue is formed from the above methodology and analysis has been done. All the observations are provided in the below section of simulations with graphs. The below section of proposed model describes the implementation of priority on the decaying function.

PROPOSED MODEL:

Upon conducting extensive analysis, we have gleaned that the embedded Markovian queueing model expounded in the paper employs a priority-based technique. This technique involves arranging transactions in the queue based on their priority. The State Transition Probability Algorithm implemented in the model utilizes a priority-based queuing mechanism.

Transactions are received in batches of varying sizes and are subsequently appended to the queue. As soon as a slot becomes available, the transaction positioned at the head of the queue (i.e., the one with the highest priority) is processed and then added to the pool of mined transactions. In cases where multiple transactions possess identical size, the transaction fee serves as the basis for processing.

The mechanism here explicated, advocates for the speedy processing of transactions. This can be advantageous in blockchain systems where larger transactions may have greater worth or hold more considerable influence over the network's performance, as opposed to smaller transactions.

Our algorithm accounts various priority levels which significantly affect the transaction processing time and waiting times for users. Transactions can be assigned to different priority levels based on various criteria, and separate service parameters, including strict priority, and age-weighted priority, can be introduced to determine the order in which transactions in different priority levels are processed. Strict priority prioritizes high-priority transactions, leading to reduced waiting times. This mechanism also accounts for preventing starvation of low-priority transactions through the age-weighted priority, that considers both the priority level and waiting time, ensuring low-priority transactions are eventually processed and giving priority to high-priority transactions that have been waiting for longer. Overall, the selection of a priority policy can have a significant impact on the efficiency and fairness of transaction processing in a system.

Incorporating the priority aging function into the k-VBASBS model requires modifying the states, state transition rates, assumptions, and state transition probabilities to account for the new priority values. Here's a detailed breakdown of the necessary changes:

Modified states: The modified states will now include an additional parameter representing the age of the transactions in the queue. Therefore, the new states are defined as:

$P_{0,1,ta}$: No transaction arrived in the queue yet, with one distributed node to store a chain of a block and transaction age t_a.
$P_{n,k,ta}$: n slots arrived in the queue for the posting in the block, with k distributed nodes and transaction age t_a.

$P_{i,k,ta}$: i slots (where $0 < i < n$) arrived in the queue for the posting in the block, with k distributed nodes and transaction age t_a.
Random variables: for state transition rates: The state transition rates will now depend on the updated priority value p':

$$\lambda' = \lambda(p')$$

is the rate for a slot of a transaction with priority p' to arrive.

$$\mu_a(p')$$

is the rate for the slots of the transactions in the entire queue with priority p' to be posted and purged for the same number of distributed nodes.

$$\mu_e(p')$$

is the rate for the slots of the transactions in the entire queue with priority p' to be posted and purged for a different number of distributed nodes.

Assumptions: In addition to the existing assumptions, we now assume that:

1. The aging factor a, maximum increase in priority b, and priority increase rate c are positive constants.
2. The priority value p' for each transaction is updated periodically based on its age t_a.

State transition probabilities: The modified state transition probabilities are defined as:

$$t\big((i, d, t_a) \to (j, d, t_a + \Delta t)\big) = (j - i) * \lambda\big(p'(p, t_a)\big),$$

where $0 \le i < j \le i$, $1 \le d \le k$, and Δt is the time increment.

$$t\big((n, d, t_a) \to (0, d, t_a + \Delta t)\big) = \mu_a\big(p'(p, t_a)\big)$$

$$t\big((n, d, t_a) \to (0, d + 1, t_a + \Delta t)\big) = \mu_e\big(p'(p, t_a)\big), \text{ where } d + 1 \le k.$$

The k-VBASBS model with priority aging will require more complex calculations and additional tracking of transaction ages. However, this modification ensures that lower-priority transactions are not starved and receive a fair chance of being processed as they spend more time in the queue.

To incorporate the new priority function

$$p'(p, t_a) = p - a * f(t_a),$$

where

$$f(t_a) = b * \left(1 - e^{(-ct_a)}\right),$$

and $p = \frac{s}{f}$,

where s is the size of the transaction and f is the transaction fees, into the steady state equations, we need to replace the origin, priority p with the updated priority p' in the random variables for state transition rates.

Modified random variables for state transition rates:

$$\lambda' = \lambda(p')$$

is the rate for a slot of a transaction with priority p' to arrive.

$$\mu_a(p')$$

is the rate for the slots of the transactions in the entire queue with priority p' to be posted and purged for the same number of distributed nodes.

$$\mu_e(p')$$

is the rate for the slots of the transactions in the entire queue with priority p' to be posted and purged for a different number of distributed nodes.

Modified steady state equations: With the updated priority p', the steady state equations for the k-VBASBS model will be as follows:

Row 1:

$$P_{0,1}(\lambda' + 2\lambda' + 3\lambda' + 4\lambda' + \cdots + (n-1)\lambda' + n\lambda') = \mu_a(p')P_{n,1}$$

$$P_{0,1}(1 + 2 + 3 + 4 + \cdots + (n-1) + n)\lambda' = \mu_a(p')P_{n,1}$$

$$P_{0,1}\left(\frac{(n)(n+1)}{2}\right)\lambda(p') = \mu_a(p')P_{n,1}$$

Steady state equations for state $\boldsymbol{P_{1,1}}$ are expressed as follows:

$$P_{1,1}(\lambda' + 2\lambda' + 3\lambda' + 4\lambda' + \cdots + (n-1)\lambda') = \lambda' P_{0,1}$$

$$P_{1,1}\big(1 + 2 + 3 + 4 + \cdots + (n-1)\big)\lambda' = \lambda' P_{0,1}$$

$$P_{1,1}\left(\frac{\big(n(n-1)\big)}{2}\right)\lambda' = \lambda' P_{0,1}$$

Generalizing for the entire first row:

$$P_{i,1}(\lambda' + 2\lambda' + 3\lambda' + 4\lambda' + \cdots + (n-i)\lambda') = \lambda' P_{i-1,1} + 2\lambda' P_{i-2,1} + 3\lambda' P_{i-3,1} + \cdots + i\lambda' P_{0,1}$$

$$P_{i,1}\big(1 + 2 + 3 + 4 + \cdots + (n-i)\big)\lambda' = \lambda' P_{i-1,1} + 2\lambda' P_{i-2,1} + 3\lambda' P_{i-3,1} + \cdots + i\lambda' P_{0,1}$$

$$P_{i,1}\left(\frac{(n-i)(n-i+1)}{2}\right)\lambda' = \lambda' P_{i-1,1} + 2\lambda' P_{i-2,1} + 3\lambda' P_{i-3,1} + \cdots + i\lambda' P_{0,1},$$

where $1 \leq i \leq$ n

Last state in the first row:

$$P_{n,1}\big(\mu_a(p') + \mu_e(p')\big) = \lambda' P_{n-1,1} + 2\lambda' P_{n-2,1} + 3\lambda' P_{n-3,1} + \cdots + n\lambda' P_{0,1}$$

Row k:

$$P_{0,k}(\lambda' + 2\lambda' + 3\lambda' + 4\lambda' + \cdots + (n-1)\lambda' + n\lambda') = \mu_a(p')P_{n,k} + \mu_e(p')P_{n,k-1}$$

$$P_{0,k}(1 + 2 + 3 + 4 + \cdots + (n-1) + n)\lambda' = \mu_a(p')P_{n,k} + \mu_e(p')P_{n,k-1}$$

$$P_{0,k}\left(\frac{(n)(n+1)}{2}\right)\lambda' = \mu_a(p')P_{n,k} + \mu_e(p')P_{n,k-1}$$

Steady state equations for row $\boldsymbol{P_{1,k}}$ are expressed as follows:

$$P_{1,k}(\lambda' + 2\lambda' + 3\lambda' + 4\lambda' + \cdots + (n-1)\lambda') = \lambda' P_{0,k}$$

$$P_{1,k}\big(1 + 2 + 3 + 4 + \cdots + (n-1)\big)\lambda' = \lambda' P_{0,k}$$

$$P_{1,k}\left(\frac{\big(n(n-1)\big)}{2}\right)\lambda' = \lambda' P_{0,k}$$

Continuing with the modified steady state equations for the k^{th} row:

$$P_{j,k}(\lambda' + 2\lambda' + 3\lambda' + 4\lambda' + \cdots + (n-j)\lambda') = \lambda' P_{j-1,k} + 2\lambda' P_{j-2,k} + 3\lambda' P_{j-3,k} + \cdots + j\lambda' P_{0,k}$$

$$P_{j,k}\big(1 + 2 + 3 + 4 + \cdots + (n-j)\big)\lambda' = \lambda' P_{j-1,k} + 2\lambda' P_{j-2,k} + 3\lambda' P_{j-3,k} + \cdots + j\lambda' P_{0,k}$$

$$P_{j,k}\left(\tfrac{(n-j)(n-j+1)}{2}\right)\lambda' = \lambda' P_{j-1,k} + 2\lambda' P_{j-2,k} + 3\lambda' P_{j-3,k} + \cdots + j\lambda' P_{0,k},$$

where $1 \leq j \leq n$

Last state in the k^{th} row:

$$P_{n,k}\mu_a(p') = \lambda' P_{n-1,k} + 2\lambda' P_{n-2,k} + 3\lambda' P_{n-3,k} + \cdots + n\lambda' P_{0,k}$$

To solve these modified steady state equations, you would need to follow a similar procedure as with the original steady state equations. Keep in mind that the new priority function $p'(p, t_a)$ and the aging factor $f(t_a)$ are now part of the transition rates. This will require tracking the age of each transaction and updating the priority accordingly.

The final equation is where all the steady state probabilities sum up to 1:

$$\sum_{i=0}^{n}\sum_{j=0}^{k} P_{i,j} = 1$$

The following are a few performance measurements of primary interests in proposed model.

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).

$$L_Q = \sum_{i=0}^{n}\sum_{j=0}^{k}(i+j)\big(P_{i,j}\big)$$

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

$$W_Q = \frac{L_Q}{\lambda'}$$

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

$$W = W_Q + \frac{1}{\mu_{inter}}$$

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).

$$L = \lambda' W$$

γ: the proportion of time that the system is busy serving transactions (i.e., throughput).

$$\gamma = \mu P_{n,k}$$

SIMULATION AND ANALYSIS:

The Priority Queuing on Distributed Decentralized Chain System with VBASBS model is tested and verified through numerical analysis using MATLAB to determine the Steady State probabilities of the equations and the Lq, Wq, W, and γ versus k (i.e., the number of distributed nodes), λ' (i.e., the rate at which a transaction slot arrives), and 1/ µ (i.e., block service time).

Figure 1 plots the W, which is the average time a customer spends (or, equivalently, a transaction) in the system (i.e., the blockchain's transaction pool) in the decentralized distributed chain model involving K distributed nodes.

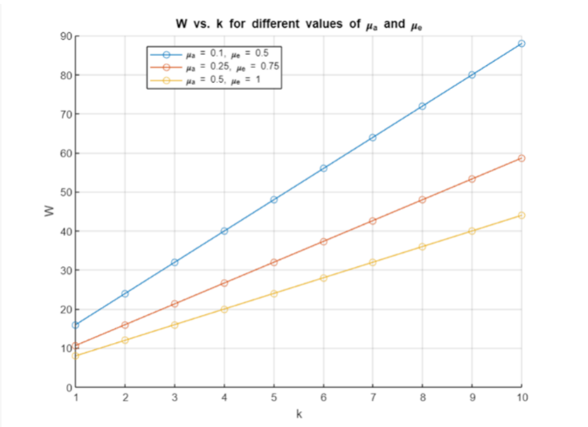


Figure 1: W vs. k for DDC

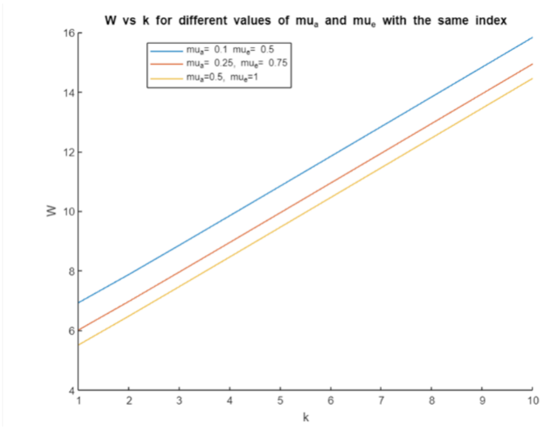
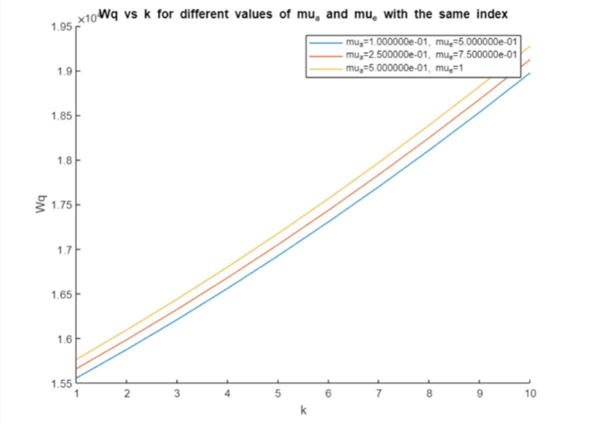


Figure 1.1: W vs. k for priority queuing.

Here, the average time spent by a transaction in DDC for different µ-rates is plotted. We can see that the W value for the same ratios, shows a decreasing trend for model with priority queuing. Therefore, the waiting time is reduced.

Figure 2 plots the graph for variable Wq which is the average amount of time a customer (i.e., equivalently, a transaction) is in the queue (i.e., the block currently being mined) in the decentralized distributed chain model involving K distributed nodes.



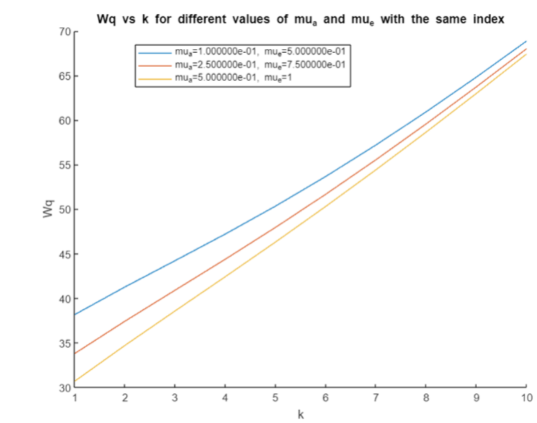
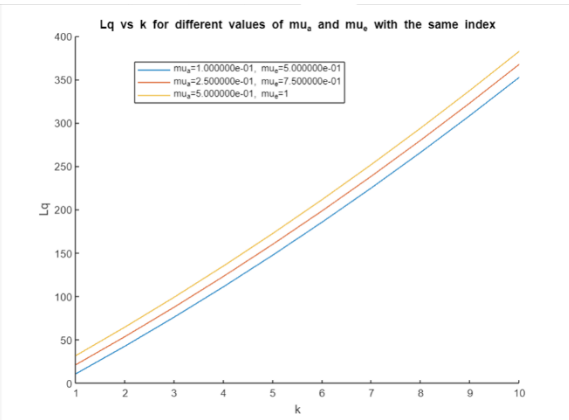


Figure 2: Wq vs. k for DDC *Figure 2.1: Wq vs. k for priority queuing*.

Here, the average time spent by a transaction in the queue for different µ-rates is plotted. We can see that the Wq value for the same ratios, shows that W is stabilizing and maintaining mean trend for model with priority queuing compared to the traditional DDC model.

Figure 3 plots the graph for variable Lq which is the average number of customers (i.e., equivalently the average number of transactions) in the queue (i.e., the block currently being mined) in the decentralized distributed chain model involving K distributed nodes.



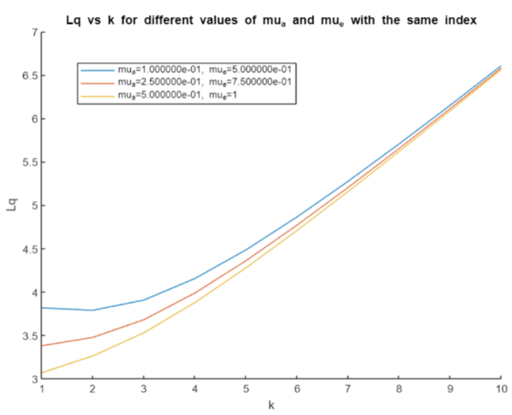
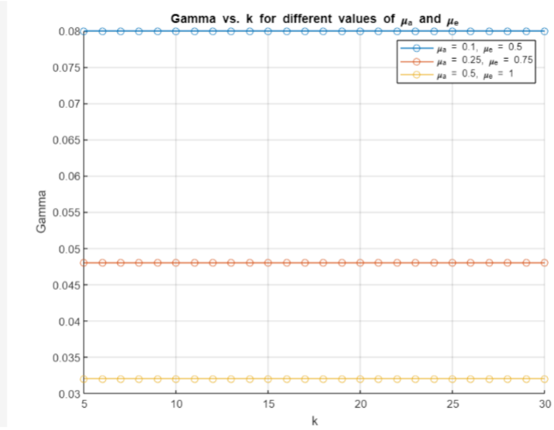


Figure 3: Lq vs. k for DDC *Figure 3.1: Lq vs. k for priority queuing.*

Here, the average number of transactions in the queue for different µ-rates is plotted. We can see that the Lq value for the same ratios, shows a average length of transactions remaining constant for models with priority queuing compared to the traditional DDC model.

Figure 4 plots the graph for variable γ which is the proportion of time that the system is busy serving transactions (i.e., throughput) in the decentralized distributed chain model involving K distributed nodes.



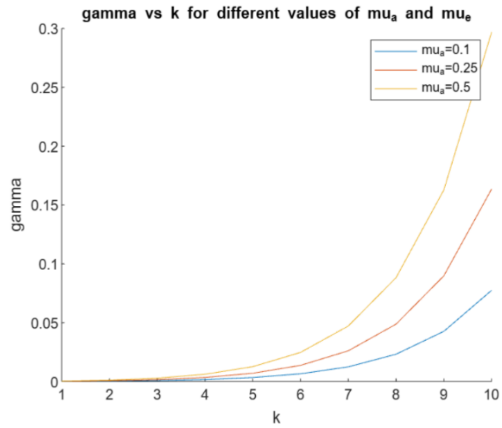


Figure 4: γ vs. k for DDC

Figure 4.1: γ vs. k for priority queuing.

Here, the proportion of time that the system is busy serving transactions (i.e., throughput) for different μ-rates is plotted. We can see that the γ value for the same ratios shows an increasing trend in the throughput for models with priority queuing while throughput being constant in the regular model.

CONCLUSION:

This report introduces a novel and innovative embedded Markovian queueing model of the $M^{1,n}/M^n/1$ type for designing a blockchain-based system. The proposed model is based on the assumption of variable bulk arrivals of transactions that follow a poisson distribution, and static bulk service of transactions in exponential time. The primary performance measures considered in this model include the average number of slots, the average waiting time per slot, and the throughput in terms of the average number of slots processed per unit of time. To validate the model's effectiveness, numerical simulations are conducted using the powerful and widely used MATLAB software.

The proposed model incorporates the priority-based methodology to schedule the transaction process in the blockchain system. The simulations conducted on MATLAB demonstrated that the proposed model improved the system's performance in terms of the average waiting time per slot and the throughput. The results indicated that the priority-based technique reduced the waiting time for the transactions and increased the system's efficiency. Additionally, the simulation results showed that the proposed model was effective in managing the transaction flow in the blockchain system, making it suitable for implementation in real-world applications.

In conclusion, the proposed embedded Markovian queueing model with the priority-based technique is an effective approach for designing a blockchain-based system with improved performance. The model is useful in managing the transaction flow in the system, reducing the waiting time for transactions, and increasing the system's efficiency.