Performance Modeling of Hyperledger Fabric 2.0

Ou Wu, Shanshan Li, Liwen Liu, He Zhang, Xin Zhou and Qinghua Lu

May 31, 2022

Performance Modeling of Hyperledger Fabric 2.0

Ou Wu[1,2], Shanshan Li[1,*], Liwen Liu[1], He Zhang[1,*], Xin Zhou[1], Qinghua Lu[3]
[1]State Key Laboratory of Novel Software Technology, Software Institute, Nanjing University, Nanjng, China
[2]Department of Basic Courses, The PLA Army Engineering University, Nanjing, China
[3] Data61, CSIRO, Sydney, Australia
Email:ouwu123@126.com,lss@nju.edu.cn,llwSDUNJU@163.com
hezhang@nju.edu.cn,job@wetist.com,qinghua.lu@data61.csiro.au

ABSTRACT

Hyperledger Fabric has become one of the most widely used consortium blockchain frameworks with the ability to execute custom smart contracts. Performance modeling and network evaluation are necessary for performance estimation and optimization of the Fabric blockchain platform. The compatibility and effectiveness of existing performance modeling methods must be improved. For this reason, we proposed a compatible performance modeling method using queuing theory for Fabric considering the limited transaction pool. Taking the 2.0 version of Fabric as a case, we have established the model for the transaction process in the Fabric network. By analyzing the two-dimensional continuous-time Markov process of this model, we solved the system stationary equation and obtained the analytical expressions of performance indicators such as the system throughput, the system steady-state queue length, and the system's average response time. We collected the required parameter values through the official test suite. An extensive analysis and simulation was performed to verify the accuracy and the effectiveness of the model and formula. We believe that this method can be extended to a wide range of scenarios in other blockchain systems.

CCS CONCEPTS

• **Networks → Network performance modeling**.

KEYWORDS

Blockchain; Performance modeling; Hyperledger Fabric; Queuing theory; Simulation

1 INTRODUCTION

Hyperledger Fabric (Fabric in short) is a representative consortium blockchain platform that uses the smart contract paradigm and provides fully operable functionality. The distributed nature and limitations of data processing make it very difficult to evaluate and optimize the performance of various blockchains in academia now [3], not except for Fabric [11, 14, 16, 17]. Given that the rapid development and practical application, Fabric requires in-depth planning and deployment of network configuration and developers need to know in advance whether the throughput and latency of the blockchain network can meet their requirements [7]. Moreover, different versions and configurations of Fabric may impact the evaluation of the performance in terms of throughput, transaction rejection probability, and average transaction response delay.

Many studies have used experimental and formal methods to evaluate Fabric performance with various versions [5, 15]. However, to the best of our knowledge, no formal method that considers comprehensive parameters has been proposed for Fabric 2.0 [2]. Different network parameters of the Fabric 2.0 framework should be considered, such as the number of peer nodes, sequencers, organizations, block size, and the limited transaction pool, to support the estimation of the system's throughput and latency.

To bridge this gap, we propose a reproducible modeling approach to evaluate the performance of Fabric 2.0 based on the raft consensus algorithm using queuing theory. Compared to existing studies, we modeled the transaction processes of the Fabric 2.0 network using a queuing theory model with limited transaction pools. The analytical solution of the system model is obtained using the matrix geometric solution, and we calculated several key performance indicators (i.e., the average number of transactions in the queue, the rejection probability, the average transaction execution time, and the average transaction response time). We simulate the impact of changes in system performance indicators after adjusting the system parameters through MATLAB R2016a, including system capacity, transaction arrival rate, block size, etc. Through the evaluation, we verified the effectiveness of the proposed model.

2 BACKGROUND AND RELATED WORK

2.1 Hyperledger Fabric

Hyperledger Fabric[1] is an open-source, permission-based blockchain platform that provides modular components, that is, membership services, chaincodes, and subscription services. Figure 1 shows the transaction flow of Fabric 2.0. The client application sends the transaction proposal to the peer nodes and calls the smart contract to generate a ledger update proposal. Then the result will be endorsed and the endorsed transaction proposal is fed back to the application. The endorsed transactions will be sent to the order nodes and then queued and packaged to generate blocks. Finally these blocks are distributed to all peer nodes for final verification and submission.

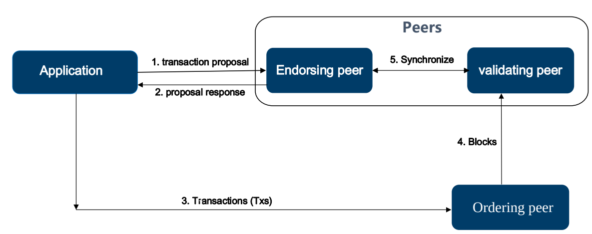


Figure 1: Fabric 2.0 transaction flow.

Fabric 2.0 supports the Raft-based consensus algorithm, whose process is shown in Figure 2. The ordering consensus process of

[1]https://github.com/hyperledger/fabric

transactions is implemented by a group of nodes by electing a leader responsible for managing the replication log. The application client submits the endorsed transaction proposal to the order node and the order node will receive transactions from different application clients at the same time. And each order node automatically routes the received transactions to the current leader of the channel. These transactions are packaged into blocks in a defined order, saved in the ledger of the order node, and distributed to all nodes that have joined the channel. Thereafter, each node will independently verify the received block of transactions in a deterministic manner to ensure that the ledger remains consistent. Transactions that are verified as invalid remain in the blocks created by the order node, but the node marks them as invalid and does not update the state of the ledger. When all nodes are verified, the block ledger is updated.

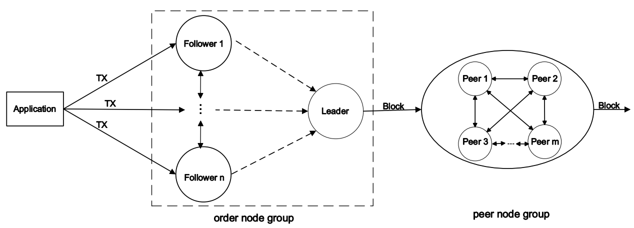


Figure 2: Consensus process of Fabric.

2.2 Performance evaluation of Fabric

Many studies have used experimental evaluation methods to evaluate the performance of various versions of Fabric, such as Fabric v0.6 [12], v1.0 [16], v1.1 [1], v1.2.1 [13] v1.4 [8], etc. Dreyer et al. studied the impact of indicators on Fabric 2.0 performance using test methods and found that Fabric 2.0 is superior to the previous version in almost all aspects of performance [2].

Modeling is another effective performance evaluation method in addition to experiments in the blockchain domain [3]. Queuing theory is commonly used to model performance of different blockchains, for example Bitcoin [6, 9] and Fabric [4, 5]. Geyer et al. [4] introduced the queuing theory model to the Fabric platform and modeled the Solo sorting process of the Fabric platform as an $M/M^B/1$ consensus system. This model captures the characteristics of the sorting phase in the solo implementation. However, it is not suitable for the Raft or Kafka implementation of the later versions of Fabric. Moreover, it does not describe the overall transaction delay in the Fabric system. Jiang et al. [5] developed a hierarchical model for the Fabric v1.4.3 platform and applied queuing theory to derive the impact of transaction arrival rate and endorsement timeout rate on the performance parameters of Fabric transaction process.

However, the current models proposed for Fabric are limited in terms of extensibility, when considering the transaction processing process refined by the generalized Erlang distribution of block generation and block consensus, as well as the limitation of the transaction pool. Moreover, no relevant research has provided an analytical solution for the performance modeling of various versions of Fabric, including 2.0. Our study solves the challenges through the establishment of an analytical modeling using the queuing theory and supports the better performance analysis of Fabric 2.0.

3 PERFORMANCE MODELING

Queuing theory, also known as stochastic service system theory, is a mathematical method for solving the performance and service quality of different types of consensus systems. Arbitrarily complex queues of computer communication networks and blockchain network systems often require the help of queuing theory to be solved. The main research work of this paper is to establish a related queuing theory model to provide system performance evaluation by analyzing the transaction process of Fabric 2.0.

3.1 Model and parameter

Figure 3 shows the transaction consensus process of Fabric 2.0: all order nodes that receive transactions will route the transactions to the leader node's transaction pool to queue. After that, the transactions are packaged to generate blocks according to the set block generation time and block size. Blocks are distributed to peer groups to execute smart contracts and verify transactions. When all peers are verified, the blocks are sent to the blockchain network.

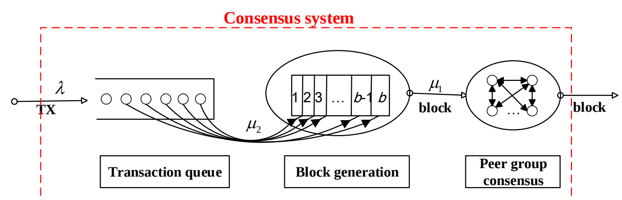


Figure 3: Consensus system of queuing theory.

Table 1 shows all the parameters involved and their meaning.

Table 1: Definition of parameters

Parameter	Description
N	The capacity of consensus system
b	The number of transactions contained in a block
λ	The average arrival rate of transactions in consensus system
μ_1	Average consensus service rate of transaction
μ_2	Average block generation service rate
$I(t)$	The number of transactions in the queue at time t
$J(t)$	The number of transactions in the block at time t
L_q	The average number of transactions in the queue
T_{exe}	Average transaction execution time
P_{rjc}	Transaction rejection probability
T_{resp}	Transaction response delay
TPS	Transaction throughput

3.2 Consensus system

We built a consensus system for the Fabric 2.0 process.

3.2.1 *Introduction of the consensus system.*

Arrival process: The client sends the endorsed transaction randomly to the order node for queuing, because the randomly sent transaction flow has no aftereffect (that is, the number of transaction arrivals in a non-overlapping time interval is independent of each other) and stability. Therefore, it is assumed that the transaction arrival is a Poisson flow, that is, the interval between the arrival of two adjacent transactions obeys the exponential distribution with the parameter λ.

Service process: We divide the transaction service into two separate phases. Transactions arrive at the order node group and queued for the leader to package them into blocks. This is the first stage of the service, block generation, where block generation time obeys the exponential distribution with a parameter of μ_2. After that, the leader sends the packaged block to the peer group to verify the transactions in the block is regarded as the second stage of the service, where transaction validate time obeys an exponential distribution with a parameter of μ_1. Therefore, the transaction service time obeys the generalized Erlang distribution and the average service time is $\frac{1}{\mu_1} + \frac{1}{\mu_2}$.

Block generation rules: Transaction arrivals follow the first-come-first-served (FCFS) principle.

The maximum system capacity: There are at most N transactions in the consensus system. Therefore, when no blocks are generated in this system, the transaction pool can receive at most N transactions. When the system performs consensus verification on a block, there are at most $(N-b)$ transactions in queue.

Independence: We assume that all the random variables defined above are independent of each other.

3.2.2 *A Markov process of consensus system.*

We regard the order node group and the peer node group as a service station, establish a continuous-time Markov process of the consensus system, and derive the stable probability vector of the system through the matrix analysis method.

Let $I(t)$ and $J(t)$ be the numbers of transactions in the queue and in the block at time t, respectively. Then $(I(t), J(t))$ can be regarded as a state of the consensus system at time t. Note that $i = 0, 1, \cdots, N$ and $j = 0, 1, 2, \cdots, b$. Among them, $(N-b+1, 0)$, $(N-b+2, 0), \cdots, (N, 0)$ means that there is no block generation in the system, and the maximum number of transactions in the queue can reach N. $(N-b, 0), (N-b, 1), \cdots, (N-b, b)$ means that the system has a block being processed and the maximum number of transactions in the queue can reach $(N-b)$. For various cases of $(I(t), J(t))$, we write:

$$\begin{aligned}\Omega &= \{(i,j) : i = 0, 1, \cdots, N; j = 0, 1, 2, \cdots, b\} \\ &= \{(0,0), (0,1), (0,2), \cdots, (0,b); (1,0), (1,1), (1,2), \cdots, \\ &\quad (1,b); \cdots; (b,0), (b,1), (b,2), \cdots, (b,b); (b+1,0), \\ &\quad (b+1,1), (b+1,2), \cdots, (b+1,b); \cdots; (N-b,0), \\ &\quad (N-b,1), (b+1,1), (b+1,2), \cdots, (b+1,b); \cdots; \\ &\quad (N-b,0), (N-b,1), (N-b,2), \cdots, (N-b,b); \\ &\quad (N-b+1,0), (N-b+2,0), \cdots, (N-1,0), (N,0)\}\end{aligned}$$

Then $(I(t), J(t))$ is a continuous-time Markov process in the state space Ω. Figure. 4 denotes the state transition relation of the Markov process $\{(I(t), J(t)) : t \geq 0\}$.

According to the state transition diagram, the stationary state equation of this system is expressed as follows:

•*State* $\{(0,0)\}$:

$$-\lambda p(0,0) + \mu_1 [p(0,1) + p(0,2) + \cdots + p(0,b)] = 0 \tag{1}$$

•*State* $\{(0,j), j = 1, 2, \cdots, b\}$:

$$-(\lambda + \mu_1) p(0,j) + \mu_2 p(i,0) = 0 \tag{2}$$

•*State* $\{(i,0), i = 1, 2, \cdots, N-b\}$:

$$\begin{aligned}-(\lambda + \mu_2) p(i,0) + \lambda p(i-1,0) + \mu_1 [p(i,1) + p(i,2) + \\ ... + p(i,b)] = 0\end{aligned} \tag{3}$$

•*State* $\{(i,b), i = 1, 2, \cdots, N-b-1\}$:

$$-(\lambda + \mu_1) p(i,b) + \lambda p(i-1,b) + \mu_2 p(b+i,0) = 0 \tag{4}$$

•*State* $\{(N-b,b)\}$:

$$-\mu_1 p(N-b,b) + \lambda p(N-b-1,b) + \mu_2 p(N,0) = 0 \tag{5}$$

•*State* $\{(N-b,j), j = 1, 2, \cdots, b-1\}$:

$$-\mu_1 p(N-b,j) + \lambda p(N-b-1,j) = 0 \tag{6}$$

•*State* $\{(i,j), i = 1, 2, \cdots, N-b-1; j = 1, 2, \cdots, b-1\}$:

$$-(\mu_1 + \lambda) p(i,j) + \lambda p(i-1,j) = 0 \tag{7}$$

•*State* $\{(i,0), i = N-b+1, N-b+2, \cdots, N-1\}$:

$$-(\mu_2 + \lambda) p(i,0) + \lambda p(i-1,0) = 0 \tag{8}$$

•*State* $\{(N,0)\}$:

$$-\mu_2 p(N,0) + \lambda p(N-1,0) = 0 \tag{9}$$

The state $\{(i,0), i = N-b+1, N-b+2, \cdots, N-1\}$ means that the system has not generated any blocks and the transactions continue to accumulate; here, we analyze this situation separately.

It can be obtained from Equations (8) and (9):

$$\begin{cases} p(N-b+j,0) = (\frac{\lambda}{\lambda+\mu_2})^j p(N-b,0), j = 1, 2, \cdots, b-1 \\ p(N,0) = \frac{\lambda^b}{(\lambda+\mu_2)^{b-1}\mu_2} p(N-b,0) \end{cases} \tag{10}$$

Putting equation (10) into equation (4) and (5), we can get,

$$\begin{cases} -(\mu_1 + \lambda) p(i,b) + \lambda p(i-1,b) + \mu_2 p(b+i,0) = 0, \\ i = 1, 2, \cdots, N-2b \\ -(\mu_1 + \lambda) p(i,b) + \lambda p(i-1,b) + \mu_2 (\frac{\lambda}{\lambda+\mu^2})^{i-(N-2b)} p(N-b,0) = 0, \\ i = N-2b+1, N-2b+2, \cdots, N-b-1 \end{cases} \tag{11}$$

$$-\mu_1 p(N-b,b) + \lambda p(N-b-1,b) + \frac{\lambda^b}{(\lambda+\mu_2)^{b-1}} p(N-b,0) = 0 \tag{12}$$

Simultaneous equations (1)-(3), (6)-(7) and (11),(12) get the $(N-b+1) \times (N-b+1)$ order minimum generator of the system:

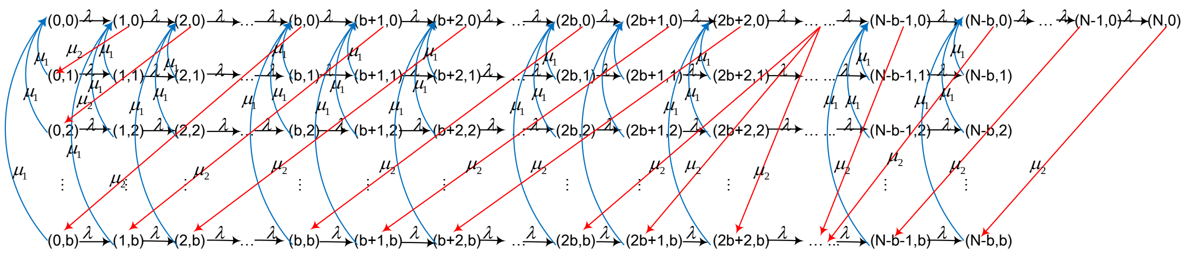


Figure 4: State transition diagram.

$$Q=\begin{bmatrix} B_0 & A_0 & & & & & & & & \\ B_1 & A_1 & A_0 & & & & & & & \\ B_2 & & A_1 & A_0 & & & & & & \\ \vdots & & & A_1 & A_0 & & & & & \\ \vdots & & & & \ddots & \ddots & & & & \\ B_b & & & & & A_1 & A_0 & & & \\ & B_b & & & & & A_1 & A_0 & & \\ & & B_b & & & & & \ddots & \ddots & \\ & & & \ddots & & & & & A_1 & A_0 \\ & & & & B_b & C_1 & C_2 & \cdots & C_{b-1} & A_M \end{bmatrix}$$

Where $A_0, A_1, B_0, B_i(i=1,2,\cdots,b), C_j(j=1,2\cdots,b-1)$, and A_M are $(b+1)\times(b+1)$ order matrices, and

$$A_0=\begin{bmatrix} \lambda & & & \\ & \lambda & & \\ & & \ddots & \\ & & & \lambda \end{bmatrix}$$

$$A_1=\begin{bmatrix} -(\lambda+\mu_2) & & & \\ \mu_1 & -(\lambda+\mu_1) & & \\ \vdots & & \ddots & \\ \mu_1 & & & -(\lambda+\mu_1) \end{bmatrix}$$

$$B_0=\begin{bmatrix} -\lambda & & & \\ \mu_1 & -(\lambda+\mu_1) & & \\ \vdots & & \ddots & \\ \mu_1 & & & -(\lambda+\mu_1) \end{bmatrix}$$

$$B_1=\begin{bmatrix} 0 & \mu_2 & 0 & \cdots & 0 \\ & & & & \\ & & & & \end{bmatrix}$$

$$B_2=\begin{bmatrix} 0 & 0 & \mu_2 & \cdots & 0 \\ & & & & \\ & & & & \end{bmatrix}$$

$$\vdots$$

$$B_b=\begin{bmatrix} 0 & 0 & 0 & \cdots & \mu_2 \\ & & & & \\ & & & & \end{bmatrix}$$

$$C_1=\begin{bmatrix} 0 & \cdots & 0 & 0 & \mu_2(\frac{\lambda}{\lambda+\mu_2}) \\ & & & & \\ & & & & \end{bmatrix}$$

$$C_2=\begin{bmatrix} 0 & \cdots & 0 & 0 & \mu_2(\frac{\lambda}{\lambda+\mu_2})^2 \\ & & & & \\ & & & & \end{bmatrix}$$

$$\vdots$$

$$C_{b-1}=\begin{bmatrix} 0 & \cdots & 0 & 0 & \mu_2(\frac{\lambda}{\lambda+\mu_2})^{b-1} \\ & & & & \\ & & & & \end{bmatrix}$$

$$A_M=\begin{bmatrix} -(\lambda+\mu_2) & 0 & \cdots & 0 & \frac{\lambda^b}{(\lambda+\mu_2)^{b-1}} \\ \mu_1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ \mu_1 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Let $\pi=(\pi_0,\pi_1,\pi_2,\cdots,\pi_{N-b})$ be the stationary probability vector of this system, and each sub-vector $\pi_i=(\pi_{i0},\pi_{i1},\cdots,\pi_{ib}), i=0,1,2,\cdots,N-b$ is a $b+1$ dimensional row vector, then the system of stationary equations is

$$\begin{cases} \pi Q=0 \\ \pi e=1 \end{cases} \tag{13}$$

Where e is a column vector of ones with proper dimension.

Then, we have

$$\pi_0B_0+\pi_1B_1+\pi_2B_2+\cdots+\pi_bB_b=0 \tag{14}$$

$$\pi_0A_0+\pi_1A_1+\pi_{b+1}B_b=0 \tag{15}$$

$$\pi_{i-1}A_0+\pi_iA_1+\pi_{i+b}B_b=0, i=2,3,\cdots,N-2b \tag{16}$$

$$\begin{aligned} &\pi_{i-1}A_0+\pi_iA_1+\pi_{N-b}C_{i-(N-2b)}=0, \\ &i=N-2b+1,N-2b+2,\cdots,N-b-1 \end{aligned} \tag{17}$$

$$\pi_{N-b-1}A_0+\pi_{N-b}A_M=0 \tag{18}$$

$$\pi e=1 \tag{19}$$

Since the matrix A_0 is a diagonal matrix, the matrix analysis method proposed in [10] is used to solve the steady-state probability vector. The diagonal matrix A_0 is expressed as $A_0=\lambda I$ (I is the $(b+1)$ order identity matrix). Let $R_{N-b}=I$, then

$$\pi_{N-b}=\pi_{N-b}R_{N-b} \tag{20}$$

According to equation (18), we have

$$\pi_{N-b-1}=\pi_{N-b}(-\frac{1}{\lambda}A_M)=\pi_{N-b}R_{N-b-1} \tag{21}$$

Here $R_{N-b-1}=-\frac{1}{\lambda}A_M$ is called the subrate matrix.

Substituting equation (21) into equation (17), we get

$$\begin{aligned}\pi_{N-b-(i+1)} &= \pi_{N-b}[-\frac{1}{\lambda}(R_{N-b-i}A_1 + C_{b-i})] \\ &= \pi_{N-b}R_{N-b-(i+1)}, \\ & i = 1,2,\cdots,b-1\end{aligned} \tag{22}$$

Here

$$R_{N-b-(i+1)} = -\frac{1}{\lambda}(R_{N-b-i}A_1 + C_{b-i}), i = 1,2,\cdots,b-1.$$

Substituting Equation (22) into Equation (16), we get

$$\begin{aligned}\pi_{N-b-(i+1)} &= \pi_{N-b}[-\frac{1}{\lambda}(R_{N-b-i}A_1 + R_{N-i}B_b)] \\ &= \pi_{N-b}R_{N-b-(i+1)}, \\ & i = b, b+1,\cdots,N-b-1\end{aligned} \tag{23}$$

Here

$$\begin{aligned}R_{N-b-(i+1)} &= -\frac{1}{\lambda}(R_{N-b-i}A_1 + R_{N-i}B_b), \\ & i = b, b+1,\cdots,N-b-1.\end{aligned}$$

According to equation (14), we have

$$\begin{aligned}\pi_0 &= -\pi_{N-b}(R_1B_1 + R_2B_2 + \cdots + R_bB_b)B_0^{-1} \\ &= \pi_{N-b}R_0,\end{aligned} \tag{24}$$

Here $R_0 = -(R_1B_1+R_2B_2+\cdots+R_bB_b)B_0^{-1}$, the process of solving $R_i, i = 0,1,2,\cdots,N-b$ is shown in Figure 5.

Simultaneous equations (15) and (19), we have

$$\begin{cases}\pi_{N-b}(R_0B_0 + R_1B_1 + R_{b+1}B_b) = 0 \\ \pi_{N-b}(R_0 + R_1 + R_2 + \cdots + R_{N-b-1} + I)e = 1\end{cases} \tag{25}$$

Solving for π_{N-b}, according to equations (20)-(24), the steady-state probability vector π can be solved.

3.2.3 Performance analysis.

In this section, we provide the performance measurement indicator of the consensus system, and its expression is given by π and R. Analyze the influence of parameters on system performance indicators through numerical calculations.

When the consensus system is stable, we write

$$\lim_{t\to+\infty} I(t) = L_q, \lim_{t\to+\infty} J(t) = J_b, \tag{26}$$

(a) Average queue length of consensus system

$$E(L_q) = \sum_{i=0}^{N-b}(i\sum_{j=0}^{b}\pi_{ij}) = \pi_{N-b}[R_1+2R_2+\cdots+(N-b)R_{N-b}]e \tag{27}$$

(b) Transaction rejection probability of consensus system

$$P_{rjc} = \sum_{j=0}^{b}\pi_{N-b,j} = \pi_{N-b}e \tag{28}$$

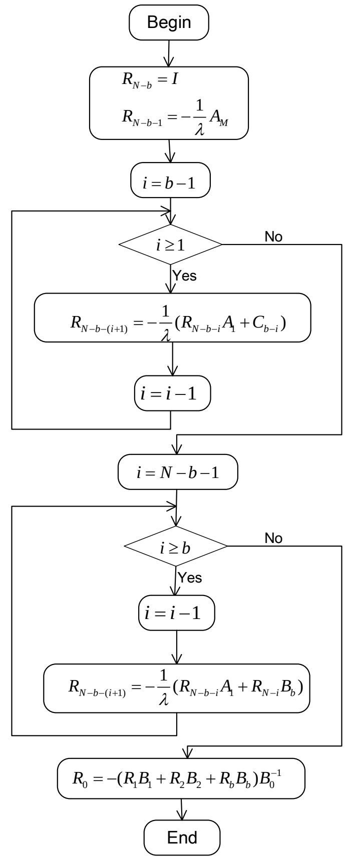


Figure 5: R_i calculation flow chart.

(c) Average transaction execution time of consensus system

$$\begin{aligned}E(T_{exe}) = &\sum_{k=0}^{[\frac{N-b-l}{b}]}\sum_{l=0}^{b-1}\pi_{kb+l,0}(k+1)(\frac{1}{\mu_1}+\frac{1}{\mu_2}) \\ &+ \sum_{k=0}^{[\frac{N-b-l}{b}]}\sum_{l=0}^{b-1}\sum_{j=1}^{b}\pi_{kb+l,j}[\frac{1}{\mu_1}+(k+1)(\frac{1}{\mu_1}+\frac{1}{\mu_2})]\end{aligned} \tag{29}$$

The proof process is analogous to [9], where $[\frac{N-b-l}{b}]$ is the rounding function.

(d) Average transaction response time of consensus system

$$E(T_{resp}) = \frac{E(L_q)}{\lambda(1-p_{rjc})} \tag{30}$$

(e) Throughput of consensus system

$$TPS = \lambda(1-p_{rjc}) \tag{31}$$

4 MODEL SIMULATION AND VALIDATION

In this section, we will simulate the performance indicators of the Fabric 2.0 system by varying several important parameters such as transaction arrival rate, consensus system capacity, transaction consensus rate, and block generation rate.

4.1 Simulation experiment setup

To analyze the influence of system parameters on the above system performance indicators, we installed the software platform $MATLABR2016a$ to verify the approximate accuracy of our model by setting some parameter values and mathematical simulation of the impact of the range on the performance indicators.

4.2 Performance evaluation

4.2.1 Influence of transaction arrival rate (λ).

We set the range of variation of λ to 100 to 2000 (transactions/second), $\mu_1 = \mu_2 = 100$ (transactions/second), the capacity of the consensus system capacity $N = 500$ (transactions). Figure $6(a) - (e)$ show the changes in performance indicators such as the probability of transaction rejection and the average delay in transaction response under different block sizes b.

From Figure $6(a) - (e)$, we can conclude that, except for queue length, changes in other performance metrics cannot avoid the transaction arrival rate, especially when throughput and transaction arrival rate are linearly related. When the transaction arrival rate is determined, the larger the block size b is, although the queue length, transaction execution time, and response time will decrease, the rejection probability will increase rapidly. Therefore, blindly following a large area will not achieve good performance.

4.2.2 Influence of consensus system capacity (N).

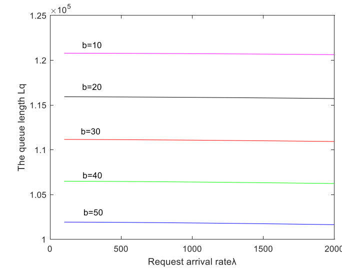
We set the range of variations of N to be 100 to1000 , $\mu_1 = \mu_2 = 100$, the transaction arrival rate $\lambda = 1000$. Figure $7(a) - (e)$ show the changes in performance indicators under different block sizes b.

From Figure $7(a) - (e)$, it can be concluded that when the consensus system capacity is $N \geq 200$, the change of N have Little effect to the probability of rejection, transaction execution time and throughput rate, but queue length and transaction response time will be increase as N increases. Therefore, the impact of N on the performance of the existing system cannot be ignored. When N is large and fixed, the size of the block does not affect the values of rejection probability and throughput, but is inversely proportional to both the length of the queue and the execution time of the transaction.

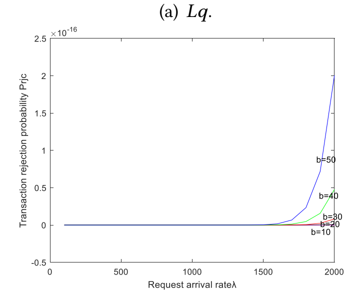
4.2.3 Influence of transaction consensus rate (μ_1) or transaction generate rate (μ_2).

The parameter μ_1, μ_2 represents the consensus rate for the block and the block generation rate. Its numerical value is related to the number of order nodes and the number of peer nodes. Here, we assume that the range of both μ_1 and μ_2 is $0 - 1000$, when $\lambda = 1000, N = 500$, Figure $8(a) - (e)$ show the variation law of the performance indicators in μ_1 under different block sizes b.

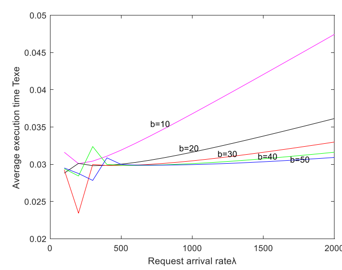
From Figure $8(a) - (e)$, it can be seen that the change of μ_1 has little effect on queue length, transaction response time and throughput. However, as μ_1 increases, the rejection probability first decreases and then increases to 0, and the execution time accordingly decreases. This is because the higher the block efficiency that generates transactions (block processing efficiency), the higher the probability of receiving the transaction, the smaller the rejection probability and the shorter the execution time of the transaction. When μ_1 is fixed, the change in block size b affects throughput



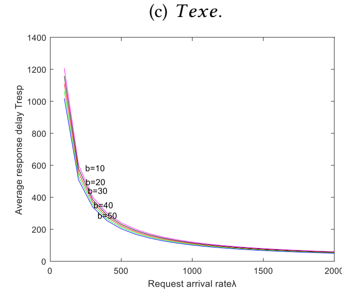
(a) Lq.



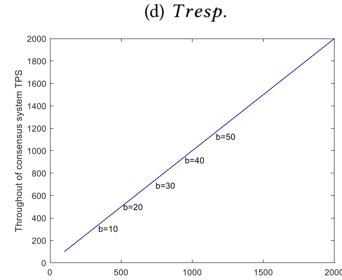
(b) $Prjc$.



(c) $Texe$.

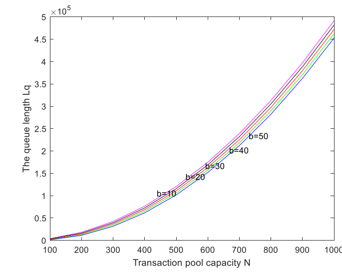


(d) $Tresp$.

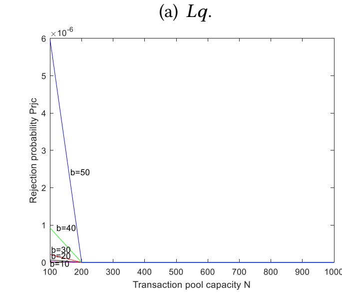


(e) TPS.

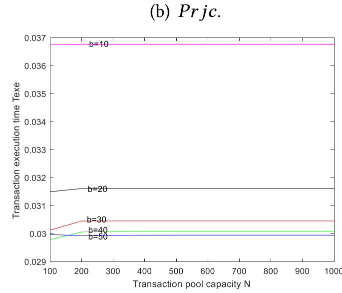
Figure 6: Performance indicators with λ



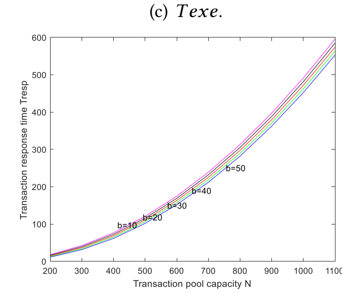
(a) Lq.



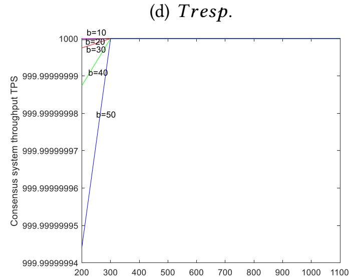
(b) $Prjc$.



(c) $Texe$.

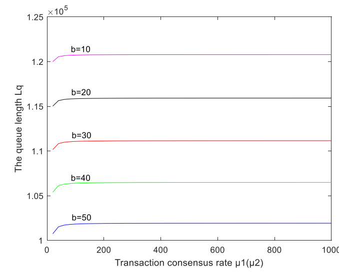


(d) $Tresp$.

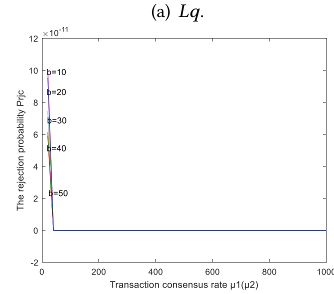


(e) TPS.

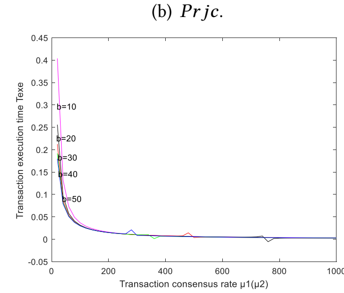
Figure 7: Performance indicators with N



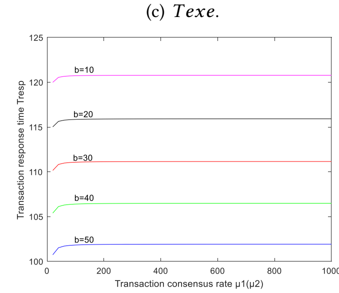
(a) Lq.



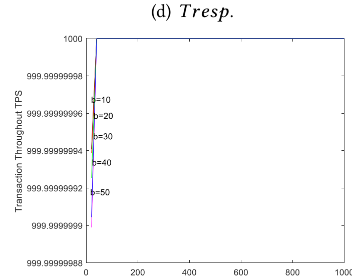
(b) $Prjc$.



(c) $Texe$.



(d) $Tresp$.



(e) TPS.

Figure 8: performance indicators with μ_1

and execution time very little, but as the block size increases, the queue length and response time decrease, so large blocks are the best choice in the current situation.

Since the impact of μ_2 on each performance indicator is similar to μ_1, it will not be described in detail here.

We have shown the impact of different parameters on system performance indicators in four dimensions. Summarized as follows: The transaction arrival rate λ is the only factor that affects the system's throughput. Increasing the size of the block b can reduce the length of the queue, the execution time of the transaction, and the response time. Meanwhile, the rejection probability of the system may also increase. Finding a suitable block size is important to optimize Fabric performance. Unlike relevant studies, we find that the impact of the size of the transaction pool on system performance cannot be ignored, and the transaction pool N cannot easily be made infinite. Different values of N will affect the accumulation of queues. The size of the block generation rate and the consensus rate directly affect the efficiency of the entire consensus system, so selecting an appropriate number of peer and order nodes can ensure transaction processing efficiency and system security.

4.3 Model experimental verification

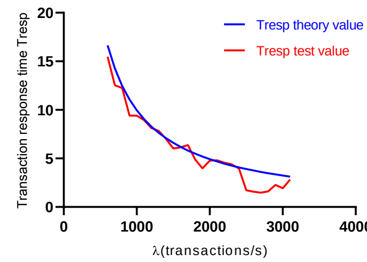
To verify the validity of the model, we installed a Fabric 2.0 network on the 48C 187G server with the Raft ordering service and set up an order node and a validator group consisting of three peer nodes. We used Hyperledger Caliper[2] to test different blockchain solutions with custom use cases. We set the system capacity $N = 150$, the block size $b = 10$, the transaction arrival rate λ to range from 500 to 3000. The average transaction delay and throughput of the consensus system are tested, and the theoretical and test values are compared as follows.

Figure $9(a) - (b)$ show the comparison between the theoretical value and the test value of the transaction response time. It can be easily obtained, when the error is 0.5, the cumulative probability reaches 50%, and the maximum transaction error does not exceed 26, indicating that the response time equation of the consensus system is valid. Figure $10(a) - (b)$ show the comparison between the theory value and the test value of the throughput of the consensus system. We found that the test and theory values of throughput were almost identical. And the probability that the error is less than 20 reaches 75%, so it has also been verified that the throughput equation of the consensus system is valid.

5 CONCLUSION

This study proposed a feasible and extensible modeling method for Fabric 2.0, using the queuing theory model with a limited transaction pool. We obtained key performance indicators related to the consensus system models and conducted a series of experiments to validate the simulated models. During the experiments, we simulated the system performance indicators through the parameter change process to verify the effectiveness of the model. Finally, the validity of the proposed model was verified by comparing the experimental data with the theoretical data using benchmark tests. The future research work of this study are twofold: 1) Optimizing the proposed queuing theory model by supplementing other relevant variables for performance analysis, such as peer group is attacked; 2) Expanding the proposed model to other similar blockchain scenarios to improve the generality of our work.

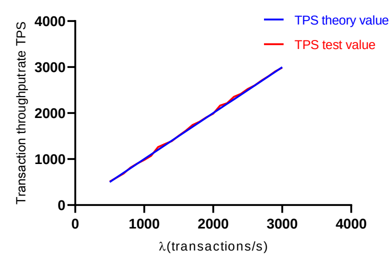
[2]https://github.com/hyperledger/caliper



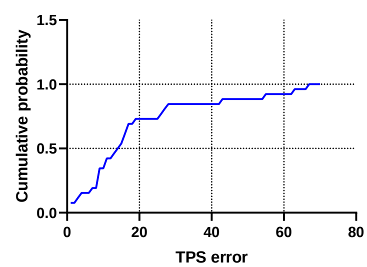
(a) T_{resp} comparison.

(b) T_{resp} error probability cumulative graph.

Figure 9: Response time of the consensus system.



(a) TPF comparison.



(b) TPF error probability cumulative graph.

Figure 10: Throughput of the consensus system.

ACKNOWLEDGMENTS

This work is jointly supported by the National Key Research and Development Program of China (No.2019YFE0105500) and the Research Council of Norway (No.309494), as well as the National Natural Science Foundation of China (Grants No.62072227, 61802173), the Key Research and Development Program of Jiangsu Province (No. BE2021002), and the Intergovernmental Bilateral Innovation Project of Jiangsu Province (BZ2020017).

REFERENCES

[1] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. 2018. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the thirteenth EuroSys conference*. 1–15.

[2] Julian Dreyer, Marten Fischer, and Ralf Tönjes. 2020. Performance analysis of hyperledger fabric 2.0 blockchain platform. In *Proceedings of the Workshop on Cloud Continuum Services for Smart IoT Systems*. 32–38.

[3] Caixiang Fan, Sara Ghaemi, Hamzeh Khazaei, and Petr Musilek. 2020. Performance evaluation of blockchain systems: A systematic survey. *IEEE Access* 8 (2020), 126927–126950.

[4] Fabien Geyer, Holger Kinkelin, Hendrik Leppelsack, Stefan Liebald, Dominik Scholz, Georg Carle, and Dominic Schupke. 2019. Performance perspective on private distributed ledger technologies for industrial networks. In *2019 International Conference on Networked Systems (NetSys)*. IEEE, 1–8.

[5] Lili Jiang, Xiaolin Chang, Yuhang Liu, Jelena Mišić, and Vojislav B Mišić. 2020. Performance analysis of Hyperledger Fabric platform: A hierarchical model approach. *Peer-to-Peer Networking and Applications* (2020), 1–12.

[6] Yoshiaki Kawase and Shoji Kasahara. 2017. Transaction-confirmation time for bitcoin: A queueing analytical approach to blockchain mechanism. In *International Conference on Queueing Theory and Network Applications*. Springer, 75–88.

[7] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman. 2019. Performance Analysis of a Hyperledger Fabric Blockchain Framework: Throughput, Latency and Scalability. In *2019 IEEE International Conference on Blockchain (Blockchain)*.

[8] Murat Kuzlu, Manisa Pipattanasomporn, Levent Gurses, and Saifur Rahman. 2019. Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability. In *2019 IEEE international conference on blockchain (Blockchain)*. IEEE, 536–540.

[9] Quan-Lin Li, Jing-Yu Ma, and Yan-Xia Chang. 2018. Blockchain queue theory. In *International Conference on Computational Social Networks*. Springer, 25–40.

[10] Ehem Molle. [n.d.]. N the Solution to QBD Processes with Finite State Space", Stochastic analysis and Applications. ([n. d.]).

[11] Qassim Nasir, Ilham A. Qasse, Manar Abu Talib, and Ali Bou Nassif. 2018. Performance Analysis of Hyperledger Fabric Platforms. *Security and Communication Networks* 2018 (2018), 1–14.

[12] Qassim Nasir, Ilham A Qasse, Manar Abu Talib, and Ali Bou Nassif. 2018. Performance analysis of hyperledger fabric platforms. *Security and Communication Networks* 2018 (2018).

[13] Thanh Son Lam Nguyen, Guillaume Jourjon, Maria Potop-Butucaru, and Kim Loan Thai. 2019. Impact of network delays on Hyperledger Fabric. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 222–227.

[14] Suporn Pongnumkul, Chaiyaphum Siripanpornchana, and Suttipong Thajchayapong. 2017. Performance analysis of private blockchain platforms in varying workloads. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 1–6.

[15] Harish Sukhwani, Nan Wang, Kishor S Trivedi, and Andy Rindos. 2018. Performance modeling of hyperledger fabric (permissioned blockchain network). In *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*. IEEE, 1–8.

[16] Parth Thakkar, Senthil Nathan, and Balaji Viswanathan. 2018. Performance benchmarking and optimizing hyperledger fabric blockchain platform. In *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE, 264–276.

[17] Canhui Wang and Xiaowen Chu. 2020. Performance characterization and bottleneck analysis of hyperledger fabric. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 1281–1286.