ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal
Regular Issue, Vol. 9 N. 3 (2020), 17-28
eISSN: 2255-2863
DOI: https://doi.org/10.14201/ADCAIJ2020931728

# Modelling and Simulation of Queuing Models through the concept of Petri Nets

Shadab Siddiqui, Manuj Darbari, Diwakar Yagyasen

BBD University. India
cseshadabsiddiqui@gmail.com; manuj_darbari@acm.org; dylucknow@gmail.com

KEYWORD

*Petri Nets; graphical formalism; modeling; queuing.*

ABSTRACT

*In recent years Petri Nets has been in demand due to its visual depiction. Petri Nets are used as an effective method for portraying synchronization, a concurrency between different system activities. In queuing models Petri networks are used to represent distributed modeling of the system and thus evaluate their performance. By specifying suitable stochastic Petri Nets models, the authors concentrate on representing multi-class queuing systems of various queuing disciplines. The key idea is to define SPN models that simulate a given queue discipline 's behavior with some acceptable random choice. Authors have found system queuing with both a single server and multiple servers with load-dependent service rate. Petri networks in the queuing model have enhanced scalability by combining queuing and modeling power expressiveness of 'petri networks.' Examples of application of SPN models to performance evaluation of multiprocessor systems demonstrate the utility and effectiveness of this modeling method. In this paper, authors have made use of Stochastic Petri nets in queuing models to evaluate the performance of the system.*

## 1. Introduction

Petri Nets are used as a graphical representation to explain the movement within the network of various activities. As opposed to block diagrams and logical trees, Petri Nets are the most effective method for graphic representation. The Petri Nets queuing can be used to reflect business applications in a graphic way. Also, timed Petri Nets can be used to express the notion of time to evaluate device efficiency and performance. Deterministic or random variables can be used in Petri Nets. Analysis of the structure and behavior of a given system can be done using Petri Nets by modelling and simulation of discrete events.

Petri Nets are used to describe the logical structure of the system (Peterson., 1977). The petri net graphs are formed by collecting nodes (places) representing system status and represented by arc-connected circles. The transitions that reflect system behavior are represented by bars. In Petri Nets, timed interfaces can also be used in system representation according to its consideration. In the Petri Nets there are flow indicators in the circles representing nodes, called tokens. The number of tokens can differ as per Petri Nets' functions. A number of tokens are used during the active transition as (input, places) thus producing the (output, places).

The transitions can be fired while Petri Nets is being executed. Although more than one transition is randomly ready to fire, this does not define which transition will initially begin to fire. While Petri Nets are widely used in modeling, there is still a lack of application integration knowledge about its use.

The paper structure is arranged according to the following: -

First describes the introduction and related work followed by elements of petri net. The various queuing models followed by the proposed Petri Nets will then be addressed and their effect and interpretation. Finally, it deals with the conclusion and work to come.

# 2. Related Work

(Varela et al., 2015) have proposed a model for operational and strategic decision making in tire industry. This method was developed to reduce the waste of raw material from manufacturing process. (Pauleve et al., 2010) have proposed a technique for manipulating the characteristics of stochastic petri net calculations by taking into consideration exponential rate. Queuing theory and simulation is proposed by (Camelo et al. 2010) for determining the service characteristics of mineral ships so that 'mean' can be calculated.

(Wanini et al., 2020) did the performance analysis of bus line using stochastic Petri Nets. The proposed model proposes more power and is used to test the system using various scenarios. The given model is easy to implement as it does not use any mathematical theories. (Bakshandeh et al., 2019) have made use of stochastic Petri Nets and queuing theory to calculate the performance of BPEL. The authors have proposed a model based on "stochastic" [Petri Nets] using matrix calculation. The model made use of exponential distribution for transition and Poisson distribution for arcs. (Luo et al., 2019) have made use of Petri Nets for awareness of airport operation. The authors have developed an algorithm for predicting the accurate situation. Authors have done experiments to verify the accuracy of the proposed algorithm.

(Djamila Boukredera et al., 2020) suggested that CR networks be configured as a retrial queueing mechanism in which PUs have preventive priority over SU. Authors construct the simulation model to this effect of the Synchronized Stochastic Colored Petri Nets queuing method. Similar practical findings hence will be drawn while the network conditions differ. Both exponential distributions and Erlang distributions are called SU service time modeling. The findings obtained with restrictive effect assumptions match the empirical findings for very similar queuing models observed. What shows the efficacy of the STCPN simulation model proposed? For modeling and study of spectrum occupancy in CR networks, authors used a single server retrial queueing method with preemptive priority. The proposed model is of the view that secondary users can access the unused bands dynamically and opportunistically without interfering with primary users.

(Tilak Agerwala., 1979) has brought together a large body of work on useful Petri Nets applications. Modeling a system using (interpreted) Petri nets has three possible advantages: first, because of the graphic and detailed design of the representation scheme, the overall structure is often easier to grasp. Second, system behavior can be analyzed using Petri net theory, which includes analytical tools such as marking trees and invariants, as well as established relationships between certain net structures and dynamic behavior. We can also apply techniques developed for checking parallel programs. Finally, since bottom-up and top-down approaches can be used to synthesize Petri nets, it is possible to systematically design systems whose behavior is either known or easily verifiable.

(MARCO AJMONE MARSAN et al., 1984) presented GSPNs equivalent to continuous-time stochastic processes, and solution methods for steady-state probability distribution derivation. Examples of the application of GSPN models to multiprocessor performance measurement systems demonstrate the utility and efficacy of this modeling tool. Embedded Markov 's analysis Chain makes the estimation of a steady state distribution of GSPN labeling probabilities. Since residence times in tangible states are null, this steady state distribution of probability that only assign nonzero probabilities to tangible states. Based on this observation, a computationally efficient solution method which considers the tangible state was presented.

(Simonetta Balsamo et al., 2007) researched the multiclass relationships BCMP-like, and generalized service stations Petri stochastic nets (GSPN). The authors based on multiclass queuing schemes with different disciplines in queue by defining appropriate Finite GSPN Templates. Authors structurally describe the Finite GSPNs with single level equivalent M / M / k FCFS queuing system, LCFSPR, Processor Sharing and Limitless Servers (IS) (PS). The main aim is to define a finite GSPN model simulating the operation of a given queue discipline with other Suitable random choice. Moreover, authors said combined comparable versions launched has a closed-form constant state probability of M) property. Authors find program queuing with both servers have load dependent service rates, and several servers whose service rate is constant.

## 3. Elements in Petri Nets

The notations used in Petri Nets (Peterson, 1981) contains ('PL' 'TI' 'IN' 'OT' 'T')
01: "PL"-> It is the set of 'places' represented by circles ('pl1' 'pl2' 'pl3'----'pln')
02: "TI"-> It is the set of 'transitions' represented by bars ('ti1' 'ti2' 'ti3-----'tin')
03: "IN"-> The places from which an arc runs to a transition are called transition input places
04: "OT"-> ; The places where arcs run from a transition are called transition output places.
05: "T"-> The set of tokens represented as dots in the diagram (t1, t2, ,t3, ----, tn)

The graph in Petri Nets contains the nodes containing places and transitions and the arcs defining input and output relations.

01: "PL" = ['pl1'~ 'pl2'~'pl3'~'pl4'~'pl5']
02: "TI" = ['ti1'~ 'ti2'~ 'ti3'~ 'ti4'~ 'ti5']

| | |
|---|---|
| 03: IN(~ti1~)= [~pl1~] | 08: OT(~ti1~)=['pl2'~ 'pl3'] |
| 04: IN (~ti2~)=[~pl2~] | 09: OT(~tI2~)=[~pl4~] |
| 05: IN(~ti3~)=[~pl3~] | 10: OT(~ti3~)= [~pl5~] |
| 06: IN(~ti4~)= [~pl4~] | 11: OT(~ti4~)= [~pl2~] |
| 07: IN (~ti5~)= ['pl4'~ 'pl5'] | 12: OT (~ti5~)= [~pl1~] |

13: T1 = ('1,0,0,0,0')

**Figure 1** shows the petri net with transitions and places. If the input carries token, then the transition is enabled.

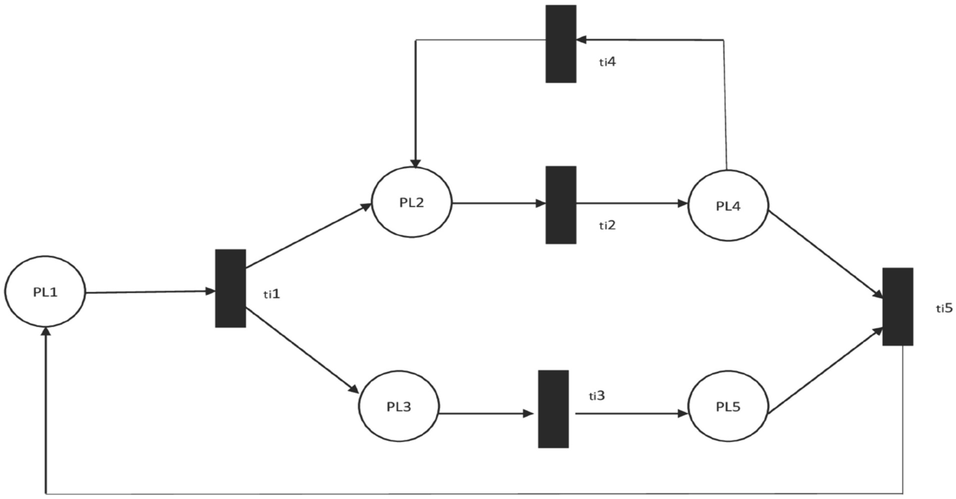


*Figure 1: Petri Net with input and output elements.*

# 4. Queuing Theory(QT)

"Queuing theory(QT)" is the study of waiting lines (WL)" or "Queue length(QL)". There are various parts of the queue as arrival rate is the total number of users arriving in the "queue", "service rate (SR)" determines the services provided to the user and the "queue length (QL)" contains the capacity of the "queue". (Basak et al., 2019). Queuing theory helps in the derivation of 'waiting time', response time in the system. It also helps in identifying whether the queue is empty or full. There are stochastic models of queue which represents the probability of finding queue in a steady state.

## 4.1. "M-M-1"

"M-M-1" is the basic queuing model used to model machines or other communication equipment information (Vijayashree et al., 2018). In "M-M-1" queue model there is only one server and it follows Poisson service distribution. The various characteristics are: -

- "Arrival Rate(**AR**) λ"
- "Service Rate(**SR**) μ"
- "Utilization Rate(**UR**)"
- "Customers" waiting in the queue
- "Waiting time(**WT**)"

## 4.2. "M-M-c"

In multi-server model there can be 'c' servers. "M-M-c" model follows exponential distribution for service rate. The 'c' servers are independent of "arrival rate" and "service rate".

## 4.3. 'Kendell' Notation' in Queuing Model

'Notation' of Kendell is as follows: ~A / S / c / B / N / D~ (Khomonenko et al., 2016)

'A'->A is often called the time of inter-arrival. This is the time between two customers coming in. Poisson distribution is known as the distribution of probabilities for A.

'S'->S is often referred to as 'time of operation.' It's the time it takes to represent the customer after he exits the queue.

'C'-> 'c' is the total number of servers that have one server in the queuing system/M/1 model and M/M/k has several servers in the queuing system

'B'- > Specifies the (customer number) that is being serviced in the queue.

'N'->N is the minimum (customer count) that can join the queue.

'D'->D is the structure of the queuing, including 'FIFO' or priority.

## 4.4. Markov process

Markov cycle, named after the Russian scientist Andrey Markov (Vijayashree et al., 2018), is a time-varying random process for which the Markov property holds a particular property. Markov Chain: If we consider that the search space, I, is discrete, then the Markov cycle is known as the Markov Chain.

### 4.4.1. DTMC (Discrete Time Markov Chain)

The Markov chain is defined as a discrete time Markov chain if the parametric T, is also discrete. Assuming T={0,1,2, ...} in this case. The Markov property may be specified for a DTMC as:-

$$P(Y_n = j_n | Y_0 = j_0, Y_1 = j_1, \ldots, Y_{n-1} = j_{n-1}) = P(Y_n = j_n | Y_{n-1} = j_{n-1}), j_0, j_1, \ldots j_n \in J \quad (1)$$

y-> random variable

j-> j is the state

### 4.4.2. CTMC (Continuous Time Markov Chain)

The Markov chain is defined as a Continuous Time Markov Chain if the parametric T is continuous. Assuming T= [0,∞), the Markov property can be declared for a CTMC as:-

$$P\big(Y_{(t)} = Y | Y(t_n) = Y_n, Y(t_{n-1}) = Y_{n-1}, \ldots, Y(t_0) = Y_0\big) = P(Y(t) = Y | Y(t_n) = Y_n) \quad (2)$$

y-> random variable

# 5. Proposed Petri Net Models

Petri Nets are used in queuing model to provide better service availability to the users, reducing the 'waiting time' in the system and thereby increasing the performance.

**Figure 2** depicts the petri net model on M-M-1. The 'places' and 'transitions' is five. In M-M-1 model number of 'server' is one, therefore we have made single server queue.

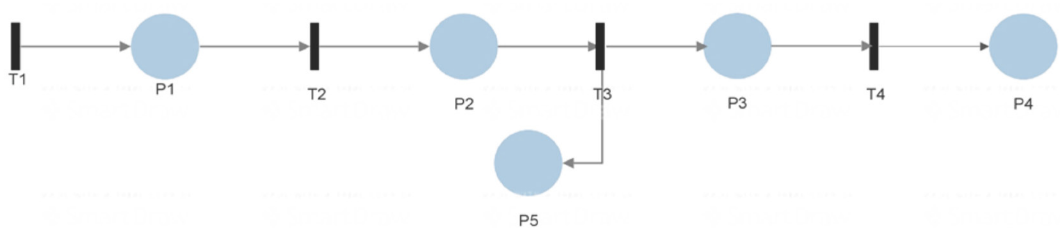


*Figure 2: Petri Net(PN) model on "M-M-1".*

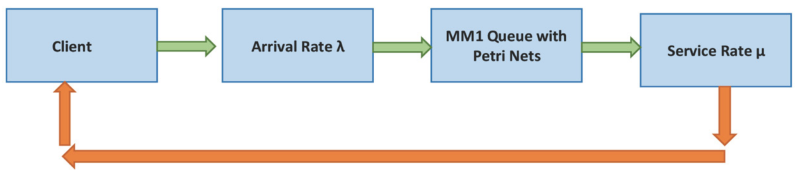


*Figure 3: "M-M-1" Model with Petri Nets.*

**Figure 4** depicts the 'petri-net' model on "M-M-c". In "M-M-c" model there can be 'n' number of servers, therefore multiple layer of server queues are formed.

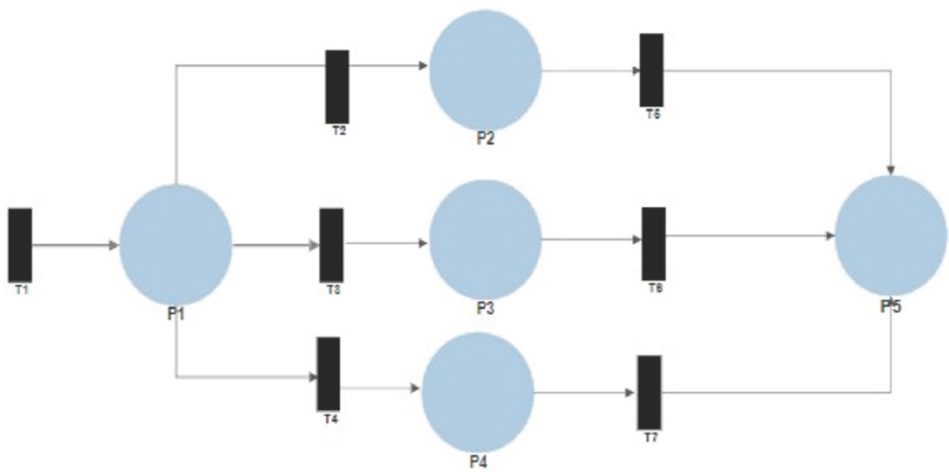


*Figure 4: Petri net model on "M-M-c".*

The M / M/1/c queuing system, which is modeled as SPN, is shown in **Figure.3, 5**. The customers will seek service at a moment in time. The number of customers / jobs are stored in the Clients. The arrival time value, determines the inter-arrival mean time for service seeking clients / jobs. Entry in the network at arrival rate, λ. The arrival rate is reciprocal to the value of time of inter-arrival i.e. λ = 1/ t1. This is the function that exponentially determines the distributed transitional delays in firing. Each change firing takes a single token and put it in the position, that is similar to the real-life situation of a client / job joining a service queue. The server consists of the place itself and the transition. The Server gives an exclusive service, i.e. only one customer can use μ at the same time.

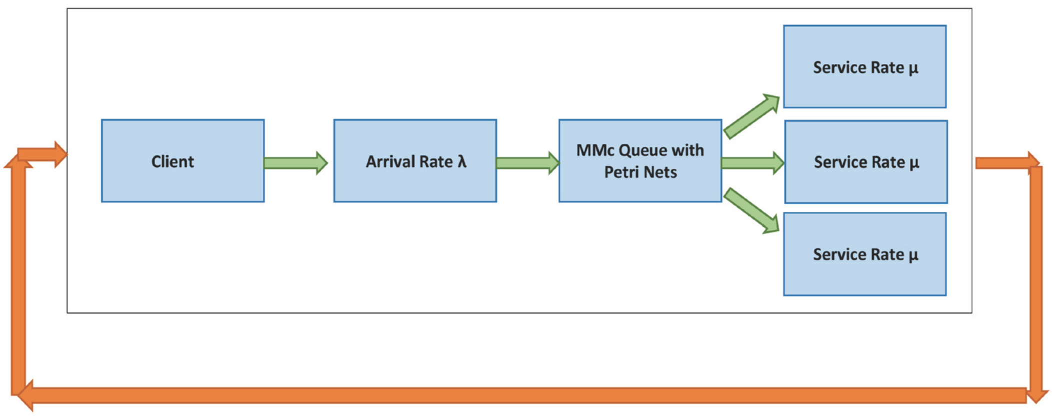


*Figure 5: "M-M-c" Model with Petri Nets.*

# 6. Result and Analysis of Queuing Models Using Petri Nets

Matlab simulation with PN Toolbox is used to implement the "Petri Nets" with queuing model. Total number of servers taken is 5.

**Table 1** shows the various parameters used in execution of "Petri Nets" with queuing model. The proposed model assumes that 20 request per hour will be arriving in a queue (Siddiqui et al., 2020).

*Table 1: Matlab Simulation Parameters*

| ~Type~ | ~Parameter~ | ~Value~ |
|---|---|---|
| "System Configuration" | 'Windows OS' | 'OS' (64-bit) |
| | 'CPU' | "Intel I5", (3.2GHz) |
| | 'RAM' | "4-GB" |
| "M-M-c" Queuing Model | 'Petri Nets(PT)' | "5" |
| | 'Simulation Area' | (50*50) |
| | 'Arrival Rate(AR)' | (20) Request/ hr |

## 6.1. Performance Parameters of Petri Net

Various performance parameters (Koriem et al, 2004) of Petri Nets are as follows: -

### 6.1.1. Expected Number in the System 'E(N)'

E(N) is defined as: -

$$[EN= \rho + \rho 2/ 1\text{-}\rho\ (1+cv2)] \tag{3}$$

'$\rho= \lambda/\mu$'

- "ρ"-> "Server Utilization(SU)"
- "λ"-> "Arrival Rate(AR)"
- "μ"-> "Service Rate(SR)"
- "cv"-> "coefficient" of "service time v"

### 6.1.2. Expected Waiting Time in the System 'E(WT)'

The E(WT) is defined as the time required for waiting in the system to get the desired response.

$$[E(WT)= 1/\mu + \lambda/ 2(1\text{-} \rho)\ E(v2)] \tag{4}$$

### 6.1.3. Expected Response Time in the System 'E(RT)'

E(RT) is defined as the system time taken in producing first response.

$$[E(RT) = 1/\mu(1\text{-} \rho) = 1/(\mu\text{-} \lambda)] \tag{5}$$

**Table 2, 3** shows the various performance of M-M-1 and M-M-c model with Petri net. The number of places in petri net ranges from 0 to 5. The arrival rate(AR) is "λ" and service rate(SR) is "μ" in the given table. The tables have calculated the values of petri net on various servers and their expected waiting time and response time respectively. The values of λ (arrival rate) is assumed to be 20 and μ (service rate) is taken in random manner. The use of Petri Nets helps in evaluating performance of queuing models and thereby lowering waiting time and response time.

*Table 2: Performance of "M-M-1" Queuing Model with Petri Net*

| 'M/M/1' | λ | μ | P0 | P1 | P2 | P3 | P4 | P5 | Expected waiting time in the system 'E(W)' | Expected response time in the system 'E(R)' | Expected number in the system 'E(N)' |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Server1 | 20 | 42 | 0.52 | 0.24 | 0.114 | 0.053 | 0.025 | 0.011 | 0.045 | 0.045 | 0.89 |
| Server2 | 20 | 39 | 0.48 | 0.25 | 0.12 | 0.06 | 0.033 | 0.017 | 0.053 | 0.052 | 1.06 |
| Server3 | 20 | 38 | 0.47 | 0.24 | 0.13 | 0.068 | 0.035 | 0.018 | 0.056 | 0.055 | 1.11 |

| ‘M/M/1’ | λ | μ | P0 | P1 | P2 | P3 | P4 | P5 | Expected waiting time in the system ‘E(W)’ | Expected response time in the system ‘E(R)’ | Expected number in the system ‘E(N)’ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Server4 | 20 | 41 | 0.512 | 0.25 | 0.12 | 0.059 | 0.028 | 0.014 | 0.049 | 0.047 | 0.95 |
| Server5 | 20 | 35 | 0.428 | 0.24 | 0.139 | 0.079 | 0.045 | 0.025 | 0.064 | 0.066 | 1.33 |

*Table 3: Performance of “M-M-c” Queuing Model with Petri net*

| ‘M/M/c’ | λ | μ | P0 | P1 | P2 | P3 | P4 | P5 | Expected waiting time in the system ‘E(W)’ | Expected response time in system ‘E(R)’ | Expected number in the system ‘E(N)’ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Server1 | 20 | 45 | 0.55 | 0.244 | 0.106 | 0.046 | 0.020 | 0.009 | 0.040 | 0.041 | 0.792 |
| Server2 | 20 | 42 | 0.52 | 0.24 | 0.114 | 0.053 | 0.025 | 0.011 | 0.043 | 0.043 | 0.911 |
| Server3 | 20 | 46 | 0.56 | 0.241 | 0.103 | 0.044 | 0.019 | 0.008 | 0.042 | 0.041 | 0.77 |
| Server4 | 20 | 39 | 0.48 | 0.25 | 0.12 | 0.06 | 0.033 | 0.017 | 0.048 | 0.049 | 1.05 |
| Server5 | 20 | 41 | 0.512 | 0.25 | 0.12 | 0.059 | 0.028 | 0.014 | 0.046 | 0.047 | 0.93 |

The bar graph in **figure 6** shows the value of expected number in the system versus the number of servers. From the above graph it is clear that 5th server has the highest number in the system whereas server 1 provides lower number in the system.

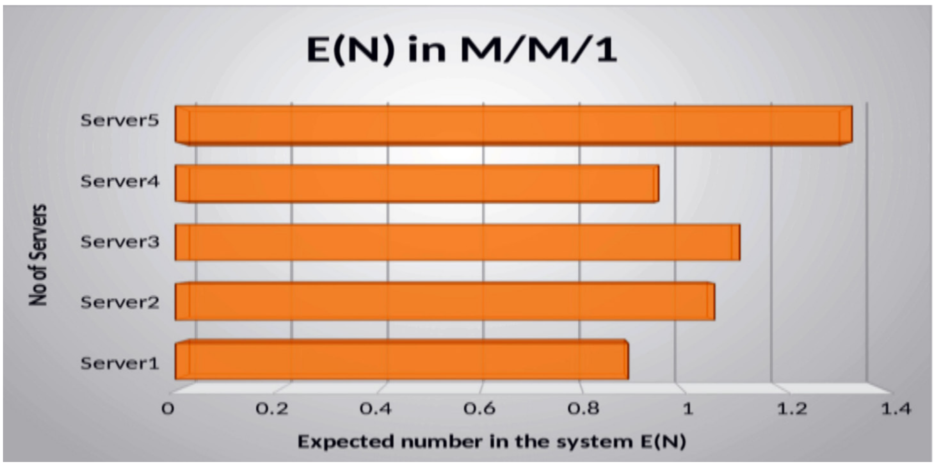


*Figure 6: “Expected Number(EN)” in “M-M-1” Model.*

**Figure 7** shows the expected waiting and response time(RT) in (seconds) in M-M-1 model. The 'E(W)' and 'E(R)' shows variation in time on server 1 to server 5. Server 4 gives minimum expected waiting and response time.

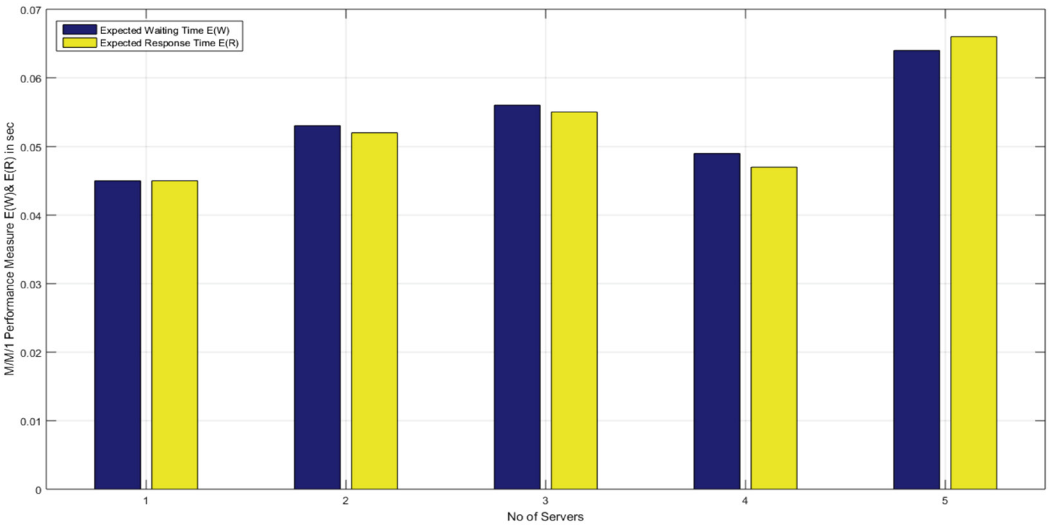


*Figure 7: "Expected Waiting Time" and "Response Time(RT)" in "M-M-1" Model.*

**Figure 8** shows the range of expected number in the system from server 1 to server 5. Different servers have different expected number in the system. The user request varying from 10-150 requests on server 1 to server 5.

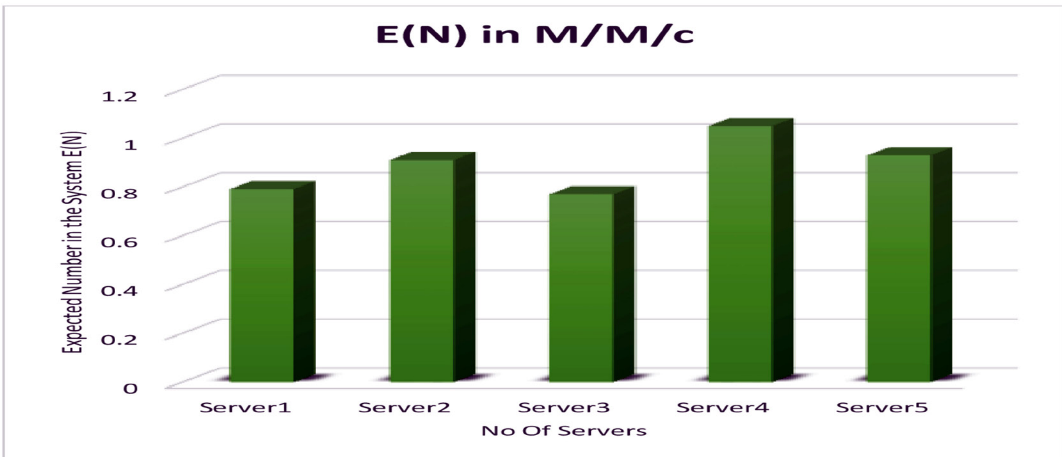


*Figure 8: "Expected Number" in "M-M-c" Model.*

**Figure 9** shows the expected waiting and response time in (seconds) M/M/c model. Server 3 gives lowest E(W) and E(R) as 0.0415 and 0.042 respectively.

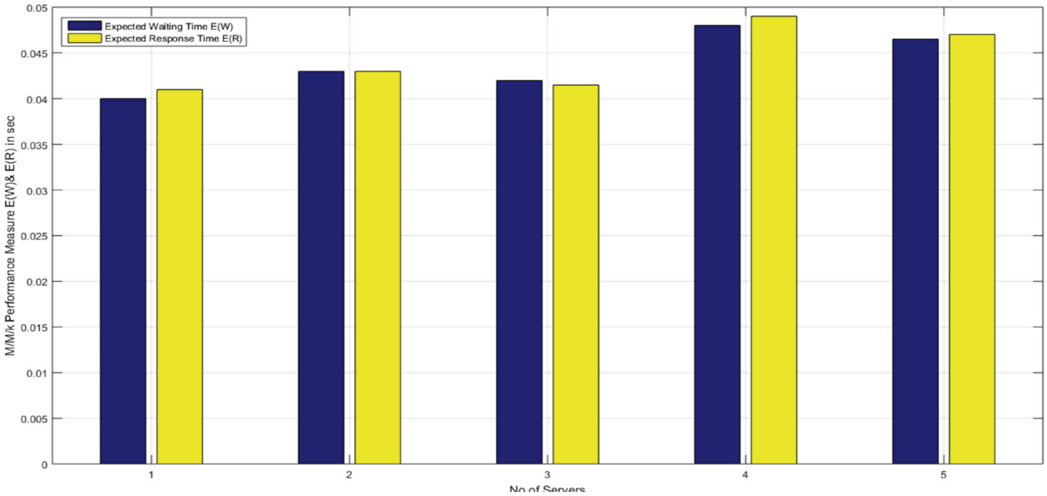


*Figure 9: "Expected Waiting Time" and "Response Time(RT)" in "M-M-c" Model.*

## 7. Conclusion and Future Work

Stochastic Petri Nets are widely used for the graphic representation and simulation of distributed networks. This can be used to define the operation of systems through the expressiveness of Petri Nets' modeling ability. The research paper used petri net to model queuing to boost the efficiency of the units. The authors introduced Petri Nets in the M-M-1 and M-M-C models. The performance evaluation of queuing using stochastic petri~nets[PT] can be done through experimental analysis. Stochastic petri net modeling can be used to evaluate system performance by means of an experimental research. SPNs are analogous to stochastic point processes where one can identify embedded Markov chain. The approach can also be tested via other high level Petri Nets in future.

## 8. References


Agerwala, T. (1979). Special feature: Putting petri nets to work. Computer, (12), 85-94.

Ajmone Marsan, M., Conte, G., & Balbo, G. (1984). A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. ACM Transactions on Computer Systems (TOCS), 2(2), 93-122.

Bakhshandeh, M., Mehrjerdi, Y. Z., & Nasab, H. H. (2019, January). A Proactive Approach to Evaluate Performance of BPEL Workflows by Queuing Theory and Stochastic Petri Net. In 2019 15th Iran International Industrial Engineering Conference (IIIEC) (pp. 52-56). IEEE.

Balsamo, S., & Marin, A. (2007). On representing multiclass M/M/k queues by generalized stochastic Petri nets. In Proc. of ECMS/ASMTA-2007 Conference (pp. 121-128).

Basak, A., & Choudhury, A. (2019). Bayesian inference and prediction in single server M/M/1 queuing model based on queue length. Communications in Statistics-Simulation and Computation, 1-13.

Boukredera, D., & Adel-Aissanou, K. (2020). Modeling and Performance Analysis of Cognitive Radio Networks Using Stochastic Timed Colored Petri Nets. Wireless Personal Communications, 1-29.

Camelo, G. R., Coelho, AS., Borges, RM., & de Souza, RM. (2010). Theory of queues and simulation applied to the shipment of iron ore and manganese at the tip terminal of Madeira. Notebooks of the IME-Série Estatística. 29(2), 1-14.

Khomonenko, A., & Gindin, S. (2016, April). Performance evaluation of cloud computing accounting for expenses on information security. In 2016 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT) (pp. 100-105). IEEE.

Koriem, S. M., Dabbous, T. E., & El-Kilani, W. S. (2004). A new Petri net modeling technique for the performance analysis of discrete event dynamic systems. Journal of systems and software, 72(3), 335-348.\

Luo, Q., Chen, Y., Chen, L., Luo, X., Xia, H., Zhang, Y., & Chen, L. (2019). Research on situation awareness of airport operation based on Petri nets. IEEE Access, 7, 25438-25451.

Pauleve, L., Magnin, M., & Roux, O. (2010). Tuning temporal features within the stochastic π-calculus. IEEE Transactions on Software Engineering, 37(6), 858-871.

Peterson, J. L. (1977). Petri nets. ACM Computing Surveys (CSUR), 9(3), 223-252.

Peterson, J. L. (1981). Petri net theory and the modeling of systems. Prentice Hall PTR.

Siddiqui, S., Darbari, M., & Yagyasen, D. (2020). An QPSL Queuing Model for Load Balancing in Cloud Computing. International Journal of e-Collaboration (IJeC), 16(3), 33-48.

Varela, A. M., Ramírez, J. A. R., Gómez, L. H. H., González, Á. M., & Reyes, M. Y. J. (2015). Lean production system model with Petri nets to support for decision making. Ingeniare, 182-195.

Vijayashree, K. V., & Janani, B. (2018). Transient analysis of an M/M/1 queueing system subject to differentiated vacations. Quality Technology & Quantitative Management, 15(6), 730-748.

Wanini Gonçalves de Araújo, K., de Andrade, M. O., Lima, R. M. F., & de Oliveira, C. A. L. (2020). Performance analysis of metropolitan bus rapid transit line via generalized stochastic petri nets. Journal of Urban Planning and Development, 146(1), 05019019.

Corchado, J. M., Bajo, J., De Paz, Y., and Tapia, D. I., 2008. Intelligent environment for monitoring Alzheimer patients, agent technology for health care. Decision Support Systems, 44(2):382–396.

Corchado, J. M., Pavón, J., Corchado, E. S., and Castillo, L. F., 2004. Development of CBR-BDI agents: a tourist guide application. In Advances in case-based reasoning, pages 547–559. Springer.

Zato, C. et al., 2012. PANGEA–Platform for Automatic coNstruction of orGanizations of intElligent Agents. In *Distributed Computing and Artificial Intelligence*, pages 229–239. Springer.