

Implementation of Fuzzy Logic for M/M/C Model in Packet Switched Network Using HoL Priority Queue

Sampath Kumar. K¹, R. Jahir Hussain²,

Department of mathematics Roever Engineering College¹, Jamal Mohammed College²

Email: sampathjmc1@gmail.com¹, hssnjhr@yahoo.com²

Abstract- In this paper, an HoL priority queue is utilised to give preference to some high priority traffic using fuzzy logic controller. The same controller is also used to choose the number of servers in M/M/C queue model since this represents the speed of service. The service rate has been altered between one and two for altering the C value in the queue. Using MATLAB, the queuing model is simulated and various performance metrics like utilization factor and mean delay time is obtained for a particular period of time. Also, the number of units have been kept at both high and low and based on that the service rate is altered.

Index Terms- Queuing theory, Fuzzy set theory, Telecommunication network, HOL Queue.

1. INTRODUCTION

A Queuing system is a system where the customers or services that are coming in find that the server is preoccupied with another service and hence have to wait in a queue to get serviced. The requests are serviced at a random or a fixed time interval and these services are given to the customers either on a first come first serve basis or on a priority basis, where some customers are given more priority over the others. The queuing systems are used widely for mathematical modelling in various applications like computers and tele-communication networking. E.g. the processing in a computer may send multiple requests to another processing unit and they have to wait their turns to get serviced [1]. The computer or server that hosts the LAN (Local Area Network) will have to wait in queues to make a link with the communication system as per the requirement of the CSMA (Carrier Sense Multiple Access) protocol which has control of this functionality [2]. Retrial queue is a form of queuing system where the arriving requests find the server busy and do not form a queue but exit and then come back again to server their requests. This takes place continuously till their service has been processed [3]. Priority queue is another form of queuing system, where the service with a higher priority is granted a forward position in the queue, whereas a general FCFS (First Come First Serve) queue services the request based on those who first enter the queue [4].

HOL priority queue is also called strict priority queue and is short for Head of Line priority queue. There are multiple real time applications like video-

conferencing and video on demand for this type of queue and the mean delay should not be large in it.

The estimation for the average waiting time may be applied easily to the priority queuing structure for identifying the mean delay in the queue. A modified Head of the line (HOL) priority queue has been introduced and analysed in Maertens et al. [5]. This has been performed to existing HOL queue to solve the starvation problems in general HOL queues. A separate time single server queueing system has been considered with 2 priority queues that have no limit to the capacity. It has been seen that the probability creating functions are used for analysing the contents in the system and packet delay. Performance metrics of the stochastic quantities has been derived and necessary expressions of the probabilities have been obtained from functions that generate the probability using dominant singularity method. The significance and impact of the modified HOL priority queue on these have been measured from the performance metrics. This paper has studied the literature of various queuing models including priority queues, retrial queues and fuzzy logic control. The method of calculating the performance measure has been given by defining the equations and then the step by step methodology has been given. After this, the work has been implemented and the results have been displayed and discussed. Finally, the work has been concluded.

2. LITERATURE REVIEW

There have been multiple researchers in the last twenty years who have studied the queuing system on the basis of negative and positive customers. Any queue with negative customers is known as G queues and have accumulated a large amount of interest due

to wide usage in neural networking, computers and communicating server [6], [7]. This type of queue has been used widely with the customers perceived negatively and named after Gelenbe who had initially introduced this type of queuing [8], [9]. The distribution of the waiting time has been analysed in this type of M/G/C retrial queue. The moments of the distribution of the waiting time has also been obtained from [10]. In the case of an arbitrary customers, it is more difficult to analyse the distribution for the retrial queue when compared to the number of requests in the queue. Analysing the distribution is a complicated process since the processes operate randomly for servicing the requests. In an M/M/C queue, the author gave only the first two moments in the distribution of the waiting time. A repeating method to compute the variance of the distribution of waiting time has been developed in Hanschke [11]. There has been several literature for the distribution of the M/M/C queuing system and it has been seen that the calculation of the distribution for waiting time is very difficult, whereas the average waiting time for the steady state is obtained with ease using Little's expression [12].

As per the available data, [13], [14] relied on a combination of queueing and scheduling concepts to resolve problems associated with static scheduling, with only two recent research articles [15], [16], and a paper that was prepared subsequently [13], suggested modes of dynamic scheduling that would optimally utilise developments in both these areas.

There has been lots of interest related to retrial queueing among the researchers in the field of telecommunication queueing theory. Hence, these queues have been studied in detail in Artalejo and Gomez-Corral [17] Kim et al. [18]. From the different literature, it has been assumed that the arrival rate of the requests is characterised by a fixed Poisson technique or its different Markovian extensions like PH, MAP and BMAP [18]. Moreover, the queues are also used for modelling mathematically for various computer applications. The network with packet switching has a LAN system with a shared bus that is operating under carrier sense MAP (Multiple Access Protocol).

Akar and Sohraby [19] proposed a retrial queueing framework for FDL optical barriers in nonconcurrent optical switching hubs. This model utilized optical packet and post reservation sort are permitted to re-flow over the FDLs in a probabilistic way. The simulation results show this algorithm exact and quick as well as permitting one to utilize the methodology to measurement optical buffers in next-generation optical packet switching frameworks.

Sohraby and Kim [20] proposed a retrial queueing system for the process of generating peakedness. Further, they had driven the numerical algorithm for calculation of peakedness departure process through M/M/1 single server retrial queue. It shows decreasing of retrial function rate as well as maximum ratio equal to the Poisson process in peakedness function. Chang and Jang [21] proposed an analytical approach by time scaling technique accessible CRN performance in terms of throughput; radio spectrum occupancy and average delay with considerably decrease the computation complexity. The systematic outcomes mathematically compared and demonstrated with CRN systems in the existing approach.

Wei and Xu [22] suggested a self-tuning technique for end to end QoS guarantees in web applications using fuzzy control technique. These Fuzzy controllers have in the recent past been used to resolve multiple industrial control issues of a complex nature. It was proposed that fuzzy control of multi-rate traffic in networks that were multimedia cell-based were to be used for this issue. Further, it was also suggested to use independent and free resource management in data centers that were virtualized by relying on fuzzy logic-based approaches. In literature pertaining to fuzzy logic, fuzzy queues are deeply analysed in Mary et al. [23] Kanyinda et al. [24]. A majority of these studies were concentrated on finding system performance measures by utilizing the method of alpha-cuts.

Wu and Tan [25] examine a multi-server retrial queueing approach, in which the clients opt to exit the system as a result of impatience. By choosing this innovative method, it is possible to read the status of the queue, the desires of those in the queue and the nature of urgency of such requests. Furthermore, they analysed the impact of retrial, buffer size on steady state conditions and correlation between the different parameters. The simulation results are analyzed by relying on OPNET Modeler tool. Nagy et al. [26] had an objective to create and build a model that simulated the IMS (IP Multimedia Subsystem) network on the basis of single servers with FIFO queues whose capacity was unlimited in Matlab environment. The product of the devised algorithms received from simulations are assessed, and comparisons were conducted with similar results. It was proved that the algorithms vastly enhanced the performance of IMS in comparison to a standardized solution. stated that Uncertainty Measures were linked with Fuzzy Procedures for connection admission control in asynchronous transfer mode networks.

Consequent to the above suggestions, queueing processes with repetitive attempts have been extensively used to create numerous solutions for practical problems faced in tele-communication systems, telephone switching operations,

communication systems and computers. The aim of this analysis is to offer a review of the work undertaken by researchers in the theory of retrial queueing that is of relevance to scheduling, and to offer a basis for investigating the integration of queueing theory and scheduling methods for resolving problems associated with dynamic scheduling. Recent literature on different types of queueing systems can be found in Chang et al. [27] Dimitriou [28].

3. PERFORMANCE MEASURES OF QUEUEING SYSTEMS

Priority queueing is used to segregate different classes in a network. They allow some traffic to bypass the waiting time and go to the service. Head of line Priority queue (HOL) is a type of priority queue where high priority is given more preference when compared to low priority traffic. This means that high priority traffic is scheduled ahead even if the arrival time is late. This is performed by assigning lower delay time for the high priority traffic. This is not a good thing in some systems since if the network contains large portion of high priority traffic, then the performance of the low priority traffic will be damaged and cause delay in large scales for the low priority network and this is known as starvation problem. However, this problem occurs only when there are lots of high priority traffic. If the amount of high priority traffic is very less, then tis starvation problem will not exist.

The performance metrics for a queueing system with a single servicing window is obtained by calculating the utilisation factor, response time, length of the queue and the time taken for waiting.

Utilisation factor: The utilisation factor gives the fraction of the time where the server is not free and is attending to some other request. It can be expressed as the ratio of rate of arrival to the rate of service. It is represented by the expression in equation 1.

$$\rho = \frac{\lambda}{m\mu} + \frac{\lambda_s}{k\mu_s} \tag{1}$$

$$\rho' = \frac{\lambda'}{n\mu'} + \frac{\lambda_s}{k\mu_s} \tag{2}$$

Queue length: The length of the queue represents the maximum extent to where the number of customers is defined in the queueing system. The length can be represented by equation 3 and 4.

$$E[N_{tot}] = E[N] + E[N_s] \tag{3}$$

$$E[N_{tot}'] = E[N'] + E[N_s] \tag{4}$$

Where

$$E[N] = m \left(\frac{1}{m\mu} \right) + \left(\frac{1}{m\mu} \right) \frac{\left(\frac{\lambda}{\mu} \right)^m}{m!} \frac{1}{\left[\sum_{x=0}^{m-1} \frac{\left(\frac{\lambda}{\mu} \right)^x}{x!} + \frac{\left(\frac{\lambda}{\mu} \right)^m}{m!} \frac{1}{\left(1 - \frac{\lambda}{m\mu} \right)} \right]} \left(1 - \frac{\lambda}{m\mu} \right)^2$$

$$E[N'] = n \left(\frac{1}{n\mu'} \right) + \left(\frac{1}{n\mu'} \right) \frac{\left(\frac{\lambda'}{\mu'} \right)^m}{n!} \frac{1}{\left[\sum_{x=0}^{n-1} \frac{\left(\frac{\lambda'}{\mu'} \right)^x}{x!} + \frac{\left(\frac{\lambda'}{\mu'} \right)^m}{n!} \frac{1}{\left(1 - \frac{\lambda'}{m\mu'} \right)} \right]} \left(1 - \frac{\lambda'}{m\mu'} \right)^2$$

$$E[N_s] = k \left(\frac{1}{Kn\mu} \right) + \left(\frac{1}{Kn\mu} \right) \frac{\left(\frac{\lambda}{\mu} \right)^k}{K!} \frac{1}{\left[\sum_{x=0}^{K-1} \frac{\left(\frac{\lambda_s}{\mu_s} \right)^x}{x!} + \frac{\left(\frac{\lambda_s}{\mu_s} \right)^k}{K!} \frac{1}{\left(1 - \frac{\lambda_s}{k\mu_s} \right)} \right]} \left(1 - \frac{\lambda_s}{k\mu_s} \right)^2$$

For internet server

Waiting time: The time taken for each request to get processed in the queue is known as the waiting time. It can be expressed as the ration of mean length of the queue to the rate of arrival. It can be represented by the expression in equation 5 and 6.

$$E[W_{tot}] = E[W] + E[W_s] \tag{5}$$

$$E[W_{tot}'] = E[W'] + E[W_s] \tag{6}$$

Where

$$E[W] = m \left(\frac{1}{m\mu} \right) + \left(\frac{1}{m\mu} \right) \frac{\left(\frac{\lambda}{\mu} \right)^m}{m!} \frac{1}{\left[\sum_{x=0}^{m-1} \frac{\left(\frac{\lambda}{\mu} \right)^x}{x!} + \frac{\left(\frac{\lambda}{\mu} \right)^m}{m!} \frac{1}{\left(1 - \frac{\lambda}{m\mu} \right)} \right]} \left(1 - \frac{\lambda}{m\mu} \right)^2$$

$$E[W'] = n \left(\frac{1}{n\mu'} \right) + \left(\frac{1}{n\mu'} \right) \frac{\left(\frac{\lambda'}{\mu'} \right)^m}{n!} \frac{1}{\left[\sum_{x=0}^{n-1} \frac{\left(\frac{\lambda'}{\mu'} \right)^x}{x!} + \frac{\left(\frac{\lambda'}{\mu'} \right)^m}{n!} \frac{1}{\left(1 - \frac{\lambda'}{m\mu'} \right)} \right]} \left(1 - \frac{\lambda'}{m\mu'} \right)^2$$

$$E[W_s] = k \left(\frac{1}{Kn\mu} \right) + \left(\frac{1}{Kn\mu} \right) \frac{\left(\frac{\lambda}{\mu} \right)^k}{K!} \frac{1}{\left[\sum_{x=0}^{K-1} \frac{\left(\frac{\lambda_s}{\mu_s} \right)^x}{x!} + \frac{\left(\frac{\lambda_s}{\mu_s} \right)^k}{K!} \frac{1}{\left(1 - \frac{\lambda_s}{k\mu_s} \right)} \right]} \left(1 - \frac{\lambda_s}{k\mu_s} \right)^2$$

Source: Adopted from Raghunath and Sravani [29]

Response time: The total time taken by a customer for both waiting and servicing is known as the response time.

4. METHODOLOGY

A single server queue can only serve one user at a time. However, using multiple servers are also not feasible since more cost and resources are required to maintain the servers. Hence, in this research, the number of servers will not be fixed, but will vary between one and two. The number of servers will change depending upon the queue length. Predominantly, only one server will be used which means that $C = 1$. When the queue gets filled up, the arriving users will start to leave. There is no waiting space once the queue gets filled. To avoid this, the number of servers will increase to two when the queue gets filled up, i.e $C = 2$.

The rate of arrival and service is determined by the Poisson process and hence is known as M/M/c model. The arrival rate is selected at $\lambda=0$, which means that it will always be a positive value. However, when the queue gets filled, renegeing starts taking place. If a customer is waiting for a too long time in a queue, the confidence in the service is lost and the user starts to leave the queue and this is called renegeing. This can be solved by increasing the number of servers or reducing the service time. Since service time is determined by the speed of the service, it is not advisable to change it and hence the service server is increased to two. Hence the model used in this study will be both M/M/1 and M/M/2. To alternate between single server and dual server, a controller is required to determine the number of people in the queue and services. If the number of servicing stations are kept constant at $C = 2$, controller is not required since the queue will never get filled up. But, the number of resources used will be high and it will not be efficient to keep the servicing stations constantly at 2 and using a controller is mandatory for efficient processing. Fuzzy logic is used in this study as a controller to shift between the two services.

The rules are framed in the fuzzy logic by assigning the membership functions in the fuzzy logic controller in such a way that it contains the maximum queue length. When the queue length increases and reaches the peak, the fuzzy rules are compared and finds that the maximum queue length has been achieved. After this, the fuzzy controller increases the number of servers from one to two. Similarly, when the number of users in the queue goes below the prescribed queue length set in the fuzzy rules, the

fuzzy controller activates and reduces the number of servers from two to one.

The customers arrive based on the Poisson's process and usually those who arrive first are serviced first. However, some may have to be given a higher preference over the others and hence priority is given to some customers over others. Head of Line (HOL) priority queue will be used in this study for giving preference to the required services. In this research, the queue is analysed whether the priority is low or high using fuzzy logic. For this, the rules have been framed in the fuzzy logic controller. The priority is generated by the users and a value is assigned to each element in the traffic. There is no separate queue to the high priority traffic in this work. All the traffic enter the same queue, the status of the priority is scanned and the any high priority traffic is sent directly to the front of the queue. When the queue gets empty, there is no one to serve and hence the resources in the server gets wasted. To solve this problem, it is assumed that the server leaves on a vacation. A single vacation requires a work that is not a negative value with variable V and with a speed of $\beta > 0$. The length of the vacation is random and aits end, the server is inspected to check if another vacation is required or if the customers are to be served. N policy will be used in this study where the vacations will be of a fixed length. Once again, after the vacation, the service is started and continues till all the units are served and then if the queue is again empty, it takes a vacation. At the time of vacation, the model is only M/M.

The renewal point is the time taken for the vacation, setting up of termination, completing the service, or the starting time of service. The status of the method for the time t can be represented using the Markov process

$$\{(C(t), X(t), \xi(t)), t \geq 0\}$$

where "C(t)" represents the state of the server, whether it is free or busy or on vacation. "X(t)" represents the number of requests currently waiting to be processed, " $\xi^*(t)$ " signifies the time that has been elapsed from the last time of renewal. The function $v_e(x)$, $\mu(x)$, $\eta(x)$ and $\delta(x)$ represent the rate of completing the requests repeatedly for servicing, vacation and setting up. From the above expressions, detailed representations can be obtained as shown in equation 7.

$$v_e(x) = \frac{a_e(x)}{1 - A_e(x)}, \mu(x) = \frac{b(x)}{1 - b(x)}, \eta(x) = \frac{v(x)}{1 - v(x)}$$

$$\delta(x) = \frac{s(x)}{1 - s(x)} \tag{7}$$

Where

$$A_e(x) = \frac{1}{a_1} \int_0^x (1 - A(x)) dx \text{ and } a_e(x) = \frac{1 - A(x)}{a_1}$$

After this, the model has to be analysed to see if it is stable. Initially, the embedded Markov chain is analysed at the completed locations. It is assumed that $\{t_n, n \in N\}$ represents the series of time of completions for the services.

Performance measures and numerical results

Let U represent the server utilisation in terms of whether the server busy or free or on vacation,

$$u = Q(1) = \frac{\lambda b_1 P_0 [\lambda p(s_1 + v_1) + V^*(\lambda) S^*(\lambda) A_e^*(\lambda)]}{V^*(\lambda) S^*(\lambda) [1 + A_e^*(\lambda) - \lambda(b_1 + q s_1 + q v_1)]} \quad (8)$$

Where p_0 was given in theorem

$$L_s = K(1) = \frac{P_0}{V^*(\lambda) S^*(\lambda)} - \frac{NM - UV}{2M^2}$$

Where

$$M = 1 + A_e^*(\lambda) - \lambda(b_1 + q s_1 + q v_1)$$

$$N = \left(\frac{7}{3} q V^*(\lambda) S^*(\lambda) - 2p\right) \lambda (b_1 - A_e^*(\lambda)) - \left(\frac{10}{3} q V^*(\lambda) S^*(\lambda) + p\right) \times (s_2 + 2s_1 v_1 + v_2) + V^*(\lambda) S^*(\lambda) \quad (9)$$

$$U = \lambda b_1 V^*(\lambda) S^*(\lambda) (q A_e^*(\lambda) + 1) - 2q V^*(\lambda) S^*(\lambda) A_e^*(\lambda) + \lambda(s_1 + v_1)(p + q V^*(\lambda) S^*(\lambda))$$

$$V = 2A_e^*(\lambda) [\lambda b_1 + q \lambda (s_1 + v_1) - 1] \lambda^2 b_2 - 2\lambda^2 q b_1 (s_1 + v_1) - q \lambda^2 (s_2 + 2s_1 v_1 + v_2)$$

Due to little formula, we have

$$W_s = \frac{L_s}{\lambda} = \frac{P_0}{\lambda V^*(\lambda) S^*(\lambda)} - \frac{NM - UV}{2M^2} \quad (9)$$

Source: Adopted from Wenhui [30]

$$P(z) = \frac{P_0}{V^*(\lambda) S^*(\lambda)} (1 - A_e^*(\lambda)) - \left\{ \frac{pD(z) + B^*(\lambda - \lambda z) V^*(\lambda) [p + qD(z)] - p - V^*(\lambda) S^*(\lambda)}{1 - C(z) B^*(\lambda - \lambda z) [p + qD(z)]} \right\} \quad (10)$$

$$Q(z) = \frac{P_0}{V^*(\lambda) S^*(1-z)} (1 - B^*(\lambda z)) - \left\{ \frac{pD(z) C(z) + V^*(\lambda) S^*(\lambda) [1 - C(z)] - pC(z)}{1 - C(z) B^*(\lambda - \lambda z) [p + qD(z)]} \right\} \quad (11)$$

$$R(z) = \frac{P_0}{V^*(\lambda) S^*(1-z)} (1 - V^*(\lambda - \lambda z)) - \left\{ \frac{(1 - C(z)) q V^*(\lambda) B^*(\lambda - \lambda z) + p(1 - B^*(\lambda - \lambda z) C(z))}{1 - C(z) B^*(\lambda - \lambda z) [p + qD(z)]} \right\} \quad (12)$$

$$S(z) = \frac{P_0 V^*(\lambda - \lambda z)}{V^*(\lambda) S^*(1-z)} (1 - S^*(\lambda - \lambda z)) - \left\{ \frac{(1 - C(z)) q V^*(\lambda) S^*(\lambda) B^*(\lambda - \lambda z) + p(1 - B^*(\lambda - \lambda z) C(z))}{1 - C(z) B^*(\lambda - \lambda z) [p + qD(z)]} \right\} \quad (13)$$

Where

$$E(z) = B^*(\lambda - \lambda z) V^*(\lambda) C(z) \{q[D(z)(1-z) - 1] + (1-p)z\},$$

$$F(z) = p B^*(\lambda - \lambda z) (1-z) (D(z) - 1),$$

$$P_0 = \frac{V^*(\lambda) S^*(\lambda) [1 + A_e^*(\lambda) - \lambda(b_1 + q s_1 + q v_1)]}{V^*(\lambda) S^*(\lambda) [1 + A_e^*(\lambda)] [1 - q \lambda (s_1 + v_1)] + p \lambda (s_1 + v_1)}$$

5. RESULTS AND DISCUSSION

The hybrid single server M/M/1 and dual server M/M/2 is simulated. Fuzzy logic is used to change the number of servers between one and two. The parameters were fixed in MATLAB Simulink with

$$\text{Arrival rate} = 0.01$$

$$\text{Servicing Rate} = 0.01$$

These values have been fixed in the simulation and can be changed whenever necessary. However, the arrival rate is not changed in this work and only the servicing rate changed whenever necessary.

5.1. M/M/2

The number of servers has been fixed at 2. The number of units entering into the system has been fixed at 50. These 50 units are spread across different areas. The average number of clients entering the system is shown in figure 1. It can be seen that the mean number of clients is increasing as the time goes and peaks at 0.9 seconds. The delay time and all other performance metrics depend upon the number of clients. Some of the arriving clients are given high priority and this is found by using HOL priority queuing. The priority of the arriving clients are obtained and when they are found to have a high priority, they are sent directly to the front of the queue. The average delay time for each unit is shown in figure 2. As the time increases, the delay time is also seen to be increased. It can be seen that the delay is directly proportional to the number of clients and have similar graphs. The mean delay time also peaks at 0.9 seconds just like number of clients. The maximum delay that occurs is the 0.1 hours which is 6 minutes.

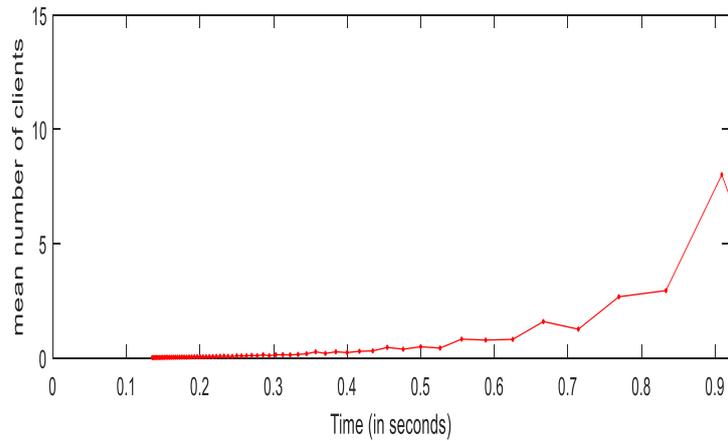


Fig. 1. Average number of clients

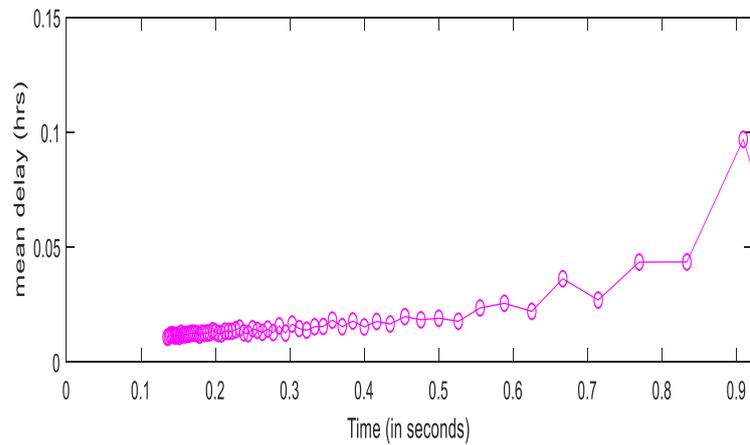


Fig . 2. Mean delay time for each unit

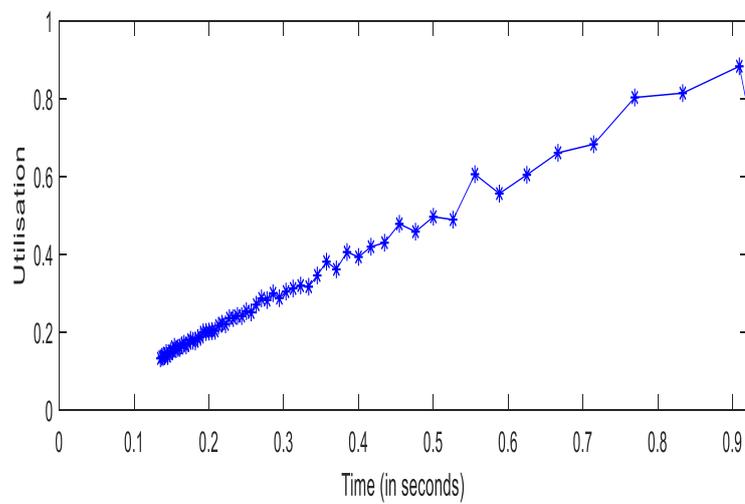


Fig. 3. Utilisation factor

The utilisation factor of the queue is seen in figure 3. As the time increases, the utilisation of the queue also increases since it is directly proportional to the number of clients. The factor increases linearly as time increases and peaks at 0.9 seconds. After this it starts to decrease. It can be seen that the average number of clients, mean delay time and the utilization factor never gets saturated at the peak since there are two servers and hence the queue never gets filled.

5.2. M/M/1

The number of servers serving the users have been decreased to one. Hence, the queue may not get filled if too much traffic enters the queue. This means that the length of the queue gets limited and the number of

users seem to be high. The mean number of clients coming has now been increased as shown in figure 4. The number of clients remain high at around 400 and any new traffic that enters the system will not stay in the queue. This work has been simulated for a total of 5 seconds. Priority queuing is followed in this model too.

The utilisation factor has been shown in figure 5. The queue has been used to the full extend which means that the queue remains full continuously throughout the period. The fuzzy is designed in such a way that when the utilization factor reaches the peak, it should shift to M/M/2. As seen in M/M/2, the utilization factor is directly proportional to the average number of clients

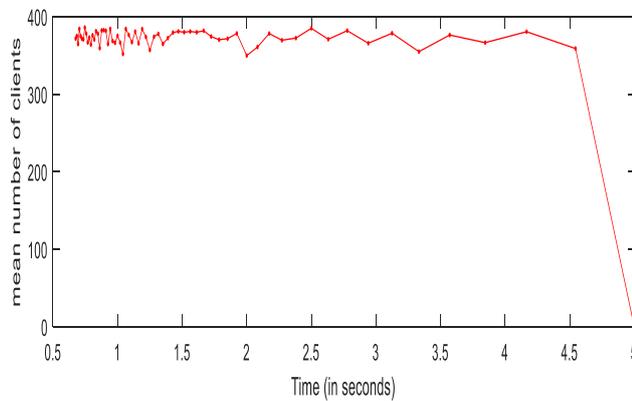


Fig. 5. Mean number of clients

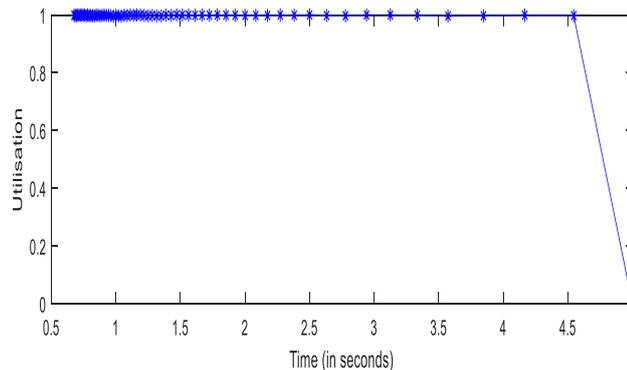


Fig. 4. Utilisation Factor

The simulated results have been shown in table 1. The arrival rate, average queue rate, average delay rate and the utilization rate has been displayed and compared for multiple values. It can be seen that all the metrics depend upon the arrival rate and is directly proportional.

Performance measure of the average waiting time of units of second priority in the queue. From various levels of possibilities, the corresponding membership functions are obtained, since the metrics are evaluated using the membership functions and not by the available values. More information is provided for during the queuing system. The presented approaches provide membership functions of the corresponding fuzzy measures for the performance metrics like the

of expected customers for the queue.

6. CONCLUSION

The queuing model has been simulated for both single server and dual server by changing the service rate between 1 and 2. Fuzzy logic has been used to make this shift and HOL priority queue is decide the person getting serviced.

Table 1. Simulated Results.

Average Queue rate	Average delay rate	Arrival Rate	Utilization rate
8.0134	0.0969	423.4153	0.8825
2.9486	0.0435	383.5796	0.8135
2.6770	0.0434	383.5661	0.8023
1.2611	0.0269	404.3121	0.6822
1.5937	0.0362	53.2576	0.6602
0.8146	0.0220	101.8864	0.6036
0.7840	0.0255	95.2520	0.5554
0.4347	0.0177	7.9123	0.4885
0.4926	0.0190	19.9029	0.4957
0.3119	0.0165	19.8553	0.4291
0.2375	0.0152	7.7846	0.3926
0.2720	0.0178	7.7846	0.4051
0.2016	0.0153	7.7846	0.3607
0.0939	0.0144	4.2607	0.2531
0.0693	0.0122	3.4108	0.2407
0.0710	0.0127	3.4108	0.2410

The model has been simulated for low number of units and also higher number of units and results have been displayed. When the model used is M/M/1, the length of the queue has to be high to compensate lots of people. When the length of the queue gets reduced, renegeing occurs for the model, but works well when the server amount is increased to M/M/2.

REFERENCES

[1] Choi, B. D.; & Chang, Y. (1999): 'Single server retrial queues with priority calls', *Mathematical and Computer Modelling*, **30**(3–4), pp. 7–32.
 [2] Kumar, B. K.; Vijayalakshmi, G.; Krishnamoorthy, A.; & Basha, S. S. (2010): 'A single server feedback retrial queue with collisions', *Computers & Operations Research*, **37**(7), pp. 1247–1255.
 [3] Atencia, I.; & Moreno, P. (2005): 'A single-server retrial queue with general retrial times and Bernoulli schedule', *Applied Mathematics and Computation*, **162**(2), pp. 855–880.

[4] Kalyanaraman, R. (2012): 'A Feedback Retrial Queueing System with Two Types of Batch Arrivals', *International Journal of Stochastic Analysis*, pp. 1–20.
 [5] Maertens, T.; Walraevens, J.; & Bruneel, H. (2007): 'A modified HOL priority scheduling discipline: Performance analysis', *European Journal of Operational Research.*, **180**(3), pp. 1168–1185.
 [6] Daduna, H. (2001): 'Queueing networks - customers, signals, and product form solutions', *Journal of Applied Mathematics and Stochastic Analysis*, **14**(4), pp. 421–426.
 [7] Harrison, P. G.; & Marin, A. (2014): 'Product-Forms in Multi-Way Synchronizations', *The Computer Journal*, **57**(11), pp. 1693–1710.
 [8] Gelenbe, E. (1989): 'Random Neural Networks with Negative and Positive Signals and Product Form Solution', *Neural Computation*, **1**(4), pp. 502–510.
 [9] Kim, J.; & Kim, B. (2013): 'Waiting time distribution in an retrial queue', *Performance Evaluation*, **70**(4), pp. 286–299.
 [10] Dombacher, C.; & Lenaugasse, N. (2010): 'Queueing Models for Call Centres', *telecomm.*
 [11] Hanschke, T. (1986): 'A Computational Procedure for the Variance of the Waiting Time in the M/M/1/1 Queue with Repeated Attempts', in *DGOR*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 525–532.
 [12] Kim, B.; & Kim, J. (2011): 'Higher moments of the waiting time distribution in M/G/1 retrial queues', *Operations Research Letters*, **39**(3), pp. 224–228.
 [13] Terekhov, D.; Tran, T. T.; Down, D.; & Beck, J. C. (2014): 'Integrating Queueing Theory and Scheduling for Dynamic Scheduling Problems', *Journal of Artificial Intelligence Research*, **50**, pp. 535–572.
 [14] Nazarathy, Y.; & Weiss, G. (2010): 'A fluid approach to large volume job shop scheduling', *Journal of Scheduling*, **13**(5), pp. 509–529.
 [15] Tran, T. (2011): 'Using Queueing Analysis to Guide Combinatorial Scheduling in Dynamic Environments', University of Toronto.
 [16] Terekhov, D. (2013): 'Integrating Combinatorial Scheduling with Inventory Management and Queueing Theory', University of Toronto.
 [17] Artalejo, J. R.; & Gomez-Corral, A. (2008): *Retrial Queueing Systems: A Computational Approach*. Tokyo: Springer.
 [18] Kim, C. S.; Klimenok, V.; & Dudin, A. (2014): 'Retrial Queue with Lattice Distribution of Inter-Arrival Times and Constant Retrial Rate', in *2014 European Modelling Symposium*, pp. 437–441.
 [19] Akar, N.; & Sohraby, K. (2011): 'Retrial queueing models of multi-wavelength FDL feedback

- optical buffers', *IEEE Transactions on Communication*, **59**(10), pp. 2832–2840.
- [20] Sohraby, K.; & K. Kim, (2013): 'On the impact of retrials on traffic burstiness', in *2013 21st Iranian Conference on Electrical Engineering, ICEE 2013*, pp. 1–5.
- [21] Chang, W. S.; & Jang, W. M. (2014): 'Spectrum occupancy of cognitive radio networks: a queueing analysis using retrial queue', *IET Networks*, **3**(3), pp. 218–227.
- [22] Wei, J.; & Xu, C.-Z. (2005): 'A Self-tuning Fuzzy Control Approach for End-to-End QoS Guarantees in Web Servers', in *Quality of Service – IWQoS 2005*, pp. 123–135.
- [23] Mary, K. J. R.; Monica, P.; & Mythili, S. (2014): 'Performance measures of fm/fm/1 queueing system with n-policy', *International Journal of Physics and Mathematical Sciences*, **4**(3), pp. 5–10.
- [24] Kanyinda, J. P. M.; Matendo, R. M. M.; & Lukata, B. U. E. (2015): 'Computing Fuzzy Queueing Performance Measures by L-R Method', *Journal of Fuzzy Set Valued Analysis*, **2015**(1), pp. 57–67.
- [25] Wu, M.; & Tan, L.; 'An infinite-source M/M/S retrial queueing network model with balking and impatient customers', in *2013 IEEE Global High Tech Congress on Electronics, GHTCE 2013*, pp. 25–29.
- [26] Nagy, L.; Tombal, J.; & Novotny, V. (2013): 'Proposal of a queueing model for simulation of advanced telecommunication services over IMS architecture', in *2013 36th International Conference on Telecommunications and Signal Processing, TSP 2013*, pp. 326–330.
- [27] Chang, F.M.; Liu, T.H.; & Ke, J.C. (2018): 'On an unreliable-server retrial queue with customer feedback and impatience', *Applied Mathematical Modelling*, **55**, pp. 171–182.
- [28] Dimitriou, I.; 'A two-class queueing system with constant retrial policy and general class dependent service times', *European Journal of Operational Research*, **270**(3), pp. 1063–1073.
- [29] Raghunath, Y. T. N. V.; & Sravani, A. S. (2012): 'Performance Measures for Internet Server By Using M/M/M Queueing Model', *International Journal of Research in Engineering and Technology*, **1**(4), pp. 619–623, 2012.
- [30] Wenhui, Z. (2005): 'Analysis of a single-server retrial queue with FCFS orbit and Bernoulli vacation', *Applied Mathematics and Computation*, **161**(2), pp. 353–364.