Applied Mathematical Sciences, Vol. 4, 2010, no. 23, 1141 – 1154

M/M/1 Retrial Queueing System with N-Policy Multiple Vacation under Non-Pre-Emtive Priority Service by Matrix Geometric Method

G. AYYAPPAN

Pondicherry Engineering College, Pondicherry, India
ayyappanpec@hotmail.com

A. MUTHU GANAPATHI SUBRAMANIAN

Tagore Arts College, Pondicherry, India
csamgs1964@gmail.com

GOPAL SEKAR

Tagore Arts College , Pondicherry , India
gopsek28@yahoo.co.in

Abstract

Consider a single server retrial queueing system with N-**policy multiple vacation under non-pre-emptive priority** service in which two types of customers arrive in a Poisson process with arrival rate λ_1 for low priority customers and λ_2 for high priority customers.The **vacation rate** follows an exponential distribution with parameter α.The service times follow an exponential distribution with parameters μ_1 **and** μ_2 for both types of customers. The concepts of **retrial and N-policy multiple vacations** are introduced for low priority customers only. Let K be the maximum number of waiting spaces for high priority customers in front of the service station. The high priorities customers will be governed by the **Non-Pre-emptive priority service**.The access from orbit to the service facility are governed by the **classical retrial policy.** This model is solved by using Matrix geometric Technique.Numerical study have been done for Analysis of Mean number of low priority customers in the orbit (MNCO), Mean number of high priority customers in the queue(MPQL),Truncation level (OCUT),Probability of server free and Probabilities of server busy with low, high priority customers and server in vacation for various values of λ_1, λ_2, μ_1, μ_2, **N, α , σ** and k in elaborate manner and also various particular cases of this model have been discussed.

Mathematics Subject Classification: 60K25, 65K30

Keywords: Single Server – Stochastic nature – Low priority – high priority – Non-Pre-emptive priority service – N-policy multiple vacation - Matrix Geometric Method – Orbit – classical retrial policy.

1. INTRODUCTION

Queueing systems in which arriving customers who find all servers and waiting positions (if any) occupied may retry for service after a period of time are called **Retrial queues([2],[3],[4],[7],[8],[9]),.** Because of the complexity of the retrial queueing models, analytic results are generally difficult to obtain. There are a great number of numerical and approximations methods are available, in this paper we will place more emphasis on the solutions by **Matrix geometric method ([12]).**

2. DESCRIPTION OF QUEUEING SYSTEM

Consider a single server retrial queueing system with N**-policy multiple vacation under non-pre-emptive priority([10])** service in which two types of customers arrive in a Poisson process with arrival rate λ_1 for low priority customers and λ_2 for high priority customers.The **vacation rate** follows an exponential distribution with parameter **α**.These customers are identified as **primary calls.** Further assume that the service times follow an exponential distribution with parameters μ_1 **and** μ_2 for both types of customers. The concepts of **retrial and N-policy** are introduced for low priority customers only. Let K be the maximum number of waiting spaces for high priority customers in front of the service station.

Description of N-policy multiple vacations:

The concept of N-policy multiple vacations was first introduced by Yadin and Naor and later it was investigated by Lee, Srinivasan and Kella in queueing system who described it as "The service does not start unless there are N customers in the system except at the beginning.Once the server begins the service, the server continues the service until all customers are served after which the server must compulsorily go for a vacation. If the number of waiting customers in the system at any vacation completion is less than N then the server continues to be in vacation (multiple vacations). If the server returns from the vacation and finds atleast N customers in the system then he immediately starts to serve the waiting customers".

The above policy is modified and has been introduced for priority service in Retrial Queueing System. **The N-policy multiple vacations for priority service in retrial queueing system is governed by the following principals**

1. The service does not start unless there are N customers in the low priority (orbit) or there is atleast one customer in the high priority except at the beginning.
2. Once the service begins, as per non-pre-emptive priority service, the server continues the service until all customers are served both from high priority queue and orbit (low) after which he must compulsorily go for a vacation.
3. If the number of waiting customers in the orbit at any vacation completion is less than N and if there is no customer in the high priority queue then the server continues to be in vaction (multiple vacation)
4. If the server returns from the vacation and finds atleast N customers in the orbit or atleast one customer in the high priority queue then he immediately starts to serve the waiting cuastomers as per the non-pre-emptive priority service.

This model is developed by using the above principal and numerical study is carried out in elaborate manner for various values of λ_1, λ_2, μ_1, μ_2, **N**, α **and K** .

If the server is **free** at the time of a primary call (low/high) arrival, the arriving call begins to be served immediately by the server and customer leaves the system after service completion. Otherwise, if the server is **busy** then the low priority arriving customer **goes to orbit** and becomes a source of **repeated calls**. The pool of sources of repeated calls may be viewed as a sort of queue. Every such source produces a Poisson process of repeated calls with intensity σ. If an incoming repeated call (low) finds the server **free**, it is served and leaves the system after service, while the source which produced this repeated call disappears. If any one of the waiting spaces is occupied by the high priority customers then the low priority customers (as a primary call) can not enter into service station and **goes to orbit**. If the server is **busy and there are some waiting spaces** then the high priority customer can enter into the service station and waits for his service. If there are no waiting spaces then the high priority customers can not enter into the service station and will be **lost** for the system. Otherwise, the system state does not change.

If there are no customers in the system after completion of service, the server must complulsorily go on vacation. At any time he may return from the vacation and starts service according to **N-policy multiple vacation rules** as defined above. If the server is engaging with low priority customer and at that time the higher priority customer comes then the high priority customer will get service only after completion of the service of low priority customer who is in service. This type of priority service is called the **Non-pre-emptive priority service ([5],[6],[8]).** This kind of priority service is followed in this paper.

Retrial Policy:

Most of the queueing system with repeated attempts assume that each customer in the retrial group seeks service independently of each other after a random time exponentially distributed with rate σ so that the probability of repeated attempt during the interval (t, $t+\Delta t$) given that there were n customers in orbit at time t is **$n\sigma\ \Delta t + O(\Delta t)$.** This discipline for access for the server from the retrial group is called **classical retrial rate policy.** The input flow of primary calls (low and high), interval between repetitions,service times,interval between returns from vacation are mutually independent.

3. MATRIX GEOMETRIC METHODS

Let N(t) be the random variable which represents the number of low priority customers in the orbit at time t and P(t) be the random variable which represents the number of high priority customers in the queue (in front of the service station) at time t and S(t) represents the server state at time t. The random process is described as{ < N(t) , P(t), S(t) > / N(t)=0,1,2,3… ; P(t)=0,1,2,3…K ; S(t) =0,1,2,3}.If S(t)=0 then server is idle. If S(t)=1 or 2 then server busy with either low or high priority customer. If S(t)=3 then server is in vacation.
The possible state spaces are

{ (u , v , w) / u = 0,1,2,3…. ; v = 0; w=0,1,2,3 } U
{ (u , v , w)/ u = 0,1,2,3…. ; v =1,2,3…k; w=1,2,3}

The infinitesimal generator matrix Q is given below

$$Q = \begin{pmatrix} A_{00} & A_0 & O & O & O & O & . & . \\ A_{10} & A_{11} & A_0 & O & O & O & . & . \\ O & A_{21} & A_{22} & A_0 & O & O & . & . \\ O & O & A_{32} & A_{33} & A_0 & O & . & . \\ . & . & . & . & . & . & . & . \end{pmatrix}$$

Notations:

$\#_1$: $-(\lambda_1+\lambda_2)$ $\#_2$: $-(\lambda_1+\lambda_2+\mu_1)$ $\#_3$: $-(\lambda_1+\lambda_2+\mu_2)$ $\#_4$: $-(\lambda_1+\lambda_2+\alpha)$
$\#_5$: $-(\lambda_1+\mu_1)$ $\#_6$: $-(\lambda_1+\mu_2)$ $\#_7$: $-(\lambda_1+\alpha)$ $\#_8$: $-(n\sigma+\lambda_1+\lambda_2)$
$\#_9$: $-(M\sigma+\lambda_1+\lambda_2)$ $\#_{10}$: $-(\lambda_2+\mu_1)$ $\#_{11}$: $-(\lambda_2+\mu_2)$ $\#_{12}$: $-(\lambda_2+\alpha)$

A_{00}, $A_{n\,n-1}$, A_{nn}, $A_{n\,n+1}$ are square matrices of order 3k+4

$$
\mathbf{A_{00}} = \begin{pmatrix}
\#_1 & \lambda_1 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & \#_2 & 0 & \mu_1 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & \#_3 & \mu_2 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 0 & \#_1 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & \mu_1 & 0 & \#_2 & 0 & 0 & \lambda_2 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & \mu_2 & 0 & 0 & \#_3 & 0 & 0 & \lambda_2 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & \alpha & 0 & 0 & 0 & \#_4 & 0 & 0 & \lambda_2 & \ldots & 0 & 0 & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \#_5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & \#_6 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & \#_7
\end{pmatrix}
$$

$\mathbf{A_{nn-1} = (a_{ij})}$ **where** $\mathbf{a_{ij}} = 0$ for all i and j except i=1 and j=2

$\mathbf{a_{12} = n\sigma}$ **n=1,2,3….**

$$
\underset{n=1,2,3..N-1}{\mathbf{A_{nn}}} = \begin{pmatrix}
\#_8 & \lambda_1 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
\mu_1 & \#_2 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
\mu_2 & 0 & \#_3 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 0 & \#_1 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & \mu_1 & 0 & \#_2 & 0 & 0 & \lambda_2 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & \mu_2 & 0 & 0 & \#_3 & 0 & 0 & \lambda_2 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & \alpha & 0 & 0 & 0 & \#_4 & 0 & 0 & \lambda_2 & \ldots & 0 & 0 & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & \#_5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & \#_6 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & \#_7
\end{pmatrix}
$$

$$
\underset{n=N,N+1,..}{\mathbf{A_{nn}}} = \begin{pmatrix}
\#_8 & \lambda_1 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
\mu_1 & \#_2 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
\mu_2 & 0 & \#_3 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
\alpha & 0 & 0 & \#_4 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & \mu_1 & 0 & \#_2 & 0 & 0 & \lambda_2 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & \mu_2 & 0 & 0 & \#_3 & 0 & 0 & \lambda_2 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & \alpha & 0 & 0 & 0 & \#_4 & 0 & 0 & \lambda_2 & \ldots & 0 & 0 & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & \#_5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & \#_6 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & \#_7
\end{pmatrix}
$$

$\mathbf{A_{nn-1} = (b_{ij})}$ **where** $\mathbf{b_{ii}} = \lambda_1$ if $i=2,3,4\ldots$ $(n=1,2,3,\ldots)$
$= 0$ if $\mathbf{i \neq j}$

If the capacity of the orbit is finite say M then

$$\mathbf{A_{MM}} = \begin{pmatrix} \#_9 & \lambda_1 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ \mu_1 & \#_{10} & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ \mu_2 & 0 & \#_{11} & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ \alpha & 0 & 0 & \#_{12} & 0 & 0 & \lambda_2 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & \mu_1 & 0 & \#_{10} & 0 & 0 & \lambda_2 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & \mu_2 & 0 & 0 & \#_{11} & 0 & 0 & \lambda_2 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & \alpha & 0 & 0 & 0 & \#_{12} & 0 & 0 & \lambda_2 & \ldots & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & -\mu_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & -\mu_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & -\alpha \end{pmatrix}$$

Let **x** be a steady-state probability vector of Q and partitioned as
x = (x(0),x(1),x(2), ….) and **x** satisfies

$$\mathbf{xQ = 0\ ,\ xe=1.} \qquad \mathbf{(A)}$$

Where $x(i) = (P_{i00}\ ,\ P_{i01}, P_{i02}, P_{i03}, P_{i11}, P_{i12}, P_{i13}, P_{i21}, P_{i22}, P_{i23}, \ldots\ldots P_{ik1}, P_{ik2}, P_{ik3})$
$(i=0,1,2,3,\ldots)$

4. DIRECT TRUNCATION METHOD

In this method one can truncate the system of equations in (A) for sufficiently large value of the number of customers in the orbit, say M. That is, the orbit size is restricted to M such that any arriving customer finding the orbit full is considered lost. The value of M can be chosen so that the loss probability is small. Due to the intrinsic nature of the system in (A), the only choice available for studying M is through algorithmic methods. While a number of approaches is available for determining the cut-off point, M , The one that seems to perform well (w.r.t approximating the system performance measures) is to increase M until the largest individual change in the elements of **x** for successive values is less than Є a predetermined infinitesimal value.

5. STABILITY CONDITION

Theorem :

The inequality $L*(\lambda_1/\mu_1) < 1$ **where** $\mathbf{L = F / ((1-x)(1-\pi_{3k}-\pi_{3k+1}) + x\pi_{3k-2})(1+x/y)}$

$F = (1+t+t^2+\ldots\ldots+t^{k-1})$, $t= \lambda_2/(\lambda_2+\mu_1)$, $s= \lambda_2/(\lambda_2+\alpha)$, $x = \lambda_2/\mu_2$ and $y= \mu_1/\mu_2$ is the necessary and sufficient condition for the system to be stable.**([1],[11])**

Proof:

Let Q be an infinitesimal generator matrix for the queueing system (without retrial)

The stationary probability vector X satisfies

$$XQ = 0 \quad \text{and} \quad Xe=1 \quad \ldots\ldots\ldots \quad (1)$$

The Rate matrix R satisfies the equation

$$A_0+RA_1+R^2A_2 = 0 \quad \ldots\ldots\ldots\ldots\ldots(2)$$

The system is stable if sp(R)<1

We know that the Matrix R satisfies sp(R)<1 if and only if

$$\Pi A_0 e < \Pi A_2 e \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

where $\Pi=(\pi_0,\pi_1,\pi_2,\ldots\pi_{3k},\pi_{3k+1},\pi_{3k+2})$ satisfies

$$\Pi A = 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

$$\Pi e = 1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

where

$$A=A_0+A_1+A_2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(6)$$

A_0, A_1, A_2 are square matrices of order 3k+3

$A_0 = \lambda_1 I$ where I is unit matrix

$$A_1 = \begin{pmatrix}
\#_2 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & \#_3 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & \#_4 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & \mu_1 & 0 & \#_2 & 0 & 0 & \lambda_2 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & \mu_2 & 0 & 0 & \#_3 & 0 & 0 & \lambda_2 & 0 & \ldots & 0 & 0 & 0 \\
0 & \alpha & 0 & 0 & 0 & \#_4 & 0 & 0 & \lambda_2 & \ldots & 0 & 0 & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & \#_5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & \#_6 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & \#_7
\end{pmatrix}$$

$$A_2 = \begin{pmatrix}
\mu_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
\mu_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
\alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0
\end{pmatrix}$$

By substituting A0 , A1 , A2 in equations (4) ,(5) and (6) ,we get

$$\boldsymbol{\pi_1 = x\pi_0 \quad \pi_3 = t\pi_0 \quad \pi_5 = s\pi_2 = 0 \quad \pi_2 = 0}$$
$$\boldsymbol{\pi_4 = x(\pi_1+\pi_3) \quad \pi_6 = t^2\,\pi_0 \quad \pi_8 = s^2\pi_2 = 0}$$
$$\boldsymbol{\pi_7 = x(\pi_4+\pi_6) \quad \pi_9 = t^3\pi_0 \quad \pi_{11} = s^3\pi_2 = 0}$$
$$\boldsymbol{\pi_{10} = x(\pi_7+\pi_9)}$$
$$\vdots \qquad \vdots \qquad \vdots$$
$$\boldsymbol{\pi_{3k+1} = (\lambda_2/\mu_2)\pi_{3k-2} \quad \pi_{3k} = (\lambda_2/\mu_1)\pi_{3k-3} \quad \pi_{3k+2} = (\lambda_2/\alpha)\pi_{3k-1} = 0}$$

From (5)

$$\boldsymbol{\pi_0+\pi_1+\pi_2+\pi_3+\pi_4+\pi_5\ldots\ldots+\pi_{3k}+\pi_{3k+1}+\pi_{3k+2} = 1}$$
$$\boldsymbol{\pi_0+(\pi_1+\pi_4+\pi_7+\pi_{10}+\ldots\pi_{3k-2})+(\pi_3+\pi_6+\pi_9+\ldots\pi_{3k-3})} = 1 - \boldsymbol{\pi_{3k}+\pi_{3k+1}}$$
$$\pi_0 F = (1-x)[1-\pi_{3k}-\pi_{3k+1}] + x\pi_{3k-2}$$
$$\pi_0 = [(1-x)(1-\pi_{3k}-\pi_{3k+1}) + x\pi_{3k-2}]\,(F^{-1})$$

where $\mathbf{F = (1+t+t^2+\ldots\ldots+t^{k-1})}$ **,** $\mathbf{t = \lambda_2/(\lambda_2+\mu_1)}$**,** $\mathbf{s = \lambda_2/(\lambda_2+\alpha)}$**,** $\mathbf{x = \lambda_2/\mu_2}$ **nd** $\mathbf{y = \mu_1/\mu_2}$

From (3)

$$\boldsymbol{(\lambda_1/\mu_1) < \pi_0\,(1+x/y)}$$
$$\boldsymbol{(\lambda_1/\mu_1) < [(1-x)(1-\pi_{3k}-\pi_{3k+1}) + x\pi_{3k-2}]\,(F^{-1})(1+x/y)}$$
$$\boldsymbol{L^*(\lambda_1/\mu_1) < 1 \text{ where } L = F / ((1-x)(1-\pi_{3k}-\pi_{3k+1}) + x\pi_{3k-2})(1+x/y)}$$

The inequality $L * (\boldsymbol{\lambda_1/\mu_1}) < \mathbf{1}$ is also a sufficient condition for the retrial queueing system to be stable. Let Qn be the number of customers in the orbit after departure n^{th} customer from the service station. we first prove the embedded Markov chain {Qn , n>=0} is erogdic if $\mathbf{L * (\lambda_1/\mu_1) < 1}$. It is readily to see that {Qn , n>=0} is irreducible and aperiodic. It remains to be proved that {Qn , n>0} is positive recurrent. The irreducible and aperiodic Markov chain {Qn , n>0} is positive recurrent if $|\psi_i| < \infty$ for all i and $\lim_{i\to\infty} \sup\ \psi_i < 0$ where

$$\boldsymbol{\psi_i = E(\,Qn+1 - Qn / Qn = i)} \quad (i=0,1,2,3\ldots)$$
$$\boldsymbol{\psi_i = L * (\lambda_1/\mu_1) - i\sigma / (\lambda_1 + \lambda_2 + i\sigma)}$$

if $\mathbf{L * (\lambda_1/\mu_1) < 1}$, then $|\psi_i| < \infty$ for all i and $\lim_{i\to\infty} \sup\ \psi_i < 0$. Therefore the embedded Markov chain{Qn , n>0} is ergodic.

6. ANALYSIS OF STEADY STATE PROBABILITIES

We are applying **Direct Truncation Method** to find Steady state probability vector **x**. Let M denote the cut-off point or Truncation level. The steady state probability vector $\mathbf{x}^{(M)}$ is now partitioned as $\mathbf{x}^{(M)}$ = (x(0) , x(1), x(2) , …..x(M)) and $\mathbf{x}^{(M)}$ satisfies

$$\mathbf{x}^{(M)}\,Q = 0\ ,\ \mathbf{x}^{(M)}\,e = 1.$$

where $x(i) = (P_{i00}, P_{i01}, P_{i02}, P_{i03}, P_{i11}, P_{i12}, P_{i13}, P_{i21}, P_{i22}, P_{i23}, \ldots\ldots P_{ik1}, P_{ik2}, P_{ik3})$ (i=0,1,2,3,…M)

The above system of equations is solved by Numerical method such as GAUSS-JORDAN elementary transformation method. Since there is no clear cut choice for M, we may start the iterative process by taking, say M=1 and increase it until the individual elements of **x** do not change significantly. That is, if M^* denotes the truncation point then $\|\mathbf{x}^{M^*}(\mathbf{i}) - \mathbf{x}^{M^*-1}(\mathbf{i})\|_\infty < \varepsilon$ where ε is an infinitesimal quantity.

7. SPECIAL CASES

1. It becomes Single server retrial queueing system with single vacation exhaustive type service under non-pre-emptive priority service if N=0
2. This model becomes Single server retrial queueing system with non-pre-emptive priority service if N=0 and $\alpha \to \infty$ **([5])**
3. This model becomes Single Server Retrial queueing system and coincide with analytic solutions given by Falin[8] and Templeton[8] for various values of $\boldsymbol{\lambda_1}$, $\boldsymbol{\lambda_2} \to 0$, $\boldsymbol{\mu_1}$, $\boldsymbol{\mu_2} \to \infty$, $\boldsymbol{\alpha} \to \infty$, N=0,**σ and K**
4. This model becomes Single Server Standard Queueing System and coincide with standard results if $\boldsymbol{\lambda_2} \to 0$, $\boldsymbol{\mu_2} \to \infty$, $\boldsymbol{\alpha} \to \infty$, N=0 and $\sigma \to \infty$

8 SYSTEM PERFORMANCE MEASURES

Numerical study has been dealt in very large scale to study the following measures. Here we define

P(u,0,0) = Probability that there are u customers(low) in the orbit and no Customers in the high priority queue and server is free .

P(u,v,1) = Probability that there are u customers(low) in the orbit and v>=0 customers in the high priority queue and server is busy with low priority customer.

P(u,v,2) = Probability that there are u customers(low) in the orbit and v >=0 customers in the high priority queue and server is busy with high priority customer.

P(u,v,3) = Probability that there are u customers(low) in the orbit and v customers in the high priority queue and server is in vacation.

We can find various probabilities for various values of λ_1, $\boldsymbol{\lambda_2}$, $\boldsymbol{\mu_1}$, $\boldsymbol{\mu_2}$, α , **σ, N and K** and the following system measures can be easily study with these probabilities

1. The probability mass function of Server state

Let S(t) be the random variable which represents the server state at time t .

S :	$\mathbf{0}_{\mathbf{idle}}$	$\mathbf{1}_{\mathbf{low}}$	$\mathbf{2}_{\mathbf{high}}$	$\mathbf{3}_{\mathbf{vacation}}$
P :	$\sum_{i=0}^{\infty} p(i,0,0)$	$\sum_{i=0}^{\infty}\sum_{j=0}^{k} p(i,j,1)$	$\sum_{i=0}^{\infty}\sum_{j=0}^{k} p(i,j,2)$	$\sum_{i=0}^{\infty}\sum_{j=0}^{k} p(i,j,3)$

2. The probability mass function of number of customers(low) in the orbit

Let X(t) be the random variable which represents the number of low priority customers in the orbit.

No.of low priority customers (orbit) Probability i $\sum_{j=0}^{k}\sum_{l=1}^{3} p(i,j,l) +$ p(i,0,0) (i=0,1,2…)

3. The Probability mass function of number of high priority customers (queue).

Let P(t) be number of high priority customers in the queue at time t. P(t) takes the values 0,1,2,3….K.

No.of high priority customers (queue)	Probability
0	$\sum_{i=0}^{\infty}\sum_{l=0}^{3} p(i,0,l)$
j	$\sum_{i=0}^{\infty}\sum_{l=1}^{3} p(i,j,l)$ **(j=1,2…k)**

4. The Mean number of high priority customers in the queue

MNHP = . $\sum_{j=1}^{k} j*(\sum_{i=0}^{\infty}\sum_{l=1}^{3} p(i,j,l))$

5. The Mean number of low priority customers in the orbit

MNCO = $(\sum_{i=0}^{\infty} i*(\sum_{j=0}^{k}\sum_{l=1}^{3} p(i,j,l) + \mathbf{p(i,0,0)}))$

6. **The probability that the orbiting customer (low) is blocked**

Blocking Probability = $\sum_{i=1}^{\infty}\sum_{j=0}^{k}\sum_{l=1}^{3} p(i,j,l)$

7. The probability that an arriving customer(low/high) enter into service station immediately

PSI = $\sum_{i=0}^{\infty} p(i,0,0)$

9. NUMERICAL STUDY

MNCO : Mean Number of Customers in the Orbit
MPQL : Mean Number of high priority customers in front of the service station
P0 : Probability that the server is idle
P1 : Probability that the server is busy with low priority customers
P2 : Probability that the server is busy with high priority customers
P_3 : Probability that the server is in vacation

From the following tables we conclude that

- Mean number of cutomers in the orbit decreses as σ increases.
- Probabilities P1 and P2 are independent of σ.
- As σ increases, P_0 decreases and P_3 increases

Table 1 : System Measures for λ_1=10 λ_2=5 μ_1=20 μ_2=25 N=5 K=2

Sigma	Ocut	P0	P1	P2	P3	MNCO	MPQL
10	41	0.2258	0.5000	0.1942	0.0800	4.1543	0.1867
20	38	0.1526	0.5000	0.1941	0.1532	3.3182	0.1911
30	37	0.1142	0.5000	0.1941	0.1917	3.0612	0.1934
40	36	0.0911	0.5000	0.1941	0.2148	2.9368	0.1948
50	36	0.0757	0.5000	0.1941	0.2302	2.8636	0.1957
60	36	0.0648	0.5000	0.1941	0.2412	2.8153	0.1964
70	35	0.0566	0.5000	0.1940	0.2494	2.7810	0.1969
80	35	0.0502	0.5000	0.1940	0.2557	2.7555	0.1973
90	35	0.0451	0.5000	0.1940	0.2608	2.7357	0.1976
100	35	0.0410	0.5000	0.1940	0.2650	2.7200	0.1978
200	35	0.0214	0.5000	0.1940	0.2846	2.6497	0.1990
300	35	0.0144	0.5000	0.1940	0.2915	2.6264	0.1994
400	35	0.0109	0.5000	0.1940	0.2951	2.6148	0.1996
500	35	0.0088	0.5000	0.1940	0.2972	2.6079	0.1997
600	35	0.0073	0.5000	0.1940	0.2987	2.6033	0.1998
700	35	0.0063	0.5000	0.1940	0.2997	2.6000	0.1999
800	35	0.0055	0.5000	0.1940	0.3005	2.5975	0.1999
900	35	0.0049	0.5000	0.1940	0.3011	2.5956	0.2000
1000	35	0.0044	0.5000	0.1940	0.3016	2.5941	0.2000
2000	35	0.0022	0.5000	0.1940	0.3038	2.5872	0.2001
3000	35	0.0015	0.5000	0.1940	0.3045	2.5849	0.2002
4000	35	0.0011	0.5000	0.1940	0.3049	2.5837	0.2002
5000	35	0.0009	0.5000	0.1940	0.3051	2.5830	0.2002
6000	35	0.0007	0.5000	0.1940	0.3053	2.5826	0.2002
7000	35	0.0006	0.5000	0.1940	0.3054	2.5822	0.2002
8000	35	0.0006	0.5000	0.1940	0.3054	2.5820	0.2002
9000	35	0.0005	0.5000	0.1940	0.3055	2.5818	0.2002

Table 2 : System Measures for λ_1=10 λ_2=5 μ_1=20 μ_2=25 N=5 K=4

Sigma	Ocut	P0	P1	P2	P3	MNCO	MPQL
10	43	0.2225	0.5000	0.1997	0.0778	4.2941	0.2092
20	40	0.1506	0.5000	0.1997	0.1497	3.4302	0.2137
30	39	0.1128	0.5000	0.1997	0.1875	3.1645	0.2161
40	38	0.0900	0.5000	0.1997	0.2103	3.0360	0.2175
50	38	0.0748	0.5000	0.1997	0.2255	2.9602	0.2184
60	38	0.0640	0.5000	0.1997	0.2363	2.9103	0.2191
70	38	0.0559	0.5000	0.1997	0.2444	2.8750	0.2196
80	38	0.0496	0.5000	0.1997	0.2507	2.8486	0.2200
90	37	0.0446	0.5000	0.1997	0.2557	2.8282	0.2203
100	37	0.0405	0.5000	0.1997	0.2598	2.8119	0.2206
200	37	0.0211	0.5000	0.1997	0.2792	2.7392	0.2218
300	37	0.0143	0.5000	0.1997	0.2860	2.7152	0.2222
400	37	0.0108	0.5000	0.1997	0.2895	2.7033	0.2224
500	37	0.0087	0.5000	0.1997	0.2916	2.6961	0.2226
600	37	0.0072	0.5000	0.1997	0.2930	2.6913	0.2227
700	37	0.0062	0.5000	0.1997	0.2941	2.6879	0.2227
800	37	0.0054	0.5000	0.1997	0.2948	2.6854	0.2228
900	37	0.0049	0.5000	0.1997	0.2954	2.6834	0.2228
1000	37	0.0044	0.5000	0.1997	0.2959	2.6818	0.2228
2000	37	0.0022	0.5000	0.1997	0.2981	2.6747	0.2230
3000	37	0.0015	0.5000	0.1997	0.2988	2.6723	0.2230
4000	37	0.0011	0.5000	0.1997	0.2992	2.6711	0.2230
5000	37	0.0009	0.5000	0.1997	0.2994	2.6704	0.2231
6000	37	0.0007	0.5000	0.1997	0.2996	2.6699	0.2231
7000	37	0.0006	0.5000	0.1997	0.2997	2.6696	0.2231
8000	37	0.0006	0.5000	0.1997	0.2997	2.6693	0.2231
9000	37	0.0005	0.5000	0.1997	0.2998	2.6691	0.2231

Table 3 : System Measures for λ_1=10 λ_2=5 μ_1=20 μ_2=25 N=5 K=6

Sigma	Ocut	P0	P1	P2	P3	MNCO	MPQL
10	43	0.2224	0.5000	0.2000	0.0777	4.3029	0.2110
20	40	0.1505	0.5000	0.2000	0.1495	3.4376	0.2155
30	39	0.1127	0.5000	0.2000	0.1873	3.1715	0.2178
40	39	0.0899	0.5000	0.2000	0.2101	3.0427	0.2193
50	39	0.0747	0.5000	0.2000	0.2253	2.9669	0.2202
60	39	0.0639	0.5000	0.2000	0.2361	2.9169	0.2209
70	39	0.0558	0.5000	0.2000	0.2442	2.8815	0.2214
80	39	0.0496	0.5000	0.2000	0.2504	2.8550	0.2218
90	39	0.0446	0.5000	0.2000	0.2555	2.8346	0.2221
100	39	0.0405	0.5000	0.2000	0.2595	2.8183	0.2223
200	39	0.0211	0.5000	0.2000	0.2789	2.7455	0.2236
300	39	0.0143	0.5000	0.2000	0.2857	2.7214	0.2240
400	39	0.0108	0.5000	0.2000	0.2892	2.7095	0.2242
500	39	0.0087	0.5000	0.2000	0.2914	2.7023	0.2243
600	39	0.0072	0.5000	0.2000	0.2928	2.6975	0.2244
700	39	0.0062	0.5000	0.2000	0.2938	2.6941	0.2245
800	39	0.0054	0.5000	0.2000	0.2946	2.6915	0.2245
900	39	0.0048	0.5000	0.2000	0.2952	2.6896	0.2246
1000	39	0.0044	0.5000	0.2000	0.2956	2.6880	0.2246
2000	39	0.0022	0.5000	0.2000	0.2978	2.6808	0.2247
3000	39	0.0015	0.5000	0.2000	0.2986	2.6784	0.2248
4000	39	0.0011	0.5000	0.2000	0.2989	2.6773	0.2248
5000	39	0.0009	0.5000	0.2000	0.2991	2.6765	0.2248
6000	39	0.0007	0.5000	0.2000	0.2993	2.6761	0.2248
7000	39	0.0006	0.5000	0.2000	0.2994	2.6757	0.2248
8000	39	0.0006	0.5000	0.2000	0.2995	2.6755	0.2248
9000	39	0.0005	0.5000	0.2000	0.2995	2.6753	0.2248

10. GRAPHICAL STUDY

Fig 1. Mean No. of low priority customers in the orbit for λ_1 = 10 λ_2=5 μ_1 =20 μ_2=25 N=5 K=6 and σ various from 10 to 90

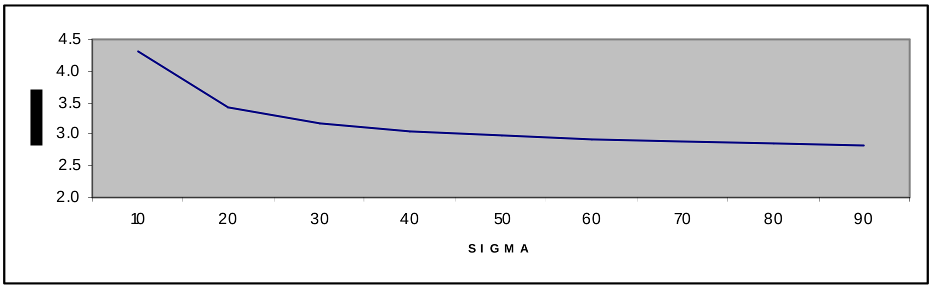


Fig 2. Mean No. of low priority customers in the orbit for λ_1 = 10 λ_2=5 μ_1 =20 μ_2=25 N=5 K=6 and σ various from 100 to 900

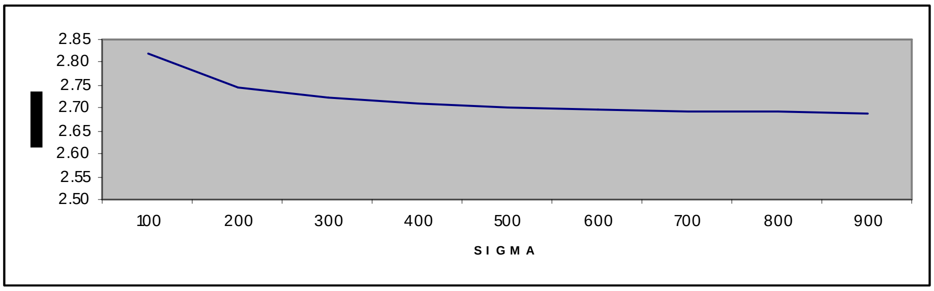
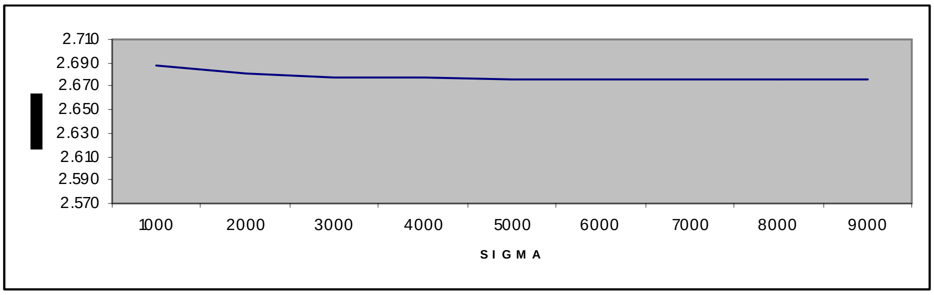


Fig 3. Mean No. of low priority customers in the orbit for λ_1 = 10 λ_2=5 μ_1 =20 μ_2=25 N=5 K=6 and σ various from 1000 to 9000



11. CONCLUSIONS

1. The Numerical study on single server retrial queueing system with N-policy multiple vacations under non pre-emptive priority service by Matrix Geometric Method have been done in elobarate manner for various values of $\boldsymbol{\lambda_1, \lambda_2, \mu_1, \mu_2, \alpha}$,**N**,$\boldsymbol{\sigma}$
2. If N=0 then our results coincides with single server retrial queueing system with single vacations-exhaustive type service under non pre-emptive priority service
3. The numerical results were obtained by us coincide with Analytic solutions of single Server Retrial Queueing System with Non-pre-emptive priority service for various values of $\boldsymbol{\lambda_1, \lambda_2, \mu_1, \mu_2, \sigma}$, ($\boldsymbol{\alpha} \rightarrow \infty$ and N=**0) and K** is large
4. From this numerical study,further we state that when retrial rate is high i.e σ>8000, and ($\boldsymbol{\alpha} \rightarrow \infty$ and **N=0)**, these results coincide with standard Single server queueing system with Non- pre-emptive priority service for various values of $\boldsymbol{\lambda_1, \lambda_2, \mu_1, \mu_2}$ **and K is large**
5. The numerical results were obtained by us coincide with Analytic solutions of single Server Retrial Queueing System (discussed by Falin and Templeton) for various values of $\boldsymbol{\lambda_1}$, **(** $\boldsymbol{\lambda_2 \rightarrow 0}$**)** , $\boldsymbol{\mu_1, \mu_2 \rightarrow \infty}$ **,** $\boldsymbol{\sigma}$ **,** ($\boldsymbol{\alpha} \rightarrow \infty$ and **N=0) and K is large**
6. Graphical studies show the impact of retrial rate on length of orbit (low).

REFERENCES

[1] Altman, E. and A.A. Borovkov (1997). On the stability of retrial queues. *Queueing Systems* 26, 343-363.

[2] Artalejo, J.R. (1995). A queueing system with returning customers and waiting line. *Operations* Research *Letters* 17, 191-199.

[3] Artalejo, J.R. and A. Gomez-Corral (1999b). Performance analysis of a single-server queue with repeated attempts. *Mathematical* and *Computer Modelling* 30, 79-88.

[4] Artalejo, J.R. (1999*b*). A classified bibliography of research on retrial queues: Progress in 1990-1999. *Top*, 7, 187-211.

[5] Ayyappan.G, A.Muthu Ganapathi Subramanian and Gopal Sekar (2009). A single server retrial queueing system with non-pre-emptive priority service by matrix geometric method , Proceeding of International conference of Mathematical and Computational models, PSGTECH Coimbatore ,India

[6] Falin, G.I., J.R. Artalejo and M. Martin (1993). On the single server retrial queue with priority customers. *Queueing Systems* 14, 439-455.

[7] Falin, G.I. (1990). A survey of retrial queues. *Queueing Systems 7,*127-167.

[8] Falin, G.I. and J.G.C. Templeton (1997). *Retrial Queues.* Chapman and Hall, London

[9] Farahmand, K. (1990). Single line queue with repeated demands. *Queueing Systems* 6, 223-228.

[10] Ksirhnareaddy.G.K, R.Natarajan and R.Arumuganathan (1998). Analysis of bulk queue N-policy multiple vacations and setup times . Computers and Operations Research Vol 25 issue11 pp 957-967

[11] Liang, H.M. and V.G. Kulkarni (1993a). Stability condition for a single-server retrial queue. *Advances in Applied Probability* 25, 690- 701. Soon Seok Lee , Ho Woo Lee, Seung Hyun Yoon and Chae.K (1995) Batch arrival queue with N-Policy single vacation . Computers and Operations Research Vol 22 issue 2 pp 173-189

[12] Marcel F. Neuts, Matrix Geometric Solutions in Stochastic Models an algorithmic approach.

Received: August, 2009