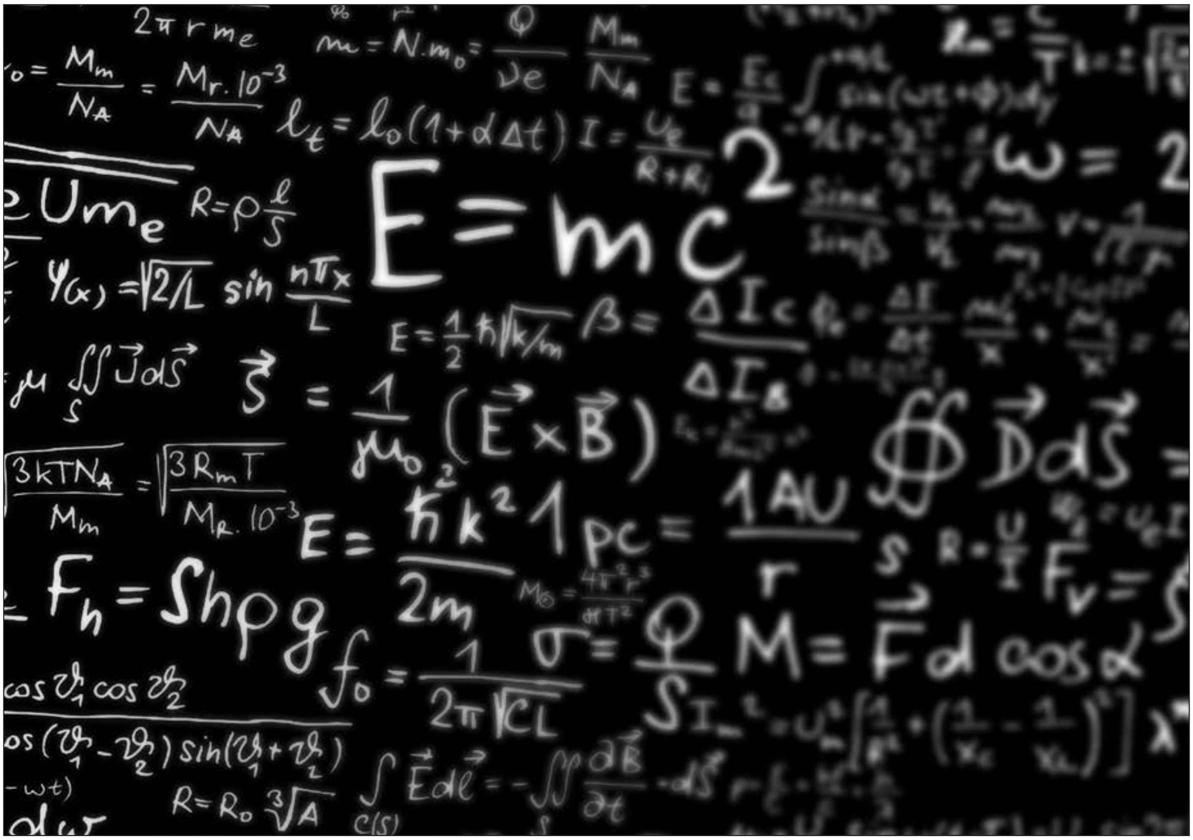

Asian Journal of Mathematics & Statistics

ISSN 1994-5418

science alert

ANSInet
an open access publisher
http://ansinet.com

Asian Journal of Mathematics and Statistics 4 (4): 214-223, 2011
ISSN 1994-5418 / DOI: 10.3923/ajms.2011.214.223

M/M/c Retrial Queueing System with Breakdown and Repair of Services

[1]Muthu Ganapathi Subramanian, [2]Ayyappan and [1]Gopal Sekar
[1]Department of Mathematics, Tagore arts College, Pondicherry, India
[2]Department of Mathematics, Pondicherry Engineering College, Pondicherry, India

Corresponding Author: Muthu Ganapathi Subramanian, Department of Mathematics, Tagore arts College, Pondicherry, India

ABSTRACT

The aim of this research was to study the unreliable nature of the servers under Retrial queueing system. These kinds of models are readily available in real life namely booking in Railway system, Call centres etc., so the analysis of such real models is most important. Consider a multi server retrial queueing system with breakdown and repair of services in which arrival rate follows a Poisson distribution with parameter λ and service time follows an exponential distribution with parameter μ. Let c be the number of servers in the system. The breakdown of service follows an exponential distribution with parameter α and repair of service follows an exponential distribution with parameter β. If any one of the servers is free at the time of a primary call arrival, the arriving call begins to be served immediately by one of the free servers and customer leaves the system after service completion. Otherwise, if c servers are busy or c servers are in breakdown then the 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 finds any one of the servers is free, it is served and leaves the system after service, while the source which produced this repeated call disappears. The access from the orbit to the service facility follows the classical retrial policy. This model is solved by using direct truncation method. Numerical study have been done for analysis of Mean Number of Customers in the Orbit (MNCO), Mean Number of Busy Servers (MNBS), Mean Number of Servers in Breakdown (MNSB) and various system measures.

Key words: Multi server, retrial queues, breakdown and repair of service, direct truncation method, classical retrial policy

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 is called retrial queues. For detailed survey of retrial queues and bibliographical information obtained from Falin (1990), Artalejo (1999a, b, 2010), monograph by Falin and Templeton (1997) and Artalejo and Gomez-Corral (2008). Retrial queues with unreliable servers have been studied by Kulkarni and Choi (1990) and Aissani and Artalejo (1998). 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 available, in this study we will place more emphasis on the solutions by direct t runcation method. The direct truncation method was studied by

Subramanian *et al.* (2009) for retrial queueing system with priority services under negative arrival. Multi server retrial queueing model is studied by Falin and Templeton (1997) and Neuts and Rao (1990) and multiserver retrial queueing with priority services is studied by Ayyappan *et al.* (2010). This study has its own importance in real models like reservation in Railway system, call centre etc., Many computational methods are available to solve Retrial queueing models. In this study, Direct Truncation Method (DTM) is used. This is one of the most reliable methods to find the steady state probability vector. By using this steady state probability vector, the system measures can be obtained. The main objective of this study was to study the unreliable nature of the servers under Retrial queueing system and also to find the system performance measures like mean number of customers in the orbit, Mean number of busy servers etc.,

MODEL DESCRIPTION

Consider a multi server retrial queueing system with breakdown and repair of services in which customers arrive in a Poisson process with arrival rate λ. These customers are identified as primary calls. Further it is assume that the service time follows an exponential distribution with parameter μ. The breakdown of service follows an exponential distribution with parameter α and repair of service follows an exponential distribution with parameter β. Let c be the number of servers in the system. If any one of the server is free at the time of a primary call arrival, the arriving call begins to be served immediately by free server and customer leaves the system after service completion. Otherwise, if c servers are busy or in breakdown then 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 finds any one of the servers is free, it is served and leaves the system after service while the source which produced this repeated call disappears. If there is a breakdown in service for a customer (active breakdown), then the server goes to the state of breakdown and the customer with his incomplete service goes to orbit.

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 of access to the server from the retrial group is called classical retrial policy. The input flow of primary calls, interval between repeated calls, times between breakdown and repair of service are mutually independent.

Let N (t) be the random variable which represents the number of customers in the orbit at time t, H (t) be the random variable which represents the number of servers in breakdown at time t and S (t) represents the number of busy servers at time t.

The random process X is described as:

$$X = \{<N(t), H(t), S(t)>/N(t) = 0, 1, 2, \ldots; H(t) = 0, 1, 2, \ldots, c; S(t) = 0, 1, 2, 3, \ldots, c\}$$

H (t) = 0 and S (t) = 0 means all servers are idle in the system
H (t) = 0 means no servers in breakdown
H (t) = j means j servers are in breakdown
S (t) = i means i servers are busy

The possible state space is:

$$\{(u, v, w)/u = 0, 1, 2, 3, \ldots; v = 0, 1, 2, 3, \ldots, c; w = 0, 1, 2, \ldots, c-v\}$$

The infinitesimal generator matrix Q is given below:

$$
\begin{matrix}
A_{00} & A_0 & O & O & O & \dots \\
A_{10} & A_{11} & A_0 & O & O & \dots \\
O & A_{21} & A_{22} & A_0 & O & \dots \\
O & O & A_{32} & A_{33} & A_0 & \dots \\
\dots & \dots & \dots & \dots & \dots & \dots
\end{matrix}
$$

The matrices described in the Infinitesimal generator matrix Q can be obtained from the following infinitesimal transition rates of process X as follows:

$$
q_{(0,j,k)\,(l,m,n)} = \begin{cases}
\lambda & \text{if } (l, m, n) = (0, j, k+1);\ k = 0, 1, 2,\dots, c-1;\ j = 0 \\
k\mu & \text{if } (l,m, n) = (0, j, k-1);\ k = 1, 2, 3,\dots, c;\ j = 0 \\
\lambda & \text{if } (l,m, n) = (1, j, k);\ k = c;\ j = 0 \\
-(\lambda + k\mu + k\alpha) & \text{if } (l,m, n) = (0, j, k);\ k = 0,1, 2, 3,\dots, c;\ j = 0 \\
k\alpha & \text{if } (l,m, n) = (1, j+1, k-1);\ k = 1, 2, 3,\dots, c;\ j = 0 \\
0 & \text{otherwise}
\end{cases}
$$

$$
q_{(0,j,k)\,(l,m,n)} = \begin{cases}
\lambda & \text{if } (l, m, n) = (0, j, k+1);\ k = 0, 1, 2,\dots,c - j-1;\ j = 1, 2, 3,\dots, c-1 \\
k\mu & \text{if } (l,m, n) = (0, j, k-1);\ k = 1, 2, 3,\dots,c- j;\ j = 1, 2, 3,\dots, c-1 \\
j\beta & \text{if } (l,m, n) = (0, j-1, k);\ k = 0,1, 2, 3,\dots,c- j;\ j = 1, 2, 3,\dots,c-1 \\
\lambda & \text{if } (l, m, n) = (1, j, k);\ k = c -j;\ j = 1, 2, 3,\dots, c-1 \\
-(\lambda + k\mu + k\alpha + j\beta) & \text{if } (l,m, n) = (0, j, k);\ k = 0,1, 2, 3,\dots,c-j;\ j = 1, 2, 3,\dots, c-1 \\
k\alpha & \text{if } (l,m, n) = (1, j+1, k-1);\ k = 1, 2, 3,\dots,c-j;\ j = 1, 2, 3,\dots, c-1 \\
0 & \text{otherwise}
\end{cases}
$$

$$
q_{(0,j,k)\,(l,m,n)} = \begin{cases}
\lambda & \text{if } (l, m, n) = (1, j, 0);\ j = c \\
j\beta & \text{if } (l,m, n) = (0, j-1,0);\ j = c \\
-(\lambda + j\beta) & \text{if } (l,m, n) = (0, j, 0);\ j = c \\
0 & \text{otherwise}
\end{cases}
$$

$$
q_{(i,j,k)\,(l,m,n)} = \begin{cases}
\lambda & \text{if } (l, m, n) = (i, j, k+1);\ k = 0,1, 2,\dots, c-1;\ j = 0;\ i \geq 1 \\
\lambda & \text{if } (l, m, n) = (i+1, j, k);\ k = c;\ j = 0;\ i \geq 1 \\
k\mu & \text{if } (l,m, n) = (i, j, k-1);\ k = 1, 2, 3,\dots,c;\ j = 0;\ i \geq 1 \\
-(\lambda + k\mu + k\alpha) & \text{if } (l,m, n) = (i, j, k)\ ;\ k = 0, 1, 2, 3,\dots, c;\ j = 0;\ i \geq 1 \\
i\sigma & \text{if } (l, m, n) = (i-1, j, k+1);\ k = 0,1,2,3,\dots , c-1;\ j = 0;\ i \geq 1. \\
-i\sigma & \text{if } (l, m, n) = (i, j, k);\ k = 0,1,2,3,\dots , c-1;\ j = 0;\ i \geq 1. \\
k\alpha & \text{if } (l,m, n) = (i+1, j+1, k-1);\ k = 1, 2, 3,\dots,c;\ j = 0;\ i \geq 1 \\
0 & \text{otherwise}
\end{cases}
$$

$$
q_{(i,j,k)\,(l,m,n)} = \begin{cases}
\lambda & \text{if } (l, m, n) = (i, j, k+1);\ k = 0, 1, 2,\dots,c -j-1;\ j = 1, 2, 3,\dots,c-1;\ i \geq 1 \\
\lambda & \text{if } (l, m, n) = (i+1, j, k);\ k = c - j;\ j = 1, 2, 3,\dots,c-1;\ i \geq 1 \\
k\mu & \text{if } (l,m, n) = (i, j, k-1);\ k = 1, 2, 3,\dots,c-j;\ j = 1, 2, 3,\dots,c-1;\ i \geq 1 \\
j\beta & \text{if } (l,m, n) = (i, j-1, k);\ k = 0,1, 2, 3,\dots,c-j;\ j = 1, 2, 3,\dots,c-1;\ i \geq 1 \\
-(\lambda + k\mu + k\alpha + j\beta) & \text{if } (l,m, n) = (i, j, k);\ k = 0, 1, 2, 3,\dots,c-j;\ j = 1, 2, 3,\dots,c-1;\ i \geq 1 \\
i\sigma & \text{if } (l, m, n) = (i-1, j, k+1);\ k = 0,1,2,3,\dots , c-j-1;\ j = 1, 2, 3,\dots,c-1;\ i \geq 1 \\
-i\sigma & \text{if } (l, m, n) = (i, j, k);\ k = 0,1,2,3,\dots , c-j-1;\ j = 1, 2, 3,\dots,c-1;\ i \geq 1 \\
k\alpha & \text{if } (l,m, n) = (i+1, j+1, k-1);\ k = 1, 2, 3,\dots,c-j;\ j = 1, 2, 3,\dots,c-1;\ i \geq 1 \\
0 & \text{otherwise}
\end{cases}
$$

$$q_{(i,j,k)\,(l,m,n)} = \begin{cases} \lambda & \text{if } (l, m, n) = (i+1, j, 0);\ j = c;\ i \geq 1 \\ j\beta & \text{if } (l,m, n) = (i, j-1,0);\ j = c;\ i \geq 1 \\ -(\lambda + j\beta) & \text{if } (l,m, n) = (i, j,0);\ j = c;\ i \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

If the capacity of the origin is finite say M then:

$$q_{(M,j,k)\,(l,m,n)} = \begin{cases} \lambda & \text{if } (l, m, n) = (M, j, k+1);\ k = 0, 1, 2,\ldots, c-1;\ j = 0 \\ k\mu & \text{if } (l,m, n) = (M, j, k-1);\ k = 1, 2, 3,\ldots,c;\ j = 0 \\ -(\lambda + k\mu + M\sigma) & \text{if } (l,m, n) = (M, j, k);\ k = 0,1, 2, 3,\ldots, c-1;\ j = 0 \\ -k\mu & \text{if } (l,m, n) = (M, j, k-1);\ k = c;\ j = 0 \\ M\sigma & \text{if } (l, m, n) = (M-1, j, k+1);\ k = 0,1,2,3,\ldots, c-1;\ j = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$q_{(M,j,k)\,(l,m,n)} = \begin{cases} \lambda & \text{if } (l, m, n) = (M, j, k+1);\ k = 0, 1, 2,\ldots,c-j-1;\ j = 1, 2, 3,\ldots,c-1 \\ k\mu & \text{if } (l,m, n) = (M, j, k-1);\ k = 1, 2, 3,\ldots,c-j;\ j = 1, 2, 3,\ldots,c-1 \\ j\beta & \text{if } (l,m, n) = (M, j-1, k);\ k = 1, 2, 3,\ldots,c-j;\ j = 1, 2, 3,\ldots,c-1 \\ -(\lambda + k\mu + j\beta + M\sigma) & \text{if } (l,m, n) = (M, j, k);\ k = 0, 1, 2, 3,\ldots,c-j-1;\ j = 1, 2, 3,\ldots,c-1 \\ -(k\mu + j\beta) & \text{if } (l,m, n) = (M, j, k);\ k = c-j;\ j = 1, 2, 3,\ldots,c-1 \\ M\sigma & \text{if } (l, m, n) = (M-1, j, k+1);\ k = 0,1,2,3,\ldots, c-j-1;\ j = 1, 2, 3,\ldots,c-1 \\ 0 & \text{otherwise} \end{cases}$$

$$q_{(M,j,k)\,(l,m,n)} = \begin{cases} j\beta & \text{if } (l,m, n) = (M, c-1,0) \\ -j\beta & \text{if } (l,m, n) = (M, c,0) \\ 0 & \text{otherwise} \end{cases}$$

DESCRIPTION OF COMPUTATIONAL PROCEDURES

The above infinitesimal generator matrix is Level Dependent Quasi Birth and Death Process (LDQBD). This type of Retrial queueing models can be solved computationally by one of the following techniques:

- Direct truncation method
- Generalized truncation method
- Truncation method using Level Dependent Quasi Birth and Death Process (LDQBD)
- Matrix geometric approximation

DIRECT TRUNCATION METHOD

In this method, one can truncate the system of equations 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, the only choice available for studying M is through algorithmic methods. While a number of approaches are available for determining the cut-off point, M, the one that seems to perform well is to increase M until the largest individual change in the elements of X for successive values is less than ε a predetermined infinitesimal value.

ANALYSIS OF STEADY STATE PROBABILITIES

Let X be a steady-state probability vector of Q partitioned as X = (x (0), x (1), x (2),...) where X satisfies:

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

where, x (i) = (P_{i00}, P_{i01}, P_{i02}, P_{i03},..., P_{i0c}, P_{i10}, P_{i11}, P_{i12}, P_{i13},..., P_{i1c-1}, P_{i20}, P_{i21}, P_{i22}, P_{i23},..., P_{i2c-2},..., P_{ic-10}, P_{ic-11}, P_{ic0}) for i = 0, 1, 2,...

If M denotes the cut-off point or Truncation level, then the steady state probability vector X (M) is partitioned as X (M) = (x (0), x (1), x (2),..., x (M)), where X (M) satisfies:

$$XM\ QM = 0 \text{ and } XM\ e = 1 \qquad (2)$$

where, x (i) = (P_{i00}, P_{i01}, P_{i02}, P_{i03},..., P_{i0c}, P_{i10}, P_{i11}, P_{i12}, P_{i13},..., P_{i1c-1}, P_{i20}, P_{i21}, P_{i22}, P_{i23},..., P_{i2c-2},..., P_{ic-10}, P_{ic-11}, P_{ic0}) for i = 0, 1, 2,... M

If the capacity of the orbit is finite say M then Eq. 1 becomes 2. The system of Eq. 2 is solved exploiting the special structure of the co-efficient matrix. It is solved by Numerical methods. 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:

$$||xM^*(i) - x^{M^*-1}(i)||\infty < \varepsilon$$

where, ε is an infinitesimal quantity.

STABILITY CONDITION

The necessary and sufficient condition for the system to be stable is:

$$\frac{\lambda}{c\mu}\left(1+\frac{\alpha}{\beta}\right)<1$$

SPECIAL CASES

- Model becomes multi servers retrial queueing system if $\alpha \to 0$
- Model becomes multi servers classical queueing system if $\alpha \to 0$ and $\sigma \to \infty$
- Model becomes single server retrial queueing system with unreliable servers if c = 1
- Model becomes single server retrial queueing system if c = 1 and $\alpha \to 0$

SYSTEM MEASURES

The system measures are used to bring out the qualitative behavior of the queueing model under study. Numerical study has been dealt to find the following measures. The following system measures can be study with steady state probability vectors for various values of λ, μ, σ, α, β and c.

- The probability mass function of number of busy servers

$$\text{Prob (all c servers are busy)} = \sum_{i=0}^{\infty} p(i,0,c)$$

$$\text{Prob (n servers are busy and n<c)} = \sum_{i=0}^{\infty}\sum_{j=0}^{c-n} p(i,j,n)$$

- The probability mass function of number of servers in breakdown

$$\text{Prob (all c servers are in breakdown)} = \sum_{i=0}^{\infty} p(i,c,0)$$

$$\text{Prob (n servers are in breakdown and n<c)} = \sum_{i=0}^{\infty}\sum_{k=0}^{c-n} p(i,n,k)$$

- The probability mass function of number of customers in the orbit

$$\text{Prob (no one in the orbit)} = \sum_{j=0}^{c}\sum_{k=0}^{c-j} p(0,j,k)$$

$$\text{Prob (n customers in the orbit)} = \sum_{j=0}^{c}\sum_{k=0}^{c-j} p(n,j,k)$$

- Mean number of busy servers

$$MNBS = \sum_{n=0}^{\infty} n\left(\sum_{i=0}^{\infty}\sum_{j=0}^{c-n} p(i,j,n) \right)$$

- Mean number of servers in breakdown

$$MNSB = \sum_{n=0}^{c} n\left(\sum_{i=0}^{\infty}\sum_{k=0}^{c-n} p(i,n,k)\right)$$

- Mean number of customers in the orbit

$$MNCO = \sum_{n=0}^{\infty} n \sum_{j=0}^{c}\sum_{k=0}^{c-j} p(n,j,k)$$

- The probability that the orbiting customer is blocked

$$\text{Blocking probability} = \sum_{i=1}^{\infty}\sum_{j=0}^{c} p(i,j,c-j)$$

- The probability that an arriving customer enter into service station immediately

$$=\sum_{i=0}^{\infty}\sum_{j=0}^{c-1}\sum_{k=0}^{c-j-1} p(i,j,k)$$

NUMERICAL STUDY

The stability condition is most important for every queueing system and λ, μ, c, α and β are chosen so that they satisfy the stability condition. System performance measures of this model have been done and expressed in the form of tables which are shown below by finding the steady state probability vector X for various values of λ, μ, σ, c, α and β.

If $\lambda = 30$, $\mu = 10$, $\sigma = 100$, $c = 5$, $\alpha = 10$ and $\beta = 100$, then the steady state probability vector is X = (x [0], x [1], x [2] ,..., x [M]) where:

x [0] = [0.0303, 0.0871, 0.1251, 0.1196, 0.0841, 0.0408, 0.0004, 0.0049, 0.0124, 0.0165, 0.0130, 0.0000, 0.0001, 0.0005, 0.0009, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000]
x [1] = [0.0045, 0.0131, 0.0199, 0.0232, 0.0278, 0.0357, 0.0045, 0.0136, 0.0207, 0.0223, 0.0228, 0.0002, 0.0011, 0.0025, 0.0037, 0.0000, 0.0000, 0.0001, 0.0000, 0.0000, 0.0000]
x [2] = [0.0003, 0.0011, 0.0024, 0.0054, 0.0129, 0.0269, 0.0007, 0.0021, 0.0042, 0.0084, 0.0190, 0.0004, 0.0012, 0.0022, 0.0048, 0.0000, 0.0001, 0.0004, 0.0000, 0.0000, 0.0000]
x [3] = [0.0000, 0.0001, 0.0005, 0.0020, 0.0068, 0.0193, 0.0001, 0.0003, 0.0011, 0.0042, 0.0142, 0.0001, 0.0002, 0.0010, 0.0039, 0.0000, 0.0001, 0.0005, 0.0000, 0.0000, 0.0000]
x [4] = [0.0000, 0.0000, 0.0002, 0.0009, 0.0038, 0.0136, 0.0000, 0.0001, 0.0004, 0.0023, 0.0101, 0.0000, 0.0001, 0.0005, 0.0029, 0.0000, 0.0001, 0.0004, 0.0000, 0.0000, 0.0000]
x [5] = [0.0000, 0.0000, 0.0001, 0.0004, 0.0021, 0.0094, 0.0000, 0.0000, 0.0002, 0.0013, 0.0071, 0.0000, 0.0000, 0.0003, 0.0021, 0.0000, 0.0000, 0.0003, 0.0000, 0.0000, 0.0000]

Similarly, we can find x (n) for $n \geq 6$ and it is noticed that x (n)$\rightarrow$0 as $n\rightarrow\infty$. For the numerical parameters chosen above, x (n)$\rightarrow$0 for $n \geq 12$ and the sum of the steady state probabilities becomes 0.9999999999. In the same manner, we can find the steady state probability vector X for all values λ, μ, σ, c, α and β.

SYSTEM PERFORMANCE MEASURES

The system performance measures are most important for every queueing system. These measures are calculated numerically by using the steady state probability vectors x [0], x [1], x [2],... and using formulas described under system measures and presented in Table 1 to 6.

Table 1 represents the for probability mass function of number of busy servers in the system for $\lambda = 30$, $\mu = 10$, $\sigma = 100$, $c = 5$, $\alpha = 10$ and $\beta = 100$. From this table we can find the Mean number of busy servers.

Mean No. of busy servers = 2.999999

Table 2 represents the for probability mass function of number of servers in breakdown for $\lambda = 30$, $\mu = 10$, $\sigma = 100$, $c = 5$, $\alpha = 10$ and $\beta = 100$. From this table we can find the Mean number of servers in breakdown.

Mean No. of servers in breakdown = 0.300000

Table 3 represents the for probability mass function of number of customers in the orbit $\lambda = 30$, $\mu = 10$, $\sigma = 100$, $c = 5$, $\alpha = 10$ and $\beta = 100$. From this table we can find the Mean number of customers in the orbit.

Table 1: Number of busy servers with probabilities

No. of busy servers	0	1	2	3	4	5
Probability	0.041364	0.125272	0.196414	0.230919	0.241011	0.165020

Table 2: Number of servers in vacation with probabilities

No. of servers in breakdown	0	1	2	3	4	5
Probability	0.739819	0.223503	0.033672	0.002872	0.000131	0.000002

Table 3: Number of customers in the orbit with probabilities

No. of customers in the orbit	Probability	No. of customers in the orbit	Probability	No. of customers in the orbit	Probability
0	0.533475	7	0.010375	14	0.000615
1	0.214937	8	0.006932	15	0.000410
2	0.092140	9	0.004633	16	0.000274
3	0.054155	10	0.003096	17	0.000182
4	0.035186	11	0.002068	18	0.000182
5	0.023318	12	0.001381	19	0.000081
6	0.023318	13	0.000922	20	0.000054

Table 4: Breakdown rate (α) and c against mean number of customers in the orbit for λ =8, μ = 10, β = 100 and σ = 100

	Mean No. of customers in the orbit						
α	c = 1	c = 2	c = 3	c = 4	c = 5	MNBS	MNSB
10.0000	7.8399	0.3855	0.1326	0.0902	0.0819	0.8000	0.0800
5.0000	5.1400	0.2767	0.0768	0.0462	0.0410	0.8000	0.0400
2.5000	4.2400	0.2251	0.0497	0.0246	0.0207	0.8000	0.0200
1.2500	3.8610	0.2000	0.0364	0.0138	0.0105	0.8000	0.0100
0.6250	3.6861	0.1876	0.0297	0.0085	0.0055	0.8000	0.0050
0.3125	3.6020	0.1814	0.0264	0.0058	0.0029	0.8000	0.0025
0.1563	3.5607	0.1784	0.0248	0.0045	0.0017	0.8000	0.0012
0.0781	3.5403	0.1768	0.0240	0.0038	0.0010	0.8000	0.0006
0.0391	3.5301	0.1761	0.0235	0.0035	0.0007	0.8000	0.0003
0.0195	3.5250	0.1757	0.0233	0.0033	0.0005	0.8000	0.0002
0.0098	3.5225	0.1755	0.0232	0.0032	0.0005	0.8000	0.0001
0.0049	3.5212	0.1754	0.0232	0.0032	0.0004	0.8000	0.0000

c: Number of servers, α: Rate of breakdown, MNBS: Mean number of busy servers, MNSB: Mean number of servers in breakdown

Mean number of customers in the orbit = 1.193782

Table 4 shows the impact of α and c over Mean number of customers in the orbit. Further, the following information can be obtained:

- Mean number of customers in the orbit decreases as α decreases
- Mean number of customers in the orbit decreases as number of servers increases
- Mean Number of Busy Servers (MNBS) is independent of α
- This model becomes multi server retrial queueing system if $\alpha \to 0$

Table 5 shows the impact of σ and c over Mean number of customers in the orbit. Further, the following information can be obtained:

- Mean number of customers in the orbit decreases as retrial rate σ increases

Table 5: Retrial rate (σ) and c over mean number of customers in the orbit for λ =30, μ = 40, α = 10 and β = 100

	Mean No. of customers in the orbit						
σ	c = 1	c = 2	c =3	c = 4	c = 5	MNBS	MNSB
10	22.5143	1.9724	0.9677	0.7901	0.7566	0.7500	0.0750
30	10.2356	0.8039	0.3475	0.2681	0.2531	0.7500	0.0750
50	7.7785	0.5695	0.2234	0.1637	0.1524	0.7500	0.0750
70	6.7255	0.4687	0.1701	0.1190	0.1092	0.7500	0.0750
90	6.1404	0.4124	0.1404	0.0941	0.0853	0.7500	0.0750
100	5.9357	0.3927	0.1300	0.0853	0.0769	0.7500	0.0750
300	4.7071	0.2726	0.0670	0.0329	0.0265	0.7500	0.0750
500	4.4614	0.2480	0.0541	0.0223	0.0163	0.7500	0.0750
600	4.4000	0.2417	0.0509	0.0197	0.0138	0.7500	0.0750
700	4.3561	0.2373	0.0486	0.0178	0.0120	0.7500	0.0750
800	4.3232	0.2339	0.0468	0.0164	0.0107	0.7500	0.0750
900	4.2976	0.2313	0.0455	0.0152	0.0096	0.7500	0.0750
1000	4.2771	0.2292	0.0444	0.0144	0.0088	0.7500	0.0750
2000	4.1850	0.2197	0.0394	0.0104	0.0050	0.7500	0.0750
3000	4.1542	0.2165	0.0378	0.0090	0.0037	0.7500	0.0750
4000	4.1389	0.2149	0.0369	0.0083	0.0031	0.7500	0.0750
5000	4.1297	0.2139	0.0364	0.0079	0.0027	0.7500	0.0750
6000	4.1235	0.2133	0.0361	0.0077	0.0024	0.7500	0.0750

c: No. of servers, σ: Retrial rate, MNBS: Mean number of busy servers, MNSB: Mean number of servers in breakdown

Table 6: Repair of service rate (β) and c against mean number of customers in the orbit for λ = 8, μ = 10, α = 10 and σ = 100

	Mean number of customers in the orbit						
β	c = 1	c = 2	c = 3	c = 4	c = 5	MNBS	MNSB
100	7.8399	0.3855	0.1326	0.0902	0.0819	0.8000	0.0800
300	4.8164	0.3088	0.1141	0.0854	0.0808	0.8000	0.0267
500	4.4257	0.2948	0.1108	0.0846	0.0807	0.8000	0.0160
700	4.2722	0.2890	0.1095	0.0843	0.0806	0.8000	0.0114
900	4.1901	0.2858	0.1088	0.0842	0.0806	0.8000	0.0089
1000	4.1620	0.2847	0.1085	0.0841	0.0806	0.8000	0.0080
2000	4.0384	0.2798	0.1074	0.0839	0.0805	0.8000	0.0040
3000	3.9984	0.2781	0.1071	0.0838	0.0805	0.8000	0.0027
4000	3.9786	0.2773	0.1069	0.0838	0.0805	0.8000	0.0020
5000	3.9667	0.2768	0.1068	0.0837	0.0805	0.8000	0.0016
6000	3.9589	0.2765	0.1067	0.0837	0.0805	0.8000	0.0013
7000	3.9533	0.2763	0.1067	0.0837	0.0805	0.8000	0.0011
8000	3.9491	0.2761	0.1066	0.0837	0.0805	0.8000	0.0010
9000	3.9459	0.2760	0.1066	0.0837	0.0805	0.8000	0.0009
10000	3.9433	0.2759	0.1066	0.0837	0.0805	0.8000	0.0008

c: Number of servers, β: Repair rate, MNBS: Mean number of busy servers, MNSB: Mean number of servers in breakdown

- Mean number of customers in the orbit decreases as number of servers increases
- Mean Number of Busy Servers (MNBS) and Mean Number of Servers in Breakdowns (MNSB) are independent of retrial rate σ
- Model becomes Multi server classical queueing system with unreliable servers if $\sigma \to \infty$

Table 6 shows the impact of **β** and c over Mean number of customers in the orbit. Further, the following information can be obtained:

- Mean number of customers in the orbit decreases as β increases
- Mean number of customers in the orbit decreases as number of servers increases
- Mean Number of Busy Servers (MNBS) is independent of β
- This model becomes Multi server retrial queueing system if β is large

CONCLUSION

It is observed from the numerical study that means number of customers in the orbit decreases as the retrial rate increases and means number of busy servers and mean number of breakdown independent of retrial rate σ. The various cases which have been discussed under special cases are particular cases of this study. This study can be further extended by introducing various concepts like second optional service, loss and feedback and vacation policies etc.

REFERENCES

Aissani, A. and J.R. Artalejo, 1998. On the single server retrial queue subject to breakdown. Queueing Sys., 30: 309-321.

Artalejo, J.R. and A. Gomez-Corral, 2008. Retrial Queueing Systems: A Computational Approach. Springer, Spain, pp: 318.

Artalejo, J.R., 1999a. A classified bibliography of research on retrial queues: Progress in 1990-1999. Busin. Econ., 7: 187-211.

Artalejo, J.R., 1999b. Accessible bibliography on retrial queues. Math. Comp. Mod., 30: 1-6.

Artalejo, J.R., 2010. Accessible bibliography on retrial queues: Progress in 2000-2009. Math. Comp. Mod., 51: 1071-1081.

Ayyappan, M., G. Subramanian and G. Sekar, 2010. Multi server retrial queueing with priority service by Matrix Geometric Method. Neural Parallel Scientific Comp. Dyn. Pub., 4: 43-47.

Falin, G., 1990. A survey of retrial queues. Queueing Sys., 7: 127-167.

Falin, G.I. and J.G.C. Templeton, 1997. Retrial Queues. Champman and Hall, London, pp: 328.

Kulkarni, V.G. and B.D. Choi, 1990. Retrial queues with server subject to breakdown and repairs. Queueing sys., 7: 191-208.

Neuts, M.F. and B.M. Rao, 1990. Numerical investigation of multi server retrial model. Queueing sys., 7: 169-189.

Subramanian, M.G., G. Ayyappan and G. Sekar, 2009. Single server retrial queueing system with negative arrival under Non-Pre emptive priority service. J. fund. Sci., 5: 130-146.