Queueing Models and
Service Management
Vol. 1, No. 1, page 031-074, 2018

Analysis of Multiprogramming-Multiprocessor Retrial Queueing Systems with Bernoulli Vacation Schedule

B. Krishna Kumar[1*], R. Navaneetha Krishnan[1], R. Sankar[1] and R. Rukmani[2]

[1]Department of Mathematics
Anna University
Chennai 600 025, India

[2]Department of Mathematics
Pachaiyappa's College
Chennai 600 030, India.

(Received March 2017 ; accepted October 2017)

Abstract: We study a Markovian finite orbit capacity retrial queue of a multiprogramming-multiprocessor computer network system in which the service channels avail vacation under Bernoulli schedule. For this system, the steady-state probabilities of the number of programs in the system and the mean number of programs in the orbit along with other descriptors of the system are obtained by adopting the matrix analytic methods. Moreover, the expressions for the Laplace-Stieltjes transforms of the busy period of the system and the waiting time of a program in the orbit are determined. The probability generating function for the number of retrials made by a program is also derived. Some key performance measures of the system and various moments of quantities of interest are discussed. Finally, extensive numerical results are carried out to reveal the influence of the system parameters on the system performance measures.

Keywords: Busy period, first-step principle, number of retrials, retrial queue, steady-state distribution, vacation, waiting time.

1. Introduction

In recent years, retrial queueing systems have been investigated by many researchers because of their broad practical applicability in the performance analysis of auto-repeat facilities in telecommunication networks, computer systems with retransmissions, packet switching networks, shared bus local area networks, optical network systems, etc., Retrial queueing systems are characterized by the fact that customers (or calls, packets) who find all servers busy upon arrival are obliged to leave the service area and repeat their demand after some random time, called "retrial time". Between retrials, the blocked customers remain in a retrial group, called 'orbit'. The literature on retrial queueing systems is very rich. For the reviews of the main results and applications of retrial queueing systems, the readers are referred to the monographs by Falin and Templeton [16] and Artalejo and Gomez-Corral [5]. For more references on this, see the bibliographical overviews in Falin [14], Kulkarni and Liang [29] and Artalejo [4].

Study of multiprogramming computer networks for performance analysis has received wide attention in the computer and communication systems. Multiprogramming computer networks with main memory incorporate both autonomous peripheral devices, central processing unit (CPU) capable of operating concurrently with input-output (I/O) devices and multiple programs (packets). The system admits several programs to reside simultaneously in the main memory (service area) of the computer, so that when one program requests an I/O operation, the CPU can be switched immediately to another waiting program for carrying out the computation. Typically, a program's execution in the system consists of a sequence of I/O and CPU activities until its request for work is completed. After that, the

* Corresponding author
Email : drbkkumarau@gmail.com ; drbkkumar@hotmail.com

program relinquishes its occupied main memory when it leaves the system and a new program immediately enters the system, if any, and joins the queue of I/O device. In other words, once the program enters into the main memory, it joins the I/O queues for input activities. In due course, after completion of its input activities, the program may join the CPU queue for processing activities. Soon after , the completion of a CPU service, the program either departs from the system (i.e., when its requirement has been completed) or continues cycling from CPU to I/O and vice versa until the completion of the requirement. While the system runs in a multiprogramming mode, several programs contend simultaneously for the resources, thereby creating queues of requests. For the reasons of efficiency, the number of programs admitted, by request, to use the resources in the main memory is kept fixed to a maximum level, say M, which is called the maximum degree of multiprogramming.

There is a great deal of literature devoted to multiprogramming computer networks. For the systematic developments of the mathematical descriptions and applications on this topic, one can refer to the research articles by Gaver [17], Lewis and Shedler [32], Gaver and Shedler [20], Adiri et al. [1], Avi-Itzhak and Heyman [9], Hofri [22], Konheim and Reiser [26], Daduna [11], Kameda [23], Avi-Itzhak and Halfin [7, 8], Latouche [30], Rege and Sengupta [33] and the monographs by Allen [2] and Gelenbe and Mitrani [21].

Several queueing protocols have been implemented in order to obtain the various significant results for the system described above either under heavy traffic/loaded situations, i.e., there is a buffer of finite/infinite capacity in front of a main memory (service area) which is never empty (see Gaver [17], Adiri et al. [1], Konheim and Reiser [25], Kameda [23], Gelenbe and Mitrani [21] or the closed queueing networks which consist of a fixed number of programs in the main memory and a cycle queue of multiple I/O and CPU service channels (see Lewis and Shedler [32], Gaver and Humfeld [18], Brandwajn [10], Daduna [11] and Gelenbe and Mitrani [21]).

In a recent paper, Krishna Kumar et al. [28] have dealt with the multiprogramming - multiprocessor computer network retrial queueing system with constant retrial policy for infinite/finite orbit capacity. For such repeated request system, the level-independent quasi-birth-death (LIQBD) process of three-dimensional Markov process is constructed. By employing the matrix geometric/analytic methods, the stationary distribution and various performance measures such as busy period, waiting time, etc., of the system have been obtained. Although several queueing models have been extensively investigated in the literature for multiprogramming-multiprocessor computer networks, to the best of our knowledge, the classical retrial policy for the multiprogramming-multiprocessor computer network with the Bernoulli vacation schedule of CPU service channels is not studied. In practical situations, allowing the service channels to take the Bernoulli vacations makes the system more realistic and flexible in studying the multiprogramming-multiprocessor computer networks with maintenance activities. In addition, repeated requests of the programs (packets) have quite an impact on the performance of vacation systems and hence cannot be neglected in computer network designs and planning.

This motivates us to study a novel computer network system with classical retrial policy as well as with Bernoulli vacation schedules of the service channels. Thus, the proposed queueing system makes the anlysis more complex and challenging. In what follows, after formulating the level-dependent quasi-birth-death (LDQBD) process of four-dimensional Markov process, where level is referred to as the number of programs in the finite retrial orbit, we derive the stationary distribution and performance measures for the various vacation systems by adopting the matrix analytical techniques. Besides, we explore both analytical and numerical results for the system which are new and different from the results of the paper by Krishna Kumar et al. [28].

The retrial queueing systems with Bernoulli vacation schedule for the multiprogramming-multiprocessor computer network systems occur in many real life situations where the service channels may be used for other tasks, maintenance activities, etc. For instance, the proposed system can be applied to analyse the performance of a host-processor which consists of several central processing units (CPUs), a main memory, several channels and disk files. The disk sub-system has the feature of rotational position sensing (RPS), by which the channels and storage controls are allowed to be released during most of the record search time, thus increasing channels and control units availability for other operations. The processors themselves are operated in a multiprogramming environment and are assumed to have more than one program waiting for processing. For an efficient use of the limited resources of the host-processor system, the maximum number of programs allowed to reside in the computer network is bounded by some number, called the degree of multiprogramming. The CPU service channel makes input/output (I/O) requests whenever the program being processed issues the I/O commands (such as read, write, etc.) either for data or for any information not available in the main memory. Having initiated the requests, the CPU starts to process the next program waiting at the CPU queue. At the same time, the requested I/O operators are performed independent of the CPUs. When the desired informations are transferred into the main memory, the program that made such requests will return to the CPU queue for another CPU processing. This process is repeated until all the required processing is completed for a particular program. Upon completion of its final request, the program leaves the system.

Apart from the programs execution in the CPU and I/O queueing systems of the multiprogramming-multiprocessor computer networks, there are many circumstances under which the normal flow of a program in the computer system is interrupted. The interruption causes the temporary suspension of the program in progress. Hence the network controller requests the CPUs to take necessary action for the special events. A major application for interruptions is to use interrupt as a method of allowing the CPUs' time at such periods to different programs or threads that are sharing the CPUs. (The threads are small pieces of a program that can be executed independently). As the CPU can execute only one sequence of instructions at a time, the ability to time share multiple programs or threads implies that the computer network system must share the CPUs by allowing small segments of time to each program or thread, in rotation among them. Further, each program sequence is allowed to execute some instructions. After a certain period of time, that sequence is interrupted and relinquishes control to a dispatcher program within the operating system that allocates the next block of time to another program sequence.

The computer network system provides an internal clock that sends an interrupt periodically to the CPUs. The time between interrupt pulses is known as a quantum which represents the time that is allotted to each program or thread. When the clock interrupt occurs, the interrupt routine returns control to the operating system, which then determines, which program or thread will receive the CPUs' time next. The interrupt is a simple but effective method for allocating the operating system to share CPU resources among several programs at the same time.

As mentioned earlier, in the host-processor, programs have to be routed and pass through a sequence of links and nodes. Programs from different sources are time-multiplexed and thus flow sequentially through the network links. When programs arrive, they need to be queued in a buffer and have to wait before they can be forwarded to the I/O and CPU queues service area and travel through the computer network. This store-and-forward procedure can cause a serious increase of the latency and congestion of programs. In order to avoid the increasing number of programs and network complexity, the buffering processes need to be designed as efficiently as possible. This can be achieved when the program which finds the I/O and CPU queues service area is fully utilized upon arrival, will leave the service area and re-initiate its attempt for execution of the I/O and CPU service activities of the computer network after a random time. This process is repeated until the blocked program enters into the service area of the computer network system.

The rest of the paper is organized as follows: In section 2, we describe briefly the mathematical model and formulation of the retrial queueing network system under consideration. The steady-state probabilities of the number of programs in the retrial group and the status of the service channels in the main memory for finite orbit capacity (FOC) system are reported in Section 3. The measures of interest are obtained in Section 4 for the retrial queueing network system under discussion. Section 5 deals with some specific probabilistic descriptors such as ideal and vain retrials of the FOC retrial queueing network system. Moreover, the optimal rate of retrial has been investigated numerically. The computational analysis for the length of a busy period of the system is studied in Section 6. In Sections

7 and 8, we analyze the waiting time of a tagged program and the number of retrials made by a tagged program before entering into the main memory (service area), respectively. Extensive numerical results have been carried out on the main performance measures against various system parameters to corroborate our theoretical analysis. Finally, in Section 9, we conclude our results and present some future directions.

2. Model Description

We consider a multiprogramming-multiprocessor computer retrial queueing network system with finite orbit capacity (FOC), say, K, and inner multiprocessor network (main memory). The structure of the system is depicted in Figure 1. The inner multiprocessor network consists of J_1 identical input-output (I/O) service channels with buffer and J_2 identical central processing (CPU) service channels with buffer. Each CPU service channel takes a Bernoulli vacation as described by Keilson and Servi [24]. That is, upon the completion of a service, if there are programs in the buffer of the CPU queue, either the service of the next program begins with probability $(1-p)$, $(0 \le p \le 1)$, or a vacation period begins for a duration with complementary probability p. On the other hand, if there is no program left in the buffer of the CPU queue after a service completion, the service channel always takes a vacation period for a duration. At the end of a vacation period, the service commences, if there is a program present in the buffer of the CPU queue. Otherwise, the service channel takes another vacation immediately and continues in the same manner until it finds at least one program waiting in the buffer upon returning from a vacation (i.e., multiple vacations). This process holds good for all J_2 service channels of the CPU queue. The length of vacation periods $\{V_k; k = 1,2,3,\dots\}$ of service channels of the CPU queue are independent and identically distributed (i.i.d) exponential random variables with parameter θ.

The important merit of the Bernoulli vacation scheduling is the existence of a control parameter p. By adjusting the value of p, we can control the congestion of the system. Besides, when $p = 0$, the Bernoulli vacation scheduling service discipline is equivalent to the exhaustive service discipline and when $p = 1$ to the 1-limited service discipline.

When the system executes in a multiprogramming mode, several programs (packets) contend simultaneously for the resources, thereby creating queues of requests in the inner multiprocessor network. For reasons of efficiency, the maximum length of I/O queue and CPU queue (including both buffers and service channels) is assumed to be a fixed number, say, M, i.e., a maximum of M programs are allowed to occupy the inner multiprocessor network.

Primary programs (those that arrive at the system for the first time) arrive at the multiprocessor retrial queueing network system according to a Poisson process with rate λ. If the number of programs in the inner multiprocessor network is less than M, an arriving primary program joins the queue in front of J_1 I/O service channels having exponentially distributed service times with rate μ_1. Soon after the completion of an I/O operation, the program either joins the queue in front of J_2 CPU service channels with probability p_1, $(0 \le p_1 < 1)$, or leaves the system forever with the complementary probability $1 - p_1$. In the former case, the programs, that have joined the queue in front of J_2 CPU service channels, will be served with exponentially distributed service times of rate μ_2. After completion of a CPU operation, the program either joins the I/O queue again for service with probability p_2, $(0 \le p_2 < 1)$, or leaves the system forever with the complementary probability $1 - p_2$. In each queue, the programs are served by the FCFS discipline. On the other hand, if an arriving primary program finds the inner multiprocessor network fully occupied by M programs, it is forced to join the orbit/retrial group of capacity K from where the retrial programs can repeat (independent of each other) their requests an exponentially distributed time period with intensity equals to $n\nu$ where n, $(1 \le n \le K)$, is the number of programs present in the orbit and ν is the rate of retrial. The arriving primary program will lost forever if the main memory and the orbit group are fully occupied, respectively, by M and K programs. Clearly, the orbital programs compete with potential primary programs to decide which program will enter next into the inner multiprocessor network. Upon retrial from the orbit, if the

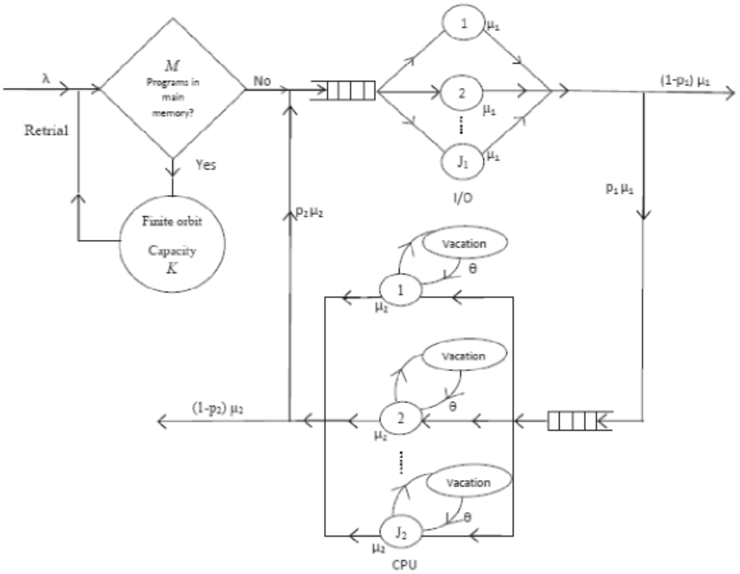

Figure 1. Retrial Queueing System for Multiprocessor- Multiprogramming Computer Network

program finds M programs in the inner multiprocessor network, then it always rejoins the orbit. This process continues until it eventually enters into the inner multiprocessor network. The behavior of a retrial program is the same as that of the primary program. The inter-arrival times of primary programs, interval of successive retrials, vacation times and service times are assumed to be mutually independent.

Based on the assumption of Poisson arrivals and exponential distributions for services, vacations and retrials, the system may be described as a continuous-time Markov chain (CTMC) model. The state of the queueing system under investigation can be described by means of a four-dimensional Markov process $\{X(t);\ t \geq 0\} = \{(N(t), I(t), C(t), S(t));\ t \geq 0\}$ where $N(t)$ is the number of programs in the orbit (source of repeated demands), $I(t)$ is the number of programs in the I/O queue with J_1 service channels, $C(t)$ is the number of programs in the CPU queue with J_2 service channels and $S(t)$ is the number of busy/active service channels in the CPU queue at time t. In the sequel, without loss of generality, we assume that $J_1 + J_2 \leq M$ in order to avoid several different cases. Thus, the process $\{X(t);\ t \geq 0\}$ forms an irreducible regular CTMC on the state space $\Omega = \{(n, m-l, l, j);\ 0 \leq n \leq K,\ 0 \leq j \leq s_l,\ 0 \leq l \leq m,\ 0 \leq m \leq M\}$ where $s_m = min(J_2, m),\ m = 0, 1, 2, \dots, M,$ and the 4-tuple $(n, m-l, l, j)$ represents that 'n'programs are in the retrial orbit, '$m-l$' programs are in the I/O queue, 'l'programs are in the CPU queue and 'j'service channels are busy in the CPU queue at an arbitrary time. We further define the level $\underline{\boldsymbol{n}}$, $\underline{\boldsymbol{n}} = \underline{\boldsymbol{0}}, \underline{\boldsymbol{1}}, \underline{\boldsymbol{2}}, \dots, \underline{\boldsymbol{K}}$, as the set of states $\underline{\boldsymbol{n}} = \{(n, 0,0,0), (n, 1,0,0), (n, 0,1,0), (n, 0,1,1), \dots, (n, 0, M, 1), \dots, (n, 0, M, s_M)\}$. The queueing system under discussion can be treated as a Markov model of level-dependent quasi-birth-death (LDQBD) process with the state space Ω and hence the technique of the classical matrix analytical method can be employed to obtain the stationary distribution of the system size.

3. Stationary Distribution

The elements of the set of states in Ω can be enumerated in the lexicographical order. Using elementary argument, the finite state Markov chain for the system under study has a transition rate matrix $\boldsymbol{Q}$ which has a block tri-diagonal structure given as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A}_{1,0} & \mathbf{A}_{0,0} & & & & & & \\ \mathbf{A}_{2,1} & \mathbf{A}_{1,1} & \mathbf{A}_{0,1} & & & & & \\ & \mathbf{A}_{2,2} & \mathbf{A}_{1,2} & \mathbf{A}_{0,2} & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & \mathbf{A}_{2,n} & \mathbf{A}_{1,n} & \mathbf{A}_{0,n} & & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & \mathbf{A}_{2,K-1} & \mathbf{A}_{1,K-1} & \mathbf{A}_{0,K-1} \\ & & & & & & \mathbf{A}_{2,K} & \mathbf{A}_{0,K} + \mathbf{A}_{1,K} \end{bmatrix}. \quad (1)$$

Matrix $\mathbf{Q}$ is an infinitesimal generator of a continuous time Markov chain (CTMC) which is a level-dependent quasi-birth-and-death (LDQBD) process (see Latouche and Ramaswami [31]). The dimension of the square matrix, $\boldsymbol{Q}$, is $\Lambda_M = (K+1)\Gamma_M$ where $\Gamma_M = \sum_{m=0}^{M} \gamma_m$ with

$$\gamma_m \;=\; \left[\frac{(s_m+1)(s_m+2)}{2} + (m - s_m)(s_m+1)\delta_{s_m J_2}\right],$$

in which $s_m = min(J_2, m)$, $m = 0,1,2,\dots,M$, and the Kronecker's delta function δ_{ij} is defined as

$$\delta_{ij} = \begin{cases} 1 & \text{if} \quad i = j, \\ 0 & \text{if} \quad i \neq j. \end{cases}$$

The non-zero blocks, $\boldsymbol{A}_{1,n}$, $0 \le n \le K$, $A_{2,n}$, $1 \le n \le K$, $\boldsymbol{A}_{0,n}$, $0 \le n \le K$ and the zero blocks, $\mathbf{0}$, are all square matrices of order Γ_M. Each of these blocks is itself highly structured. Specifically, the main diagonal blocks $\boldsymbol{A}_{1,n}$, $n = 0,1,2,\dots,K$, appearing in $\boldsymbol{Q}$ are specified as follows:

The boundary block $\mathbf{A}_{1,0}$ has a partitioned block tri-diagonal structure given as:

$$\mathbf{A}_{1,0} = \begin{bmatrix} \mathbf{A}_{1,0}^{1,0} & \mathbf{A}_{1,0}^{0,0} & & & & & & \\ \mathbf{A}_{1,0}^{2,1} & \mathbf{A}_{1,0}^{1,1} & \mathbf{A}_{1,0}^{0,1} & & & & & \\ & \mathbf{A}_{1,0}^{2,2} & \mathbf{A}_{1,0}^{1,2} & \mathbf{A}_{1,0}^{0,2} & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & \mathbf{A}_{1,0}^{2,m} & \mathbf{A}_{1,0}^{1,m} & \mathbf{A}_{1,0}^{0,m} & & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & \mathbf{A}_{1,0}^{2,M-1} & \mathbf{A}_{1,0}^{1,M-1} & \mathbf{A}_{1,0}^{0,M-1} \\ & & & & & & \mathbf{A}_{1,0}^{2,M} & \mathbf{A}_{1,0}^{1,M} \end{bmatrix}_{\Gamma_M \times \Gamma_M} \quad (2)$$

where, in the above, the main diagonal blocks $\mathbf{A}_{1,0}^{1,m}$, $0 \le m \le M$, are square matrices represented by

$$\mathbf{A}_{1,0}^{1,m}=\begin{bmatrix}\mathbf{B}_{r_m,s_0} & \mathbf{D}_{r_m} & & & & & & \\ \mathbf{E}_{s_1} & \mathbf{B}_{r_{m-1},s_1} & \mathbf{D}_{r_{m-1}} & & & & & \\ & \mathbf{E}_{s_2} & \mathbf{B}_{r_{m-2},s_2} & \mathbf{D}_{r_{m-2}} & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & \mathbf{E}_{s_l} & \mathbf{B}_{r_{m-l},s_l} & \mathbf{D}_{r_{m-l}} & & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & \mathbf{E}_{s_{m-1}} & \mathbf{B}_{r_1,s_{m-1}} & \mathbf{D}_{r_1} \\ & & & & & & \mathbf{E}_{s_m} & \mathbf{B}_{r_0,s_m}\end{bmatrix}_{\gamma_m\times\gamma_m}$$

in which the diagonal sub-blocks $\mathbf{B}_{r_{m-l},s_l}$, $0\le l\le m$, are square matrices expressed as

$$\mathbf{B}_{r_{m-l},s_l}=\begin{bmatrix}-(\lambda+r_{m-l}\mu_1+(1-\delta_{l0})J_2\theta) & J_2\theta & & & \\ & -(\lambda+r_{m-l}\mu_1+\mu_2+(J_2-1)\theta) & (J_2-1)\theta & & \\ & \ddots & \ddots & & \\ & & & -(\lambda+r_{m-l}\mu_1+(s_l-1)\mu_2+(J_2-s_l+1)\theta) & (J_2-s_l+1)\theta \\ & & & & -(\lambda+r_{m-l}\mu_1+s_l\mu_2\end{bmatrix}_{(s_l+1)\times(s_l+1)}$$

with $r_l=min(J_1,\ l)$, $0\le l\le M$, the lower diagonal sub-block matrices, $\boldsymbol{E}_{s_l}$, $1\le l\le m$, are given as

$$\mathbf{E}_{s_l}=\begin{cases}\begin{bmatrix}0 & & & & \\ pp_2\mu_2 & (1-p)p_2\mu_2 & & & \\ & 2pp_2\mu_2 & 2(1-p)p_2\mu_2 & & \\ & & \ddots & \ddots & \\ & & & (s_l-1)pp_2\mu_2 & (s_l-1)(1-p)p_2\mu_2 \\ & & & & s_lp_2\mu_2\end{bmatrix}_{(s_l+1)\times(s_{l-1}+1),} & \text{for}\quad l=1,2,3,\dots,J_2, \\ \begin{bmatrix}0 & & & & \\ pp_2\mu_2 & (1-p)p_2\mu_2 & & & \\ & 2pp_2\mu_2 & 2(1-p)p_2\mu_2 & & \\ & & \ddots & \ddots & \\ & & & (s_l-1)pp_2\mu_2 & (s_l-1)(1-p)p_2\mu_2 & \\ & & & & s_lpp_2\mu_2 & s_l(1-p)p_2\mu_2\end{bmatrix}_{(s_l+1)\times(s_l+1)} & \text{for}\quad l=J_2+1,J_2+2,\dots,m,\end{cases}$$

and the upper diagonal blocks $\mathbf{D}_{r_{m-l}}$, $0\le l\le m$, are described as

$$\mathbf{D}_{r_{m-l}}=\begin{cases}\left[r_{m-l}p_1\mu_1\boldsymbol{I}_{s_l+1}\ 0_{((s_l+1)\times1)}\right]_{(s_l+1)\times(s_{l+1}+1)}, & l=0,1,2,\dots,\mathrm{J}_2-1, \\ \left[r_{m-l}p_1\mu_1\boldsymbol{I}_{s_l+1}\right]_{(s_l+1)\times(s_l+1)}\quad, & l=\mathrm{J}_2,\ \mathrm{J}_2+1,\dots,\mathrm{m}-1,\end{cases}$$

where $\mathbf{I}_n$, in general, denotes the $n\times n$ identity matrix.

Now we specify the upper diagonal block matrices $\mathbf{A}_{1,0}^{0,m}$, $0 \le m \le M-1$, of the boundary block $\mathbf{A}_{1,0}$ as

$$\mathbf{A}_{1,0}^{0,m} = \begin{bmatrix} \mathbf{H}_{s_0} & \mathbf{0}_{s_0} & & & & & \\ & \mathbf{H}_{s_1} & \mathbf{0}_{s_1} & & & & \\ & & \ddots & \ddots & & & \\ & & & \mathbf{H}_{s_l} & \mathbf{0}_{s_l} & & \\ & & & & \ddots & \ddots & \\ & & & & & \mathbf{H}_{s_m} & \mathbf{0}_{s_m} \\ & & & & & & \end{bmatrix}_{\gamma_m \times \gamma_{m+1}}$$

with sub-blocks

$$\mathbf{H}_{s_l} = \lambda\, \boldsymbol{I}_{(s_l+1)\times(s_l+1)} \quad \text{for} \quad l = 0,1,2,\dots,m,$$

and

$$\mathbf{0}_{s_l} = \begin{cases} \mathbf{0}_{(s_l+1)\times(s_{l+1}+1)} & \text{for } l = 0,1,2,\dots,J_2-1, \\ \mathbf{0}_{(s_{l+1}+1)\times(s_{l+1}+1)} & \text{for } l = J_2,\ J_2+1,\ J_2+2,\dots,m. \end{cases}$$

the lower diagonal sub-blocks $\mathbf{A}_{1,0}^{2,m}$, $1 \le m \le M$, of the boundary block matrices $\mathbf{A}_{1,0}$ are defined by

$$\mathbf{A}_{1,0}^{2,m} = \begin{bmatrix} \mathbf{G}_{r_m} & & & & & & \\ \mathbf{F}_{s_1} & \mathbf{G}_{r_{m-1}} & & & & & \\ & \mathbf{F}_{s_2} & \mathbf{G}_{r_{m-2}} & & & & \\ & & \ddots & \ddots & & & \\ & & & \mathbf{F}_{s_l} & \mathbf{G}_{r_{m-l}} & & \\ & & & & \ddots & \ddots & \\ & & & & & \mathbf{F}_{s_{m-1}} & \mathbf{G}_{r_1} \\ & & & & & & \mathbf{F}_{s_m} \\ & & & & & & \end{bmatrix}_{\gamma_m \times \gamma_{m-1}}$$

with partitioned sub-block matrices

$$\mathbf{G}_{r_{m-l}} = \left[r_{m-l}(1-p_1)\mu_1 \mathbf{I}_{s_{l+1}} \right]_{(s_l+1)\times(s_l+1)} \qquad \text{for } l = 0,1,2,\dots,m-1,$$

$$F_{s_l} = \begin{cases} \begin{bmatrix} 0 & & & & \\ p(1-p_2)\mu_2 & (1-p)(1-p_2)\mu_2 & & & \\ & 2p(1-p_2)\mu_2 & 2(1-p)(1-p_2)\mu_2 & & \\ & \ddots & \ddots & & \\ & & & (s_l-1)p(1-p_2)\mu_2 & (s_l-1)(1-p)(1-p_2)\mu_2 \\ & & & & s_l(1-p_2)\mu_2 \end{bmatrix}_{(s_l+1)\times(s_{l-1}+1)} & \text{for } l = 1, 2, \dots, J_2, \\ \begin{bmatrix} 0 & & & & \\ p(1-p_2)\mu_2 & (1-p)(1-p_2)\mu_2 & & & \\ & 2p(1-p_2)\mu_2 & 2(1-p)(1-p_2)\mu_2 & & \\ & \ddots & \ddots & & \\ & & (s_l-1)p(1-p_2)\mu_2 & (s_l-1)(1-p)(1-p_2)\mu_2 & \\ & & s_l p(1-p_2)\mu_2 & s_l(1-p)\ (1-p_2)\mu_2 & \end{bmatrix}_{(s_l+1)\times(s_l+1)} & \text{for } \quad l = J_2+1, J_2+1, \dots, M. \end{cases}$$

The remaining main diagonal block matrices $\mathbf{A}_{1,n},\ 1 \le n \le K$, of $\boldsymbol{Q}$ can be expressed as

$$\mathbf{A}_{1,n} = \mathbf{A}_{1,0} - n\nu \begin{bmatrix} \mathbf{I}_{\Gamma_{M-1}} & \mathbf{0}_{\Gamma_{M-1}\times\gamma_M} \\ \mathbf{0}_{\gamma_M\times\Gamma_{M-1}} & \mathbf{0}_{\gamma_M\times\gamma_M} \end{bmatrix}_{\Gamma_M\times\Gamma_M}. \tag{3}$$

Next, the lower diagonal block matrices $\mathbf{A}_{2,n},\ 1 \le n \le K$, of $\boldsymbol{Q}$ are represented as

$$\mathbf{A}_{2,n} = n \begin{bmatrix} \mathbf{0} & \mathbf{A}_{2,n}^{0,0} & & & & & \\ & \mathbf{0} & \mathbf{A}_{2,n}^{0,1} & & & & \\ & & \ddots & \ddots & & & \\ & & & \mathbf{0} & \mathbf{A}_{2,n}^{0,m} & & \\ & & & & \ddots & \ddots & \\ & & & & & \mathbf{0} & \mathbf{A}_{2,n}^{0,M-1} \\ & & & & & & \mathbf{0} \end{bmatrix}_{\Gamma_M\times\Gamma_M} \tag{4}$$

in which, the off-diagonal block matrices $\mathbf{A}_{2,n}^{0,m}$ can be obtained from $\mathbf{A}_{1,0}^{0,m}$ by replacing λ with ν for $m = 0, 1, 2, \dots, M-1$.

Finally, the upper diagonal block matrices $\mathbf{A}_{0,n}$, of $\boldsymbol{Q}$ are defined as

$$\mathbf{A}_{0,n} = \begin{bmatrix} \mathbf{0}_{\Gamma_{M-1}\times\Gamma_{M-1}} & \mathbf{0}_{\Gamma_{M-1}\times\gamma_M} \\ \mathbf{0}_{\gamma_M\times\Gamma_{M-1}} & \lambda\boldsymbol{I}_{\gamma_M\times\gamma_M} \end{bmatrix}_{\Gamma_M\times\Gamma_M} \quad \text{for } n = 0, 1, 2, \dots, K. \tag{5}$$

Under the assumptions on primary program (packet) arrivals, the repeated attempts from the orbit and the representations of service time distributions, the time-homogenous Markov process $\{X(t);\ t \ge 0\}$ on the finite state space Ω is irreducible. Hence it admits a unique stationary distribution. The LDQBDs have been well studied in the literature (see Gaver et al. [19] and Servi [34]). Hence it is possible to adopt a computational procedure proposed by Elhafsi and Molle [13] to compute the steady-state probabilities associated with our system, which we now describe below.

Let $\boldsymbol{Y}$, be partitioned according to the orbit level as $\boldsymbol{Y} = (\boldsymbol{Y}_0, \boldsymbol{Y}_1, \boldsymbol{Y}_2, \dots, \boldsymbol{Y}_K)$, denote the steady-state probability vector of $\boldsymbol{Q}$. That is, $\boldsymbol{Y}$ is the unique solution for the linear system of equations

$$\boldsymbol{YQ} = \mathbf{0}, \quad \text{and} \quad \boldsymbol{Ye}_{\Lambda_M\times 1} = 1. \tag{6}$$

where $\boldsymbol{e}_{\Lambda_M\times 1} = [1,1,1,\dots,1]^T_{\Lambda_M\times 1}$.

For use in the sequel, we further partition the probability vectors $\boldsymbol{Y}_n,\ 0\le n\le K$, as

$$\boldsymbol{Y}_n = [\boldsymbol{Y}_{n,0}, \boldsymbol{Y}_{n,1}, \boldsymbol{Y}_{n,2}, \dots, \boldsymbol{Y}_{n,m}, \boldsymbol{Y}_{n,m+1}, \dots, \boldsymbol{Y}_{n,M}],$$

where, for $m = 0,1,2,\dots,M$,

$$\begin{aligned}\boldsymbol{Y}_{n,m} = \ & [\ Y(n,m,0,0), Y(n,m-1,1,0), Y(n,m-1,1,1), Y(n,m-2,2,0), Y(n,m-2,2,1)\\ & Y(n,m-2,2,2),\dots, Y(n,1,m-1,0), Y(n,1,m-1,1), Y(n,1,m-1,2),\dots,\\ & Y(n,1,m-1,s_{m-1}), Y(n,0,m,0), Y(n,0,m,1), Y(n,0,m,2),\dots, Y(n,0,m,s_m)\ ]_{1\times\gamma_m},\end{aligned}$$

in which $Y(n,m-l,l,j)$ denotes the steady-state probability that there are n programs in the retrial orbit, $m-l$ programs in the I/O queue, l programs in the CPU queue and there are j channels busy in the CPU queue. From (6) , the steady-state probabilities $\boldsymbol{Y}_n,\ 0\le n\le K$, can be represented in the form of recursive relations

$$\boldsymbol{Y}_n = \boldsymbol{Y}_{n+1}\boldsymbol{R}_n, \qquad \text{for } n = 0,1,2,\dots,K-1, \tag{7}$$

where the set of matrices $\{\boldsymbol{R}_n\}_{n=0}^{K-1}$ satisfy the recursive relations

$$\boldsymbol{R}_0 = -\boldsymbol{A}_{2,1}\boldsymbol{A}_{1,0}^{-1}, \quad \boldsymbol{R}_n = -\boldsymbol{A}_{2,n+1}[\boldsymbol{R}_{n-1}\boldsymbol{A}_{0,n-1} + \boldsymbol{A}_{1,n}]^{-1}, \quad \text{for} \quad n = 1,2,3,\dots,K-1. \tag{8}$$

Moreover, the steady-state probabilities $\boldsymbol{Y}_n,\quad 0\le n\le K$, can be expressed as

$$\boldsymbol{Y}_n = \boldsymbol{Y}_K \prod_{j=1}^{K-n} \boldsymbol{R}_{K-j}, \qquad \text{for } n = 0,1,2,\dots,K-1. \tag{9}$$

Now, using the boundary condition at $n = K$ and the normalization condition, one can obtain the following system of linear equations for $\boldsymbol{Y}_K$ as

$$\boldsymbol{Y}_K[\boldsymbol{R}_{K-1} + (\boldsymbol{A}_{0,K} + \boldsymbol{A}_{1,K})\boldsymbol{A}_{0,K-1}^{-1}] \ = \ \boldsymbol{0},$$

and

$$\boldsymbol{Y}_K[\boldsymbol{I}_{\Gamma_M} + \textstyle\sum_{n=0}^{K-1}\prod_{j=1}^{K-n} \boldsymbol{R}_{K-j}]\ \boldsymbol{e}_{\Gamma_M\times 1} = 1. \tag{10}$$

Once $\boldsymbol{Y}_K$ is determined, we obtain $\boldsymbol{Y}_n,\ 0\le n\le K-1$, via (9). It is clear that the steady-state probability vector $\boldsymbol{Y}$ is a unique solution of (6).

4. System Performance Measures

Once the steady-state probabilities have been computed, we can easily find the main system performance characteristics. These measures are used to bring out the qualitative behaviour of the retrial queueing network system under study.

1. The probability, P_{Empty}, that the system is empty as

$$P_{Empty} \ = \ Y(0,0,0,0) = \boldsymbol{Y}_0\boldsymbol{e}_1, \text{ where } \boldsymbol{e}_1 \ = \ [1,0,0,\dots,0]^T_{\Gamma_M\times 1}.$$

2. The utilization factor, U, of J_1 I/O service channels and J_2 CPU service channels, is obtained as

$$U = \sum_{n=0}^{K} \boldsymbol{Y}_n\boldsymbol{e}_u,$$

where $\boldsymbol{e}_u$ is a column vector of dimension $\Gamma_M\times 1$ whose elements are arranged according to the set $\{(n,m-l,l,j); 0\le n\le K,\ 0\le j\le s_l,\ 0\le l\le m,\ 0\le m\le M\}$ such that the elements correspondingto the set $\{(n,m-l,l,J_2);\ 0\le n\le K, m-l\ge J_1,\ l\ge J_2,\ J_1+J_2\le m\le M\}$ are equal to one and remaining elements are zero.

3. The blocking/loss probability, P_{Loss}, of a newly arriving primary program is found to be

$$P_{Loss} = \sum_{l=0}^{M} \sum_{j=0}^{J_2} Y(K, M-l, l, j) = \boldsymbol{Y}_K \boldsymbol{e}_{Loss}, \text{ where } \boldsymbol{e}_{Loss} = [\underbrace{0, 0, \dots, 0}_{\Gamma_{M-1}}, \underbrace{1, 1, \dots, 1}_{\gamma_M}]^T_{\Gamma_M \times 1}.$$

4. The throughput, T_s, of the system is the rate of successful transmission of an arriving primary program and is computed by

$$T_s = \lambda\,(1 - P_{Loss}).$$

5. The total loss rate, L_{Rate}, of the arriving primary programs is determined as

$$L_{Rate} = \lambda P_{Loss}.$$

6. The probability, $P_{INEmpty}$, that there is no program in the inner multiprocessor network is calculated by

$$P_{INEmpty} = \sum_{n=0}^{K} Y(n, 0,0,0) = \sum_{n=0}^{K} \boldsymbol{Y}_n \boldsymbol{e}_1.$$

7. The probability, P_{INF}, that the inner multiprocessor network is fully occupied by M programs is obtained as:

$$P_{INF} = \sum_{n=0}^{K} \sum_{l=0}^{M} \sum_{j=0}^{s_l} Y(n, M-l, l, j) = \sum_{n=0}^{K} \boldsymbol{Y}_n \boldsymbol{e}_{NF}, \text{ where } \boldsymbol{e}_{NF} = \boldsymbol{e}_{Loss}.$$

8. The probability, $P_{I/OEmpty}$, that there is no program in I/O queue, i.e, all J_1 service channels in I/O queue are idle, as

$$P_{I/OEmpty} = \sum_{n=0}^{K} \sum_{l=0}^{M} \sum_{j=0}^{s_l} Y(n, 0, l, j) = \sum_{n=0}^{K} \boldsymbol{Y}_n \boldsymbol{e}_{I/O,E},$$

where $\boldsymbol{e}_{I/O,E} = [\boldsymbol{e}_{i/o,0}, \boldsymbol{e}_{i/o,1}, \boldsymbol{e}_{i/o,2}, \dots, \boldsymbol{e}_{i/o,m}, \dots, \boldsymbol{e}_{i/o,M}]^T_{\Gamma_M \times 1}$ with elements $\boldsymbol{e}_{i/o,0} = [1]$, $\boldsymbol{e}_{i/o,m} = [0, 0, 0, \dots, 0, \underbrace{1, 1, \dots, 1}_{(s_m+1)}]^T_{1\times\gamma_m}$, for $m = 1, 2, 3, \dots, M$.

9. The Probability, $P_{CPUEmpty}$, that there is no program in CPU queue is computed as

$$P_{CPUEmpty} = \sum_{n=0}^{K} \sum_{m=0}^{M} Y(n, m, 0,0) = \sum_{n=0}^{K} \boldsymbol{Y}_n \boldsymbol{e}_{CPU,E},$$

where $\boldsymbol{e}_{CPU,E} = [\boldsymbol{e}_{cpu,0}, \boldsymbol{e}_{cpu,1}, \boldsymbol{e}_{cpu,2}, \dots, \boldsymbol{e}_{cpu,m}, \dots, \boldsymbol{e}_{cpu,M}]^T_{\Gamma_M \times 1}$ with elements $\boldsymbol{e}_{cpu,0} = [1]$, $\boldsymbol{e}_{cpu,m} = [1, 0, 0, \dots, 0]_{1\times\gamma_m}$, for $m = 1,2,3, \dots, M$.

10. The probability, $P_{I/O,F}$, that all M programs are buffered at I/O queue is determined by

$$P_{I/OF} = \sum_{n=0}^{K} Y(n, M, 0,0) = \sum_{n=0}^{K} \boldsymbol{Y}_n \boldsymbol{e}_{I/O,f},$$

where $\boldsymbol{e}_{I/O,f} = [\underbrace{0, 0, 0, \dots, 0}_{\Gamma_{M-1}}, 1, \underbrace{0, 0, 0, \dots, 0}_{\gamma_{M-1}}]^T_{\Gamma_m \times 1}$.

11. The probability, $P_{CPU,F}$, that all M programs buffered at CPU queue is calculated as

$$P_{CPU,F} = \sum_{n=0}^{K} \sum_{j=0}^{s_M} Y(n, 0, M, j) = \sum_{n=0}^{K} \boldsymbol{Y}_n \boldsymbol{e}_{cpu,f},$$

where $\boldsymbol{e}_{cpu,f} = [0, 0, 0, \dots, 0, \underbrace{1, 1, 1, \dots, 1}_{J_2+1}]^T_{\Gamma_M \times 1}$.

12. The probability, $P_{I/O,J_1Busy}$, that all J_1 service channels are busy at I/O queue is given as

$$P_{I/O,J_1Busy} = \sum_{n=0}^{K} \boldsymbol{Y}_n \boldsymbol{e}_{i/o,b},$$

where $\boldsymbol{e}_{i/o,b}$ is a column vector of dimension $\Gamma_M \times 1$ in which the elements are arranged according to the set $\{(n, m-l, l, j);\ 0 \le n \le K,\ 0 \le j \le s_l,\ 0 \le l \le m,\ 0 \le m \le M\}$such that the elements corresponding to the set $\{(n, m-l, l, j);\ 0 \le n \le K,\ 0 \le l \le m - J_1, J_1 \le m \le M\}$ are equal to one and the remaining elements are zero.

13. The probability, P_{CPU,J_2Busy}, that all J_2 service channels are busy at CPU queue is obtained as

$$P_{CPU,J_2Busy} = \sum_{n=0}^{K} \boldsymbol{Y}_n \boldsymbol{e}_{cpu,b},$$

where $\boldsymbol{e}_{cpu,b}$ is a column vector of dimension $\Gamma_M \times 1$ whose elements are arranged according to the set $\{(n, m-l, l, j);\ 0 \le n \le K,\ 0 \le j \le s_l,\ 0 \le l \le m,\ 0 \le m \le M\}$ such that the elements corresponding to the set $\{(n, m-l, l, J_2);\ 0 \le n \le K,\ J_2 \le l \le m, J_2 \le m \le M\}$ are equal to one and the remaining elements are zero.

14. The probability, $P_{CPU,vac}$, that all J_2 service channels are on vacation at CPU queue is computed as $P_{CPU,vac} = \sum_{n=0}^{K} Y_n \boldsymbol{e}_{cpu,v}$, where $\boldsymbol{e}_{cpu,v}$ is a column vector of dimension $\Gamma_M \times 1$ whose elements are arranged according to the set $\{(n, m-l, l, j);\ 0 \le n \le K,\ 0 \le j \le s_l,\ 0 \le l \le m,\ 0 \le m \le M\}$ such that the elements corresponding to the set $\{(n, m-l, l, 0);\ 0 \le n \le K,\ 0 \le l \le m,\ 0 \le m \le M\}$ are equal to one and the remaining elements are zero.
15. The mean number, $E(X_Q)$, of programs in the orbit is computed as
$E(X_Q) = \sum_{n=1}^{K} n\boldsymbol{Y}_n \boldsymbol{e}$, where $\mathbf{e} = [1, 1, 1, \dots, 1]^T_{\Gamma_M \times 1}$.
16. The mean number, $E(X_{IN})$, of programs in the inner multiprocessor network is calculated by
$$E(X_{IN}) = \sum_{n=0}^{K} \boldsymbol{Y}_n \boldsymbol{e}_{IN},$$
where $\boldsymbol{e}_{IN} = [\boldsymbol{e}_{IN,0},\ \boldsymbol{e}_{IN,1}, \dots,\ \boldsymbol{e}_{IN,m}, \dots, \boldsymbol{e}_{IN,M}]^T_{\Gamma_M \times 1}$ with elements $\boldsymbol{e}_{IN,m} = \mathrm{m}\,[1, 1, 1, \dots, 1]^T_{\gamma_m \times 1}$, $0 \le m \le M$.
17. The mean waiting time, $E(W_Q)$, of a program in the orbit is obtained as
$$E(W_Q) = \frac{E(X_Q)}{\lambda(1 - P_{Loss})}.$$
18. The mean number, $E(X_{I/O})$, of programs in the I/O queue is determined as
$$E(X_{I/O}) = \sum_{n=0}^{K} \boldsymbol{Y}_n \boldsymbol{e}_{I/O},$$
where $\boldsymbol{e}_{I/O} = [\boldsymbol{e}_{I/O,0},\ \boldsymbol{e}_{I/O,1}, \dots, \boldsymbol{e}_{I/O,m}, \dots, \boldsymbol{e}_{I/O,M}]^T_{\Gamma_M \times 1}$ with elements $\boldsymbol{e}_{I/O,m} = [\boldsymbol{e}_{I/O,m_0},$ $\boldsymbol{e}_{I/O,m_1}, \dots, \boldsymbol{e}_{I/O,m_l}, \dots, \boldsymbol{e}_{I/O,m_m}]^T$ in which $\boldsymbol{e}_{I/O,m_l} = (m - l)\ [1, 1, \dots, 1]^T_{(s_l+1)\times 1}$, $0 \le l \le m$, $0 \le m \le M$.
19. The mean number, $E(X_{CPU})$, of programs at the CPU queue is given by
$$E(X_{CPU}) = \sum_{n=0}^{K} \boldsymbol{Y}_n e_{CPU},$$
where $\boldsymbol{e}_{CPU} = [\boldsymbol{e}_{CPU,0},\ \boldsymbol{e}_{CPU,1}, \dots, \boldsymbol{e}_{CPU,m}, \dots, \boldsymbol{e}_{CPU,M}]^T_{\Gamma_M \times 1}$ with elements $\boldsymbol{e}_{CPU,m} = [\boldsymbol{e}_{CPU,m_0},$ $\boldsymbol{e}_{CPU,m_1}, \dots, \boldsymbol{e}_{CPU,m_l}, \dots, \boldsymbol{e}_{CPU,m_m}]$ in which $\boldsymbol{e}_{CPU,m_l} = l[1, 1, \dots,]^T_{(s_l+1)\times 1}$, $0 \le l \le m, 0 \le m \le M$.
20. The overall rate of retrial γ_1^* is obtained as
$$\gamma_1^* = \nu \sum_{n=1}^{K} n\boldsymbol{Y}_n \boldsymbol{e}_{\Gamma_M \times 1}.$$
21. The rate of retrials that are successful γ_2^* is determined as
$$\gamma_2^* = \gamma_1^* - \nu \sum_{n=1}^{K} n\boldsymbol{Y}_n \boldsymbol{e}_{rs},$$
where $\boldsymbol{e}_{rs} = [\underbrace{0,0,0,\dots,0}_{\Gamma_{M-1}}, \underbrace{1,1,1,\dots,1}_{\gamma_M}]^T_{\Gamma_M \times 1}$.
22. The fraction of successful rate of retrials is given by $F = \frac{\gamma_2^*}{\gamma_1^*}$.

We now report a few numerical results of $E(X_Q)$. In our first study, we present the mean, $E(X_Q)$, of the number of programs in the orbit of FOC level dependent retrial queueing network system for $p = 0$ (exhaustive service), $p = 0.5$ (Bernoulli scheduling service) and $p = 1$ (1-limited service), respectively, in Tables 1-3 as function of λ for $\nu = 5$, $\theta = 5$, $M = 6$, and $K = 10$, three levels of feedback probabilities (p_1, p_2), the service rates (μ_1, μ_2) and the number of service channels (J_1, J_2) of the I/O and the CPU queues. It is observed from Tables 1-3 that the performance measure $E(X_Q)$ increases as λ increases for $p = 0, 0.5$ and 1 cases. It is perceptible that with the increase in the arrival rate λ of the primary programs, the inner multiprocessor network is fully occupied by M programs and the blocked programs are going to the retrial orbit, thus $E(X_Q)$ is to be increased. Besides, Tables 1-3 indicate that $E(X_Q)$ has the lowest value for the exhaustive service vacation system than the other vacation systems. Moreover, Tables 1-3 indicate that $E(X_Q)$ increases for decreasing values of μ_1 and increasing values of μ_2. On the other hand, $E(X_Q)$ increases for increasing values of p_1 and decreasing values of p_2 for fixed value of λ.

Table 1. $E(X_Q)$ versus arrival rate λ for $\nu=5$, $\theta=5$, $p=0$, $M=6$, $K=10$.

λ	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$
	$(\mu_1=9,\ \mu_2=8)$			$(\mu_1=8.5,\ \mu_2=8.5)$			$(\mu_1=8,\ \mu_2=9)$		
	$(P_1=0.3,\ P_2=0.7)$								
5	0.1885	0.3968	0.0627	0.2378	0.517	0.0729	0.3093	0.6832	0.0877
6	0.5496	0.916	0.1809	0.6857	1.1588	0.2104	0.875	1.4729	0.2526
7	1.2354	1.7131	0.4233	1.5003	2.0776	0.4905	1.8429	2.5088	0.5849
8	2.2085	2.6727	0.8355	2.5802	3.0965	0.9601	3.0191	3.5564	1.1291
9	3.2695	3.6174	1.4257	3.673	4.0307	1.6155	4.114	4.4541	1.8625
10	4.2254	4.4359	2.1451	4.6027	4.8063	2.3884	4.9971	5.1772	2.6901
11	5.0068	5.1093	2.9079	5.3413	5.4361	3.1791	5.6859	5.7643	3.501
12	5.6326	5.6643	3.635	5.9298	5.9584	3.9094	6.2364	6.257	4.2241
	$(P_1=0.5,\ P_2=0.5)$								
5	0.5591	0.6429	0.2743	0.6419	0.7936	0.2904	0.7616	0.9983	0.3178
6	1.3985	1.4434	0.7102	1.5781	1.713	0.7513	1.8218	2.0521	0.8191
7	2.5696	2.5163	1.4317	2.8212	2.8565	1.5076	3.1373	3.2498	1.6275
8	3.745	3.6022	2.3509	4.0051	3.9376	2.4563	4.3128	4.303	2.6154
9	4.7128	4.5222	3.2959	4.9469	4.817	3.414	5.216	5.131	3.5852
10	5.4606	5.2524	4.1376	5.6666	5.5077	4.2537	5.902	5.7804	4.4172
11	6.0488	5.8373	4.8365	6.2344	6.0651	4.9441	6.447	6.311	5.0935
12	6.531	6.3231	5.4089	6.7026	6.5329	5.5076	6.8991	6.7609	5.6438
	$(P_1=0.7,\ P_2=0.3)$								
5	1.0147	0.7755	0.6591	1.0725	0.8936	0.653	1.1721	1.0603	0.6668
6	2.2136	1.734	1.5161	2.3183	1.9348	1.5069	2.4869	2.2021	1.5369
7	3.5351	2.9315	2.6308	3.657	3.1672	2.6218	3.8416	3.462	2.6651
8	4.6327	4.0467	3.7226	4.7454	4.2651	3.7157	4.9097	4.5278	3.7624
9	5.4548	4.934	4.6275	5.553	5.12	4.6221	5.6944	5.3411	4.6651
10	6.0795	5.6166	5.3358	6.1673	5.7769	5.3307	6.2932	5.9682	5.3682
11	6.5798	6.159	5.8963	6.6611	6.3031	5.8911	6.7775	6.4764	5.9243
12	6.9983	6.61	6.3562	7.075	6.7439	6.3511	7.1844	6.9055	6.3816

Table 2. $E(X_Q)$ versus arrival rate λ for $\nu=5$, $\theta=5$, $p=0.5$, $M=6$, $K=10$.

λ	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$
	$\mu_1=9,\ \mu_2=8$			$\mu_1=8.5,\ \mu_2=8.5$			$\mu_1=8,\ \mu_2=9$		
	$P_1=0.3,\ P_2=0.7$								
5	0.2119	0.3981	0.0782	0.2631	0.5176	0.089	0.3369	0.6828	0.1048
6	1.2119	0.9232	0.2251	0.7513	1.1644	0.2562	0.9434	1.4767	0.3008
7	2.2119	1.7301	0.5213	1.6165	2.0911	0.5907	1.9536	2.5187	0.6883
8	3.2119	2.6985	1.01	2.7243	3.1162	1.1342	3.1432	3.5707	1.3031
9	4.2119	3.6473	1.6794	3.8108	4.0529	1.8607	4.2234	4.4699	2.0969
10	5.2119	4.4657	2.455	4.7182	4.8281	2.6769	5.0848	5.193	2.9534
11	6.2119	5.1374	3.2386	5.4356	5.457	3.4764	5.7567	5.7797	3.7615
12	7.2119	5.6907	3.9583	6.0092	5.9782	4.1923	6.2962	6.2719	4.4651
	$P_1=0.5,\ P_2=0.5$								
5	0.7943	0.6964	0.4632	0.8758	0.8455	0.4752	0.9969	1.0482	0.5021
6	1.8722	1.5643	1.1559	2.0284	1.8235	1.1841	2.2463	2.1505	1.2438
7	3.18	2.6943	2.1657	3.3699	3.0083	2.2106	3.6185	3.3747	2.3007
8	4.3375	3.7964	3.2586	4.5153	4.095	3.3111	4.7392	4.4264	3.4121
9	5.2302	4.7046	4.23	5.3834	4.962	4.2802	5.5743	5.2433	4.375
10	5.9138	5.4175	5.0198	6.0469	5.6391	5.0632	6.2138	5.8832	5.1458
11	6.4596	5.9892	5.6536	6.579	6.1869	5.6898	6.7298	6.4073	5.7611
12	6.913	6.466	6.1731	7.0224	6.648	6.2036	7.161	6.8523	6.2663
	$P_1=0.7,\ P_2=0.3$								
5	1.805	0.9723	1.4045	1.8215	1.0823	1.3549	1.8872	1.242	1.3365
6	3.3926	2.1235	2.8304	3.4141	2.291	2.7563	3.4942	2.5234	2.7262
7	4.7151	3.4273	4.1792	4.7328	3.6005	4.1036	4.8005	3.8308	4.0701
8	5.6586	4.5314	5.2062	5.6702	4.6775	5.1374	5.7227	4.8689	5.1041
9	6.3521	5.3662	5.9668	6.359	5.4853	5.9035	6.4015	5.6422	5.8701
10	6.8963	6.0047	6.556	6.9006	6.106	6.4964	6.9373	6.2417	6.4633
11	7.3415	6.5192	7.0331	7.3446	6.6099	6.9772	7.3775	6.7329	6.9449
12	7.7118	6.9518	7.4298	7.7146	7.0355	7.3777	7.7444	7.1495	7.3473

Table 3. $E(X_Q)$ versus arrival rate λ for $\nu=5$, $\theta=5$, $p=1.0$, $M=6$, $K=10$.

λ	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$
	($\mu_1=9$, $\mu_2=8$)			($\mu_1=8.5$, $\mu_2=8.5$)			($\mu_1=8$, $\mu_2=9$)		
λ	($P_1=0.3$, $P_2=0.7$)								
5	0.2465	0.4004	0.1021	0.2998	0.5188	0.1137	0.3766	0.6827	0.1308
6	0.7053	0.9339	0.2937	0.8464	1.173	0.3264	1.0408	1.4829	0.3739
7	1.5245	1.7546	0.6708	1.7802	2.111	0.7419	2.1077	2.5337	0.8424
8	2.5906	2.7357	1.266	2.9216	3.145	1.3873	3.3122	3.5917	1.5529
9	3.6591	3.6903	2.0334	3.9971	4.085	2.2004	4.3723	4.493	2.419
10	4.5693	4.5088	2.8666	4.8766	4.86	3.0598	5.2074	5.216	3.3024
11	5.2985	5.1784	3.6633	5.5698	5.4875	3.8613	5.8599	5.8023	4.1017
12	5.8852	5.7296	4.3689	6.1276	6.0076	4.5579	6.3876	6.294	4.7825
	($P_1=0.5$, $P_2=0.5$)								
5	1.2196	0.7857	0.8359	1.2925	0.9315	0.838	1.4074	1.1306	0.8598
6	2.6121	1.7618	1.9315	2.7264	2.0033	1.9366	2.8962	2.3101	1.9763
7	4.0035	2.9744	3.2297	4.1185	3.2478	3.2367	4.2819	3.5725	3.2834
8	5.0872	4.0933	4.382	5.1827	4.3392	4.3872	5.3173	4.6206	4.4285
9	5.8944	4.9833	5.2904	5.9717	5.1884	5.2913	6.0821	5.4227	5.323
10	6.5204	5.6752	5.9992	6.5847	5.8495	5.9958	6.6789	6.0513	6.0189
11	7.0269	6.2327	6.5671	7.0823	6.3869	6.5608	7.1651	6.5682	6.5777
12	7.4466	6.6999	7.0343	7.4954	6.8405	7.0265	7.569	7.0071	7.0396
	($P_1=0.7$, $P_2=0.3$)								
5	3.1755	1.3475	2.8401	3.1315	1.437	2.7376	3.1333	1.578	2.6693
6	4.8436	2.7924	4.5476	4.802	2.9	4.449	4.7977	3.0684	4.3791
7	5.9539	4.1802	5.7292	5.9151	4.2674	5.646	5.9047	4.4059	5.5843
8	6.7359	5.2223	6.5583	6.6986	5.2842	6.4852	6.6846	5.3869	6.4293
9	7.3303	5.9857	7.1823	7.2954	6.0308	7.117	7.2804	6.1107	7.0659
10	7.7963	6.5798	7.6692	7.765	6.6152	7.6115	7.7507	6.6822	7.5658
11	8.165	7.0666	8.0545	8.1376	7.0962	8.0043	8.1248	7.1549	7.9641
12	8.4576	7.4748	8.3613	8.434	7.5005	8.318	8.4228	7.5529	8.2832

In the second set of numerical examples, the influence of the retrial rate ν, by fixing $\lambda = 7, \theta = 5$, $M = 6$, and $K = 10$, on $E(X_Q)$ for $p = 0, 0.5$, and 1, respectively, are shown in Tables 4-6. Evidently, the numerical results show that $E(X_Q)$ decreases with increasing values of ν for all three values of the vacation parameter p as is to be expected. Moreover, it can be observed from Tables 4-6 that $E(X_Q)$ is strongly influenced by the feedback probabilities (p_1, p_2), the service rates (μ_1, μ_2) and the number of service channels (J_1, J_2) of I/O and CPU queues of level dependent retrial queueing network system. As before, Tables 4-6 reveal that the exhaustive service vacation system $(p = 0)$ has the lowest expected value, $E(X_Q)$, of orbit size comparing with the Bernoulli vacation schedule $(p = 0.5)$ and 1-limited service vacation system $(p = 1)$.

Now we come to the final set of numerical illustrations corresponding to the effect of the vacation rate θ on $E(X_Q)$. We show the behaviour of $E(X_Q)$ as a function of θ for $p = 0$, 0.5 and 1, respectively, in Tables 7-9, by choosing parameters $\lambda = 7$, $\nu = 5$, $M = 6$, and $K = 10$. It is clear that as θ increases, $E(X_Q)$ decreases for all the three vacation service systems. As before, it can be seen from Tables 7-9 that $E(X_Q)$ is strongly influenced by the feedback probabilities (p_1, p_2), the service rates (μ_1, μ_2) and the number of service channels (J_1, J_2) of the I/O and CPU queues.

Table 4. $E(X_Q)$ versus retrial rate ν for $\lambda=7$, $\theta=5$, $p=0$, $M=6$, $K=10$.

ν	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$
	($\mu_1=9$, $\mu_2=8$)			($\mu_1=8.5$, $\mu_2=8.5$)			($\mu_1=8$, $\mu_2=9$)		
	($P_1=0.3$, $P_2=0.7$)								
1	2.3988	1.4504	1.2114	2.7136	1.6939	1.3425	3.0899	2.014	1.5164
3	1.4738	0.79	0.573	1.7551	0.9651	0.6543	2.1125	1.2078	0.7669
5	1.2354	0.6405	0.4233	1.5003	0.7966	0.4905	1.8429	1.0163	0.5849
10	1.0326	0.5204	0.3027	1.2792	0.6596	0.3576	1.603	0.8582	0.4355
15	0.9543	0.4763	0.2595	1.1923	0.6088	0.3096	1.5069	0.799	0.3809
20	0.9113	0.4527	0.2369	1.1443	0.5815	0.2843	1.4533	0.7671	0.352
25	0.8839	0.4379	0.2228	1.1135	0.5644	0.2685	1.4189	0.7469	0.3339
30	0.8648	0.4277	0.2132	1.092	0.5525	0.2577	1.3947	0.733	0.3215
	($P_1=0.5$, $P_2=0.5$)								
1	3.885	2.8631	2.842	4.1006	3.0967	2.9221	4.369	3.4021	3.0516
3	2.8742	1.8737	1.7493	3.1212	2.0958	1.8274	3.4311	2.3934	1.9522
5	2.5696	1.6159	1.4317	2.8212	1.8311	1.5076	3.1373	2.121	1.6275
10	2.2898	1.3972	1.1481	2.5414	1.6044	1.2214	2.8575	1.8842	1.335
15	2.1762	1.3141	1.0387	2.4263	1.5175	1.1107	2.7407	1.7924	1.2212
20	2.1128	1.2691	0.9795	2.3617	1.4703	1.0507	2.6747	1.7424	1.1594
25	2.0719	1.2407	0.9423	2.32	1.4404	1.0128	2.6321	1.7106	1.1203
30	2.0434	1.221	0.9165	2.2909	1.4197	0.9867	2.6022	1.6887	1.0932
	($P_1=0.7$, $P_2=0.3$)								
1	3.5351	3.6365	4.0341	4.7859	3.7699	4.0221	4.7859	3.9686	4.0576
3	3.8145	2.6363	2.9721	3.931	2.7814	2.9617	3.931	2.9959	3.0038
5	3.5351	2.3598	2.6308	3.657	2.5069	2.6218	3.657	2.7227	2.6651
10	3.2689	2.119	2.3088	3.3933	2.2662	2.3018	3.3933	2.4804	2.3458
15	3.1589	2.0261	2.1795	3.2836	2.1728	2.1737	3.2836	2.3856	2.2178
20	3.0973	1.9758	2.1086	3.2221	2.1221	2.1035	3.2221	2.3338	2.1476
25	3.0578	1.9439	2.0637	3.1826	2.09	2.059	3.1826	2.3011	2.1032
30	3.0302	1.922	2.0325	3.1549	2.0678	2.0282	3.1549	2.2784	2.0724

Table 5. $E(X_Q)$ versus retrial rate ν for $\lambda=7$, $\theta=5$, $p=0.5$, $M=6$, $K=10$.

ν	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$	$J_{11}=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$
	$\mu_1=9$, $\mu_2=8$			$\mu_1=8.5$, $\mu_2=8.5$			$\mu_1=8$, $\mu_2=9$		
	$P_1=0.3$, $P_2=0.7$								
1	2.5717	1.5361	1.4335	2.8689	1.7782	1.5567	3.226	2.0958	1.7229
3	1.6097	0.8451	0.7007	1.8847	1.0224	0.7825	2.2333	1.2674	0.8968
5	1.3545	0.6867	0.5213	1.6165	0.8457	0.5907	1.9536	1.0685	0.6883
10	1.1347	0.5588	0.3746	1.3809	0.7013	0.4326	1.7023	0.9037	0.5147
15	1.0493	0.5117	0.3216	1.2879	0.6477	0.375	1.6011	0.8417	0.4508
20	1.0024	0.4865	0.2938	1.2364	0.6188	0.3446	1.5445	0.8082	0.4169
25	0.9724	0.4707	0.2765	1.2034	0.6007	0.3256	1.5081	0.7871	0.3957
30	0.9515	0.4597	0.2647	1.1803	0.5881	0.3126	1.4827	0.7725	0.3811
	$P_1=0.5$, $P_2=0.5$								
1	4.4801	3.1974	3.7133	4.6224	3.4039	3.7382	4.8163	3.6786	3.8089
3	3.5003	2.177	2.5529	3.6795	2.3846	2.5911	3.9168	2.6645	2.6762
5	3.18	1.8991	2.1657	3.3699	2.1044	2.2106	3.6185	2.3812	2.3007
10	2.8696	1.6574	1.7934	3.0671	1.8587	1.8454	3.3226	2.1298	1.9397
15	2.8696	1.5637	1.6433	2.939	1.7628	1.6979	3.1961	2.0308	1.7935
20	2.6659	1.5126	1.561	2.8664	1.7103	1.6169	3.1242	1.9764	1.7129
25	2.6184	1.4801	1.5088	2.8194	1.677	1.5653	3.0775	1.9418	1.6616
30	2.5851	1.4576	1.4726	2.7865	1.6538	1.5296	3.0448	1.9177	1.6259
	$P_1=0.7$, $P_2=0.3$								
1	5.706	4.2356	5.3703	5.7082	4.3223	5.2986	5.7552	4.4683	5.2626
3	4.9749	3.2911	4.5026	4.9879	3.3913	4.4258	5.0505	3.5563	4.3901
5	4.7151	3.0131	4.1792	4.7328	3.1176	4.1036	4.8005	3.2867	4.0701
10	4.4429	2.7578	3.8365	4.4652	2.8658	3.7643	4.5374	3.037	3.7348
15	4.3234	2.6545	3.6874	4.3476	2.7637	3.6171	4.4215	2.9352	3.5897
20	4.2551	2.5972	3.6029	4.2804	2.7069	3.5339	4.3553	2.8786	3.5077
25	4.2108	2.5604	3.5485	4.2368	2.6706	3.4802	4.3124	2.8423	3.4548
30	4.1797	2.5349	3.5104	4.2061	2.6453	3.4427	4.2822	2.8171	3.4179

Table 6. $E(X_Q)$ versus retrial rate ν for $\lambda=7$, $\theta=5$, $p=1.0$, $M=6$, $K=10$.

ν	$J_1=2$ $J_2=2$	$J_1=2$ $J_2=3$	$J_1=3$ $J_2=2$	$J_1=2$ $J_2=2$	$J_1=2$ $J_2=3$	$J_1=3$ $J_2=2$	$J_1=2$ $J_2=2$	$J_1=2$ $J_2=3$	$J_1=3$ $J_2=2$
	($\mu_1=9$, $\mu_2=8$)			($\mu_1=8.5$, $\mu_2=8.5$)			($\mu_1=8$, $\mu_2=9$)		
	($P_1=0.3$, $P_2=0.7$)								
1	2.8146	1.649	1.7557	3.0857	1.888	1.8653	3.4148	2.2011	2.0175
3	1.8038	0.9193	0.8943	2.0678	1.0988	0.9751	2.4021	1.3456	1.0893
5	1.5245	0.7491	0.6708	1.7802	0.9113	0.7419	2.1077	1.1372	0.8424
10	1.2799	0.6107	0.4846	1.5238	0.7571	0.546	1.8396	0.9635	0.6329
15	1.1843	0.5595	0.4169	1.4219	0.6995	0.4741	1.7311	0.8979	0.5552
20	1.1316	0.5321	0.3813	1.3654	0.6685	0.4361	1.6704	0.8624	0.5139
25	1.098	0.5148	0.3591	1.3291	0.649	0.4124	1.6313	0.84	0.488
30	1.0745	0.5029	0.3439	1.3038	0.6355	0.3961	1.6039	0.8245	0.4702
	($P_1=0.5$, $P_2=0.5$)								
1	5.2817	3.6646	4.828	5.3438	3.8302	4.8019	5.4504	4.0589	4.8113
3	4.3417	2.6278	3.6746	4.4414	2.8076	3.669	4.5895	3.0549	3.7029
5	4.0035	2.3255	3.2297	4.1185	2.5091	3.2367	4.2819	2.7593	3.2834
10	3.6577	2.0515	2.7691	3.7866	2.2372	2.7905	3.9634	2.488	2.851
15	3.5089	1.9422	2.5757	3.643	2.1282	2.603	3.8248	2.3784	2.6689
20	3.4244	1.8819	2.4682	3.5612	2.0679	2.4986	3.7456	2.3176	2.5672
25	3.3697	1.8434	2.3996	3.5082	2.0293	2.4318	3.6942	2.2787	2.5021
30	3.3313	1.8166	2.3518	3.471	2.0025	2.3853	3.6581	2.2516	2.4567
	($P_1=0.7$, $P_2=0.3$)								
1	6.8825	5.1585	6.7701	6.8312	5.0825	6.6853	6.8062	5.1585	6.6189
3	6.2079	4.3573	6.0244	6.1658	4.2673	5.9409	6.1514	4.3573	5.8777
5	5.9539	4.0982	5.7292	5.9151	4.0024	5.646	5.9047	4.0982	5.5843
10	5.6806	3.8383	5.4048	5.6448	3.7368	5.3216	5.6383	3.8383	5.2615
15	5.5604	3.7262	5.2611	5.5257	3.6222	5.1778	5.5209	3.7262	5.1184
20	5.492	3.6624	5.1794	5.4579	3.557	5.096	5.454	3.6624	5.0369
25	5.4479	3.621	5.1266	5.4141	3.5147	5.0432	5.4108	3.621	4.9843
30	5.4169	3.592	5.0897	5.3835	3.485	5.0062	5.3806	3.592	4.9474

Table 7. $E(X_Q)$ versus vacation rate θ for $\lambda=7$, $\nu=5$, $p=0$, $M=6$, $K=10$.

θ	$J_1=2$ $J_2=2$	$J_1=2$ $J_2=3$	$J_1=3$ $J_2=2$	$J_1=2$ $J_2=2$	$J_1=2$ $J_2=3$	$J_1=3$ $J_2=2$	$J_1=2$ $J_2=2$	$J_1=2$ $J_2=3$	$J_1=3$ $J_2=2$
	($\mu_1=9$, $\mu_2=8$)			($\mu_1=8.5$, $\mu_2=8.5$)			($\mu_1=8$, $\mu_2=9$)		
	($P_1=0.3$, $P_2=0.7$)								
1	3.301	2.4698	1.8288	3.0075	2.7806	1.9465	3.301	3.1372	2.0953
3	2.1075	1.807	0.6183	1.7648	2.1631	0.6985	2.1075	2.5828	0.8086
5	1.8429	1.7131	0.4233	1.5003	2.0776	0.4905	1.8429	2.5088	0.5849
10	1.6537	1.6606	0.3059	1.3151	2.0316	0.3636	1.6537	2.4711	0.4456
15	1.5925	1.6469	0.2726	1.2562	2.0201	0.3271	1.5925	2.4623	0.4049
20	1.5621	1.6407	0.2569	1.227	2.0151	0.3098	1.5621	2.4586	0.3856
25	1.5439	1.6372	0.2478	1.2096	2.0123	0.2997	1.5439	2.4566	0.3743
	($P_1=0.5$, $P_2=0.5$)								
1	5.2726	4.2921	4.358	5.1214	4.4749	4.4121	5.2726	4.6883	4.4949
3	3.661	2.8482	2.0917	3.3868	3.1549	2.1703	3.661	3.5093	2.2924
5	3.1373	2.5163	1.4317	2.8212	2.8565	1.5076	3.1373	3.2498	1.6275
10	2.7029	2.2903	0.958	2.3573	2.6558	1.027	2.7029	3.0782	1.1371
15	2.554	2.2225	0.8157	2.2003	2.5964	0.8812	2.554	3.0281	0.9864
20	2.4791	2.1902	0.7486	2.1219	2.5682	0.8122	2.4791	3.0046	0.9145
25	2.4339	2.1713	0.7097	2.0748	2.5518	0.7721	2.4339	2.991	0.8725
	($P_1=0.7$, $P_2=0.3$)								
1	6.2406	5.3795	5.8208	6.1825	5.4531	5.8067	6.2406	5.5557	5.8158
3	4.5395	3.5289	3.5797	4.4005	3.7158	3.5698	4.5395	3.9537	3.6046
5	3.8416	2.9315	2.6308	3.657	3.1672	2.6218	3.8416	3.462	2.6651
10	3.1712	2.4618	1.7827	2.9439	2.7396	1.7752	3.1712	3.0828	1.8216
15	2.9241	2.3098	1.4986	2.6829	2.602	1.492	2.9241	2.9617	1.5378
20	2.7977	2.2356	1.3613	2.5501	2.535	1.355	2.7977	2.9029	1.4002
25	2.7212	2.1917	1.2812	2.47	2.4955	1.2751	2.7212	2.8682	1.3197

Table 8. $E(X_Q)$ versus vacation rate θ for $\lambda = 7$, $\nu = 5$, p = 0.5, M = 6, K = 10.

θ	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$
	$\mu_1 = 9$, $\mu_2 = 8$			$\mu_1 = 8.5$, $\mu_2 = 8.5$			$\mu_1 = 8$, $\mu_2 = 9$		
	$P_1 = 0.3$, $P_2 = 0.7$								
1	3.9424	3.081	3.2399	4.1148	3.3339	3.3282	4.3178	3.6246	3.4409
3	1.7935	1.8715	0.8956	2.0534	2.2189	0.9794	2.3787	2.6286	1.0947
5	1.3545	1.7301	0.5213	1.6165	2.0911	0.5907	1.9536	2.5187	0.6883
10	1.0928	1.6633	0.3319	1.3491	2.0333	0.3901	1.6861	2.4719	0.473
15	1.0204	1.648	0.2857	1.2737	2.0207	0.3404	1.6092	2.4625	0.4186
20	0.9867	1.6414	0.2653	1.2383	2.0154	0.3183	1.5728	2.4587	0.3943
25	0.9671	1.6377	0.2539	1.2177	2.0125	0.3058	1.5516	2.4567	0.3805
	$P_1 = 0.5$, $P_2 = 0.5$								
1	7.0754	5.8836	6.9102	7.0972	5.9491	6.9081	7.1323	6.0343	6.9169
3	4.2678	3.3028	3.5248	4.389	3.5513	3.5503	4.5517	3.8444	3.6125
5	3.18	2.6943	2.1657	3.3699	3.0083	2.2106	3.6185	3.3747	2.3007
10	2.3109	2.3378	1.1953	2.5568	2.6949	1.2518	2.8761	3.1091	1.352
15	2.0498	2.2459	0.9405	2.3104	2.6152	0.9987	2.6491	3.0427	1.0981
20	1.9287	2.2049	0.8301	2.1955	2.58	0.8885	2.5424	3.0136	0.9868
25	1.8594	2.1818	0.7693	2.1293	2.5602	0.8277	2.4806	2.9973	0.925
	$P_1 = 0.7$, $P_2 = 0.3$								
1	8.3563	7.3847	8.3202	8.343	7.3801	8.2994	8.3359	7.3866	8.2836
3	5.9607	4.5205	5.7001	5.946	4.6064	5.6377	5.9599	4.7348	5.5985
5	4.7151	3.4273	4.1792	4.7328	3.6005	4.1036	4.8005	3.8308	4.0701
10	3.3581	2.6183	2.4767	3.4436	2.8724	2.417	3.6039	3.1925	2.4155
15	2.8614	2.3905	1.8932	2.9785	2.6695	1.8509	3.1774	3.0166	1.8661
20	2.6206	2.2875	1.6261	2.7536	2.578	1.5937	2.9711	2.9375	1.6171
25	2.4809	2.2292	1.4771	2.6231	2.5263	1.4507	2.8511	2.8928	1.4785

Table 9. $E(X_Q)$ versus vacation rate θ for $\lambda=7$, $\nu=5$, $p=1.0$, $M=6$, $K=10$.

θ	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$	$J_1=2$, $J_2=2$	$J_1=2$, $J_2=3$	$J_1=3$, $J_2=2$
	($\mu_1 = 9$, $\mu_2 = 8$)			($\mu_1 = 8.5$, $\mu_2 = 8.5$)			($\mu_1 = 8$, $\mu_2 = 9$)		
	($P_1 = 0.3$, $P_2 = 0.7$)								
1	5.1879	3.9228	4.7455	5.2996	4.1083	4.8059	5.4325	4.3233	4.8835
3	2.2298	1.9749	1.3394	2.4673	2.309	1.4231	2.7622	2.7036	1.5387
5	1.5245	1.7546	0.6708	1.7802	2.111	0.7419	2.1077	2.5337	0.8424
10	1.137	1.6666	0.3661	1.392	2.0354	0.4248	1.727	2.4729	0.5087
15	1.042	1.6492	0.3018	1.2946	2.0213	0.3566	1.629	2.4626	0.4353
20	1.0002	1.6421	0.2752	1.2513	2.0158	0.3282	1.5851	2.4587	0.4044
25	0.9767	1.6382	0.2608	1.2269	2.0128	0.3128	1.5602	2.4567	0.3876
	($P_1 = 0.5$, $P_2 = 0.5$)								
1	8.4332	7.3915	8.3934	8.4413	7.4112	8.3948	8.4543	7.44	8.4001
3	5.5152	4.0072	5.1284	5.5579	4.1739	5.1191	5.6273	4.38	5.1322
5	4.0035	2.9744	3.2297	4.1185	3.2478	3.2367	4.2819	3.5725	3.2834
10	2.617	2.4028	1.5424	2.8292	2.7485	1.5807	3.1124	3.1514	1.6651
15	2.2092	2.2753	1.1084	2.4514	2.639	1.1565	2.7706	3.0611	1.248
20	2.0308	2.2225	0.9333	2.2853	3.1129	0.9851	2.6195	3.0243	1.0783
25	1.9328	2.194	0.8417	2.1937	2.5698	0.8952	2.5358	3.0046	0.9889
	($P_1 = 0.7$, $P_2 = 0.3$)								
1	9.309	8.7315	9.3044	9.3063	8.7261	9.3004	9.305	8.7238	9.2978
3	7.3931	5.7649	7.3203	7.358	5.7674	7.271	7.3348	5.796	7.231
5	5.9539	4.1802	5.7292	5.9151	4.2674	5.646	5.9047	4.4059	5.5843
10	4.0829	2.8453	3.4192	4.099	3.065	3.3125	4.1813	3.3514	3.2583
15	3.2945	2.498	2.4297	3.361	2.7592	2.3461	3.5076	3.0894	2.3225
20	2.9112	2.3528	1.9719	3.0074	2.6319	1.908	3.1882	2.9808	1.9039
25	2.6942	2.2746	1.7236	2.8081	2.5635	1.6728	3.0086	2.9226	1.68

5. Optimal Retrial Rate

In this section, we use some probabilistic descriptors of the multiprogramming-multiprocessor computer retrial queueing network system where the service channels are under Bernoulli vacation schedule at CPU queue, in order to identify the optimal value of the retrial rate ν for chosen parametric values of $\lambda,\ \theta,\ p_1,\ p_2,\ p,\ \mu_1,\ \mu_2,\ J_1,\ J_2,\ M$ and K. This will help us to find the retrial rate as the total utilization of the inner multiprocessor network (main memory) of the system. We now study the two extreme cases of retrials, namely, ideal retrial and vain retrial for FOC as discussed by Artalejo et al. [6] and Krishna Kumar and Raja [27].

5.1. Ideal Retrials

An ideal retrial results in $M-1$ programs (packets) in the multiprocessor network becoming M programs after a retrial attempt made by a program from the orbit. Following an ideal retrial, the inner multiprocessor network is fully occupied by M programs. Moreover, the ideal retrial avoids the possibility of an unsuccessful repeat attempt and it represents the best possible choice for a repeated attempt.

The steady-state conditional probability, P_{IR}, that a retrial is ideal given that retrial occurs is defined as

$$P_{IR}=\frac{\sum_{n=1}^{K} n\nu \sum_{l=0}^{M-1}\sum_{j=0}^{s_l} Y(n,M-l-1,l,j)}{\sum_{n=1}^{K} n\nu \sum_{m=0}^{M}\sum_{l=0}^{m}\sum_{j=0}^{s_l} Y(n,m-l,l,j)}=\frac{\sum_{n=1}^{K} nY_n\boldsymbol{e}_i}{\sum_{n=1}^{K} nY_n\boldsymbol{e}},$$

where $\boldsymbol{e}_i=[\underbrace{0,0,0,\dots,0}_{\Gamma_{M-2}},\underbrace{1,1,\dots,1}_{\gamma_{M-1}}\underbrace{0,0,\dots,0}_{\gamma_M}]^T_{\Gamma_M\times 1}$.

5.2. Vain Retrials

A vain retrial is an unsuccessful retrial when the inner multiprocessor network is fully occupied by M programs. Such a retrial of the orbiting programs does not change the status of the number of programs in the inner multiprocessor network.

The steady-state probability, P_{VR}, of vain retrial is obtained as

$$P_{VR}=\frac{\sum_{n=1}^{K} n\nu \sum_{l=0}^{M}\sum_{j=0}^{s_l} Y(n,M-l,l,j)}{\sum_{n=1}^{K} n\nu \sum_{m=0}^{M}\sum_{l=0}^{m}\sum_{j=0}^{s_l} Y(n,m-l,l,j)}=\frac{\sum_{n=1}^{K} nY_n\boldsymbol{e}_v}{\sum_{n=1}^{K} nY_n\boldsymbol{e}},$$

where $\boldsymbol{e}_v=[\underbrace{0,0,0,\dots,0}_{\Gamma_{M-1}},\underbrace{1,1,\dots,1}_{\gamma_M}]^T_{\Gamma_M\times 1}$.

We now present some numerical results on the behaviour of the probability descriptors P_{IR} and P_{VR} for the level-dependent retrial queueing network system. We plot P_{IR} versus the retrial rate ν for three levels of service rates $(\mu_1,\ \mu_2)$ of the service channels of I/O and CPU queues and for three different values of the vacation parameter p, respectively, in Figures 2 and 3. Figure 2 shows that P_{IR} increases rapidly and then decreases monotonically for increasing values of ν. It should be pointed out that a proper value of ν maximizing P_{IR} always exists for chosen parametric values $p=p_1=p_2=0.5$, $J_1=2,\ J_2=3,\ \lambda=7,\ \theta=5,\ M=6$ and $K=10$. Moreover, it can be seen from Figure 2 that the retrial rate attains its optimal value early in the case of $\mu_1=8$ and $\mu_2=9$, whereas there is an onset of delay in the attainment of the optimal retrial rate in other cases. In Figure 3, we display P_{IR} against the retrial rate ν for three different values of the vacation parameter p for chosen values $p_1=p_2=0.5$, $J_1=2,\ J_2=3,\ \lambda=7,\ \theta=5,\ \mu_1=9,\ \mu_2=8,\ M=6$ and $K=10$. In this illustration also, P_{IR} increases rapidly and then decreases monotonically for increasing values of ν. Further, by considering higher vacation rate θ (of order 10^4) in the vacation system under investigation, we obtain approximate results for corresponding non-vacation model. For the systems with and without vacation, the curves in Figure 3 reveal that a proper value of ν maximizing P_{IR} always exists. An interesting observation from the plot is that P_{IR} attains its highest maximum in the case of 1-limited service disciple vacation system, i.e., when $p = 1$, as compared to the other systems under investigation. It is also worth noting that under the non-vacation system, the descriptor P_{IR} has lower value when compared to the exhaustive service vacation $(p = 0)$ system. This phenomenon can be explained as follows.

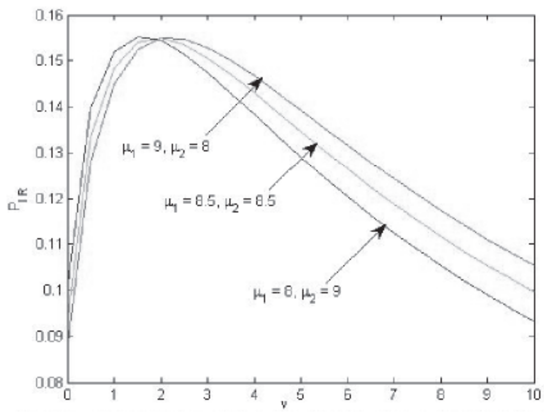

Figure 2. P_{IR} versus ν for $\lambda = 7, \theta = 5$, M=6, K=10, $p_1 = p_2 = p$=0.5, J_1=2, and J_2=3.

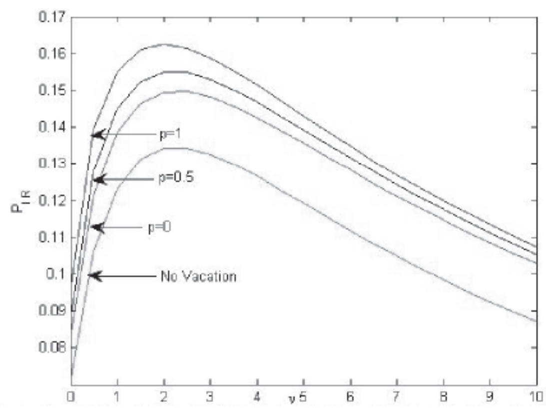

Figure 3. P_{IR} versus ν for $\lambda = 7, \theta = 5$, M=6, K=10, $p_1 = p_2$=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

In the non-vacation system, the service channels are always available even when there is no program in the CPU queue. Thus, the arriving programs to the CPU queue are immediately served without any additional delay. As a result, there is less chance that the inner network is occupied by $M - 1$ programs. On the other hand, in the exhaustive service vacation system, the service channel can avail multiple vacations when there is no program in the CPU queue. In this situation, the arriving programs during the vacation periods most likely have to wait for the remaining vacation time in the CPU queue buffer. Consequently, there is more chance that the inner network is occupied by $M - 1$ programs. Hence, under the exhaustive service vacation system, the ideal probability P_{IR} has attained higer value for the proper value of ν than that of the non-vacation system.

On the other hand, Figures 4 and 5 illustrate the trends of the vain retrial probability P_{VR} against ν, respectively, for three levels of service rates (μ_1, μ_2) of the service channels of I/O and CPU queues and three different values of the vacation parameter p. By taking $p = p_1 = p_2 = 0.5$, $J_1 = 2$, $J_2 = 3$, $\lambda = 7$, $\theta = 5$, $M = 6$ and $K = 10$ and for three different levels of service rates (μ_1, μ_2) of the service channels of I/O and CPU queues, Figure 4 exhibits the increasing trends of P_{VR} for increasing values of ν before attaining a limiting value. In Figure 5, we display the behaviour of P_{VR} versus ν for three different values of the vacation parameter p. By fixing $p_1 = p_2 = 0.5$, $J_1 = 2$, $J_2 = 3, \lambda = 7, \theta = 5$, $\mu_1 = 9$, $\mu_2 = 8$, $M = 6$ and $K = 10$, Figure 5 reveals the trends of P_{VR} versus ν for both vacation and non-vacation systems. As before, P_{VR} is the increasing function of ν before attaining a limiting value. The descriptor P_{VR} approaches the limiting value much faster in 1-limited service case than in the other cases.

Also, our numerical experience indicates that the two descriptors, the ideal retrial probability P_{IR} and vain retrial probability P_{VR} versus ν for the three levels of the feedback probabilities (p_1, p_2) behave very similar to P_{IR} and P_{VR} against ν for the three levels of the service rates (μ_1, μ_2) for other chosen parametric values of the system under discussion.

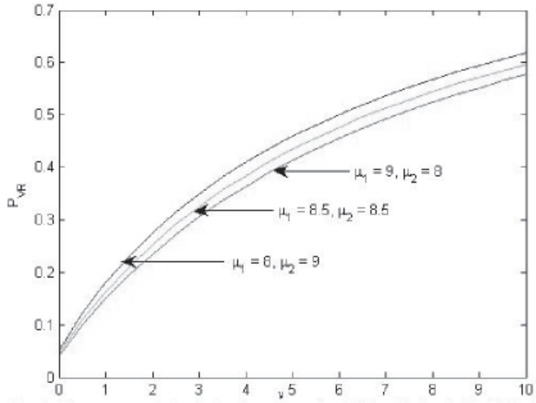

Figure 4. P_{VR} versus ν for λ =7, θ =5, M=6, K=10, p_1=p_2=p=0.5, J_1=2, and J_2=3.

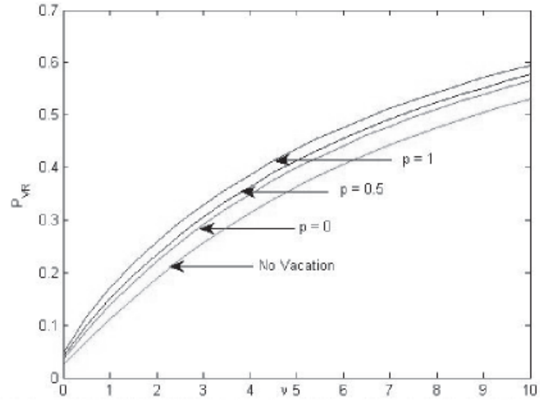

Figure 5. P_{VR} versus ν for λ =7, θ =5, M=6, K=10, p_1=p_2=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

6. Busy Period Analysis

Another relevant piece of information on the behaviour of our retrial queueing network system is the busy period. The busy period, L, of our system is defined as the period that starts at the epoch when an arriving primary program (packet) finds an empty system (no program in both the inner multiprocessor network and the orbit) and ends at the departure epoch when the system is empty again. Thus, the busy period is the first passage time from state $(0,1,0,0)$ to $(0,0,0,0)$. For the derivation of k^{th} order moments, $E(L^k)$, of the busy period for our FOC retrial queueing system with classical retrial policy, we develop algorithmic schemes for the computation of the Laplace-Stieltjes transform (LST) of the busy period L.

Let us consider again the process $\{X(t);\ t \geq 0\} = \{(N(t), I(t), C(t), J(t));\ t \geq 0\}$ which is a CTMC with state space given as

$$\Omega = \{(n, m-l, l, j);\ 0 \leq n \leq K,\ 0 \leq j \leq s_l,\ 0 \leq l \leq m,\ 0 \leq m \leq M\}.$$

We partition Ω as follows:

$$0^* = (0,0,0,0),$$

$$\mathbf{0} = \{(0, m-l, l, j);\ 0 \leq j \leq s_l, 0 \leq l \leq m, 1 \leq m \leq M\},$$

$$\boldsymbol{n} = \{(n, m-l, l, j);\ 1 \leq n \leq K,\ 0 \leq j \leq s_l,\ 0 \leq l \leq m,\ 0 \leq m \leq M\}.$$

The above states are again partitioned as

$$\mathbf{0} = (\mathbf{0}_1, \mathbf{0}_2, \mathbf{0}_3, \dots, \mathbf{0}_m, \dots, \mathbf{0}_M)$$

where, for $m = 1, 2, 3, \dots, M$,

$$\mathbf{0}_m = \{(0, m, 0,0), (0, m-1,1,0), (0, m-1,1,1), (0, m-2,2,0), (0, m-2,2,1), (0, m-2,2,2), \dots, (0,1, m-1,0), (0,1, m-1,1), (0,1, m-1,2), \dots, (0,1, m-1, s_{m-1}), (0,0, m, 0), (0,0, m, 1), (0,0, m, 2), \dots, (0,0, m, s_m)\},$$

and

$$\boldsymbol{n} = (\boldsymbol{n}_0, \boldsymbol{n}_1, \boldsymbol{n}_2, \dots, \boldsymbol{n}_m, \dots, \boldsymbol{n}_M),$$

with $\boldsymbol{n}_0 = (n, 0,0,0)$,and, for $m = 1, 2, 3, \dots, M$,

$$\boldsymbol{n}_m = \{(n, m, 0,0), (n, m-1,1,0), (n, m-1,1,1), (n, m-2,2,0), (n, m-2,2,1) (n, m-2,2,2), \dots, (n, 1, m-1,0), (n, 1, m-1,1), (n, 1, m-1,2), \dots, (n, 1, m-1, s_{m-1}), (n, 0, m, 0), (n, 0, m, 1), (n, 0, m, 2), \dots, (n, 0, m, s_m)\}.$$

Here both $\mathbf{0}_m$ and $\boldsymbol{n}_m$ are, respectively, row vectors of dimension $\frac{(s_m+1)(s_m+2)}{2}$ for $m = 1,2,3,\dots,J_2$ and of dimension $\frac{(s_m+1)(s_m+2)}{2} + (m - s_m)(s_m + 1)$ for $m = J_2 + 1, J_2 + 2, \dots, M$.

The busy period of our queueing system is the duration commencing while an arriving primary program finds the system in state 0^* and ends when the system visits state 0^* again at a service completion epoch.

Let $T_{(n,m-l,l,j)}$ be the first passage time to state 0^* (absorbing state) given that the initial state of the system is $(n, m - l, l, j)$ and $\phi_{(n,m-l,l,j)}(s) = E[e^{-sT_{(n,m-l,l,j)}}]$, $Re(s) \geq 0$, $(n, m - l, l, j) \in \Omega$, be the corresponding LST of $T_{(n,m-l,l,j)}$.

We introduce the following micro vectors which comprise the LSTs partitioned according to the orbit level: For $m = 1,2,3,\dots,M$,

$$\begin{aligned}\boldsymbol{\phi}_{0_m}(s) = \; & [\phi_{(0,m,0,0)}(s), \phi_{(0,m-1,1,0)}(s), \phi_{(0,m-1,1,1)}(s), \phi_{(0,m-2,2,0)}(s), \phi_{(0,m-2,2,1)}(s), \\ & \phi_{(0,m-2,2,2)}(s), \dots, \phi_{(0,1,m-1,0)}(s), \phi_{(0,1,m-1,1)}(s), \phi_{(0,1,m-1,2)}(s), \dots, \\ & \phi_{(0,1,m-1,s_{m-1})}(s), \phi_{(0,0,m,0)}(s), \phi_{(0,0,m,1)}(s), \phi_{(0,0,m,2)}(s), \dots, \phi_{(0,0,m,s_m)}(s)]^T,\end{aligned}$$

is a column vector whose dimension is the same as the dimension of $\mathbf{0}_m$, and, for $m = 0,1,2,3,\dots,M$,

$$\begin{aligned}\boldsymbol{\phi}_{n_m}(s) = \; & [\phi_{(n,m,0,0)}(s), \phi_{(n,m-1,1,0)}(s), \phi_{(n,m-1,1,1)}(s), \phi_{(n,m-2,2,0)}(s), \phi_{(n,m-2,2,1)}(s), \\ & \phi_{(n,m-2,2,2)}(s), \dots, \phi_{(n,1,m-1,0)}(s), \phi_{(n,1,m-1,1)}(s), \phi_{(n,1,m-1,2)}(s), \dots, \\ & \phi_{(n,1,m-1,s_{m-1})}(s), \phi_{(n,0,m,0)}(s), \phi_{(n,0,m,1)}(s), \phi_{(n,0,m,2)}(s), \dots, \phi_{(n,0,m,s_m)}(s)]^T,\end{aligned}$$

is also a column vector whose dimension is the same as the dimension of $\mathbf{n}_m$, $1 \leq n \leq K$.

Further, we generate the macro vectors

$$\boldsymbol{\phi}_{\mathbf{0}}(s) = [\boldsymbol{\phi}_{\mathbf{0_1}}(s), \boldsymbol{\phi}_{\mathbf{0_2}}(s), \boldsymbol{\phi}_{\mathbf{0_3}}(s), \dots, \boldsymbol{\phi}_{\mathbf{0_m}}(s), \dots, \boldsymbol{\phi}_{\mathbf{0_M}}(s)]^T_{(\Gamma_M-1)\times 1},$$
$$\boldsymbol{\phi}_{\mathbf{n}}(s) = [\boldsymbol{\phi}_{\mathbf{n_0}}(s), \boldsymbol{\phi}_{\mathbf{n_1}}(s), \boldsymbol{\phi}_{\mathbf{n_2}}(s), \dots, \boldsymbol{\phi}_{\mathbf{n_m}}(s), \dots, \boldsymbol{\phi}_{\mathbf{n_M}}(s)]^T_{\Gamma_M\times 1},$$

and

$$\boldsymbol{\phi}(s) = [\boldsymbol{\phi}_{\mathbf{0}}(s), \boldsymbol{\phi}_{\mathbf{1}}(s), \boldsymbol{\phi}_{\mathbf{2}}(s), \dots, \boldsymbol{\phi}_{\mathbf{n}}(s), \dots, \boldsymbol{\phi}_{\mathbf{K}}(s)]^T_{(\Lambda_M-1)\times 1}.$$

Moreover, we have $\phi_{0^*}(s) = 1$.

Theorem 1. *The LSTs* $\{\phi_{(n,m-l,l,j)}(s); (n, m - l, l, j) \in \Omega\}$ *satisfy the following tri-diagonal block system:*

$$\boldsymbol{T}_L(s)\boldsymbol{\phi}(s) = \boldsymbol{F} \tag{11}$$

where $\boldsymbol{T}_L(s) = \overline{\boldsymbol{Q}} - s\boldsymbol{I}_{(\Lambda_M-1)}$ *and* $\boldsymbol{F} = [-\boldsymbol{F}_{\mathbf{0}}, 0,0,0,\dots,0]^T_{(\Lambda_M-1)\times 1}$ *with* $\boldsymbol{F}_{\mathbf{0}} = [(1 - p_1)\mu_1,\ 0,\ (1 - p_2)\mu_2,\ 0,0,0,\dots,0]^T_{(\Gamma_M-1)\times 1}$ *and* $\mathbf{0}s'$ *are zero column vectors of order* $\Gamma_M \times 1$.

The square matrix

$$\overline{\mathbf{Q}} = \begin{bmatrix} \overline{\boldsymbol{A}}_{1,0} & \overline{\boldsymbol{A}}_{0,0} & & & & & & \\ \overline{\boldsymbol{A}}_{2,1} & \boldsymbol{A}_{1,1} & \boldsymbol{A}_{0,1} & & & & & \\ & \boldsymbol{A}_{2,2} & \boldsymbol{A}_{1,2} & \boldsymbol{A}_{0,2} & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & \boldsymbol{A}_{2,n} & \boldsymbol{A}_{1,n} & \boldsymbol{A}_{0,n} & & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & \boldsymbol{A}_{2,K-1} & \boldsymbol{A}_{1,K-1} & \boldsymbol{A}_{0,K-1} \\ & & & & & & \boldsymbol{A}_{2,K} & \boldsymbol{A}_{0,K} + \boldsymbol{A}_{1,K} \end{bmatrix}_{(\Lambda_M-1)\times(\Lambda_M-1)} \tag{12}$$

is obtained from $\boldsymbol{Q}$ *by removing the first row and first column. In other words,* $\overline{\boldsymbol{A}}_{1,0}$ *is obtained from* $\boldsymbol{A}_{1,0}$ *by removing the first row and first column,* $\overline{\boldsymbol{A}}_{0,0}$ *is obtained from* $\boldsymbol{A}_{0,0}$ *by removing its first row and* $\overline{\boldsymbol{A}}_{2,1}$ *is obtained from* $\boldsymbol{A}_{2,1}$ *by removing its first column.*

Proof. By employing the first-step principle, we get the following equations:

When $n = 0,\ j < l$,

$$\begin{aligned}\phi_{(0,m-l,l,j)}(s) &= \left(\frac{\lambda(1-\delta_{mM})}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(0,m-l+1,l,j)}(s)\\
&+\left(\frac{\lambda\delta_{mM}}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(1,m-l,l,j)}(s)\\
&+\left(\frac{(J_2-j)\theta}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(0,m-l,l,j+1)}(s)\\
&+\left(\frac{r_{m-l}p_1\mu_1}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(0,m-l-1,l+1,j)}(s)\\
&+\left(\frac{r_{m-l}(1-p_1)\mu_1}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(0,m-l-1,l,j)}(s)\\
&+\left(\frac{jpp_2\mu_2}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(0,m-l+1,l-1,j-1)}(s)\\
&+\left(\frac{j(1-p)p_2\mu_2}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(0,m-l+1,l-1,j)}(s)\\
&+\left(\frac{jp(1-p_2)\mu_2}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(0,m-l,l-1,j-1)}(s)\\
&+\left(\frac{j(1-p)(1-p_2)\mu_2}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(0,m-l,l-1,j)}(s)\end{aligned}$$

$$\text{for}\quad 0\le j\le s_l,\ 0\le l\le m,\ 1\le m\le M. \tag{13}$$

When $n = 0,\ j = l$,

$$\begin{aligned}\phi_{(0,m-l,l,l)}(s) &= \left(\frac{\lambda(1-\delta_{mM})}{s+\lambda+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(0,m-l+1,l,l)}(s)\\
&+\left(\frac{\lambda\delta_{mM}}{s+\lambda+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(1,m-l,l,l)}(s)\\
&+\left(\frac{r_{m-l}p_1\mu_1}{s+\lambda+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(0,m-l-1,l+1,l)}(s)\\
&+\left(\frac{r_{m-l}(1-p_1)\mu_1}{s+\lambda+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(0,m-l-1,l,l)}(s)\\
&+\left(\frac{lp_2\mu_2}{s+\lambda+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(0,m-l+1,l-1,l-1)}(s)\\
&+\left(\frac{l(1-p_2)\mu_2}{s+\lambda+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(0,m-l,l-1,l-1)}(s),\end{aligned}$$

$$\text{for}\ 0\le l\le m,\ 1\le m\le M. \tag{14}$$

When $n = 1, 2, 3, \ldots, K-1,\ j < l$,

$$\begin{aligned}
\phi_{(n,m-l,l,j)}(s) = &\left(\frac{\lambda(1-\delta_{mM})}{s+\lambda+(J_2-j)\theta+nv(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(n,m-l+1,l,j)}(s) \\
&+\left(\frac{\lambda\delta_{mM}}{s+\lambda+(J_2-j)\theta+nv(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(n+1,m-l,l,j)}(s) \\
&+\left(\frac{(J_2-j)\theta}{s+\lambda+(J_2-j)\theta+nv(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(n,m-l,l,j+1)}(s) \\
&+\left(\frac{nv(1-\delta_{mM})}{s+\lambda+(J_2-j)\theta+nv(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(n-1,m-l+1,l,j)}(s) \\
&+\left(\frac{r_{m-l}p_1\mu_1}{s+\lambda+(J_2-j)\theta+nv(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(n,m-l-1,l+1,j)}(s) \\
&+\left(\frac{r_{m-l}(1-p_1)\mu_1}{s+\lambda+(J_2-j)\theta+nv(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(n,m-l-1,l,j)}(s) \\
&+\left(\frac{jpp_2\mu_2}{s+\lambda+(J_2-j)\theta+nv(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(n,m-l+1,l-1,j-1)}(s) \\
&+\left(\frac{j(1-p)p_2\mu_2}{s+\lambda+(J_2-j)\theta+nv(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(n,m-l+1,l-1,j)}(s) \\
&+\left(\frac{jp(1-p_2)\mu_2}{s+\lambda+(J_2-j)\theta+nv(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(n,m-l,l-1,j-1)}(s) \\
&+\left(\frac{j(1-p)(1-p_2)\mu_2}{s+\lambda+(J_2-j)\theta+nv(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(n,m-l,l-1,j)}(s),
\end{aligned}$$

$$\text{for } 0 \le j \le s_l,\ 0 \le l \le m,\ 0 \le m \le M. \quad (15)$$

When $n = 1, 2, \ldots, K-1,\ j = l$,

$$\begin{aligned}
\phi_{(n,m-l,l,l)}(s) = &\left(\frac{\lambda(1-\delta_{mM})}{s+\lambda+nv(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(n,m-l+1,l,l)}(s) \\
&+\left(\frac{\lambda\delta_{mM}}{s+\lambda+nv(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(n+1,m-l,l,l)}(s) \\
&+\left(\frac{nv(1-\delta_{mM})}{s+\lambda+nv(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(n-1,m-l+1,l,l)}(s) \\
&+\left(\frac{r_{m-l}p_1\mu_1}{s+\lambda+nv(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(n,m-l-1,l+1,l)}(s) \\
&+\left(\frac{r_{m-l}(1-p_1)\mu_1}{s+\lambda+nv(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(n,m-l-1,l,l)}(s) \\
&+\left(\frac{lp_2\mu_2}{s+\lambda+nv(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(n,m-l+1,l-1,l-1)}(s) \\
&+\left(\frac{l(1-p_2)\mu_2}{s+\lambda+nv(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(n,m-l,l-1,l-1)}(s),
\end{aligned}$$

$$\text{for } 0 \le l \le m,\ 0 \le m \le M. \quad (16)$$

When $n = K,\ j < l$,

$$\begin{aligned}
\phi_{(K,m-l,l,j)}(s) &= \left(\frac{\lambda(1-\delta_{mM})}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(K,m-l+1,l,j)}(s)\\
&+\left(\frac{K\nu(1-\delta_{mM})}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(K-1,m-l+1,l,j)}(s)\\
&+\left(\frac{(J_2-j)\theta}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(K,m-l,l,j+1)}(s)\\
&+\left(\frac{r_{m-l}p_1\mu_1}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(K,m-l-1,l+1,j)}(s)\\
&+\left(\frac{r_{m-l}(1-p_1)\mu_1}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(K,m-l-1,l,j)}(s)\\
&+\left(\frac{jpp_2\mu_2}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(K,m-l+1,l-1,j-1)}(s)\\
&+\left(\frac{j(1-p)p_2\mu_2}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(K,m-l+1,l-1,j)}(s)\\
&+\left(\frac{jp(1-p_2)\mu_2}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(K,m-l,l-1,j-1)}(s)\\
&+\left(\frac{j(1-p)(1-p_2)\mu_2}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+j\mu_2}\right)\phi_{(K,m-l,l-1,j)}(s),
\end{aligned}$$

$$\text{for } 0 \le j \le s_l,\ 0 \le l \le m,\ 0 \le m \le M. \quad (17)$$

When $n = K,\ j = l$,

$$\begin{aligned}
\phi_{(K,m-l,l,l)}(s) &= \left(\frac{\lambda(1-\delta_{mM})}{s+\lambda(1-\delta_{mM})+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(K,m-l+1,l,l)}(s)\\
&+\left(\frac{K\nu(1-\delta_{mM})}{s+\lambda(1-\delta_{mM})+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(K-1,m-l+1,l,l)}(s)\\
&+\left(\frac{r_{m-l}p_1\mu_1}{s+\lambda(1-\delta_{mM})+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(K,m-l-1,l+1,l)}(s)\\
&+\left(\frac{r_{m-l}(1-p_1)\mu_1}{s+\lambda(1-\delta_{mM})+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(K,m-l-1,l,l)}(s)\\
&+\left(\frac{lp_2\mu_2}{s+\lambda(1-\delta_{mM})+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(K,m-l+1,l-1,l-1)}(s)\\
&+\left(\frac{l(1-p_2)\mu_2}{s+\lambda(1-\delta_{mM})+K\nu(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2}\right)\phi_{(K,m-l,l-1,l-1)}(s)
\end{aligned}$$

$$\text{for } 0 \le l \le m,\ 0 \le m \le M. \quad (18)$$

After routine block identification, we can express the system of equations (13)-(18) in matrix form as in (11).

As the busy period of our retrial queueing system starts by visiting the state $(0,1,0,0)$, the first component of the vector $\mathbf{0} = \{(0, m-l, l, j);\ 0 \le j \le s_l,\ 0 \le l \le m,\ 1 \le m \le M\}$, we define the unconditional version of LST as

$$L(s) = [1,0,0,\dots,0]_{1\times(\Lambda_M-1)}\phi(s). \tag{19}$$

Let $f_L(x)$ denote the unconditional density associated with $L(s)$. Owing to the Tauberian result, the value of $f_L(x)$ at $x = 0$ is

$$f_L(0) = \lim_{s\to\infty} sL(s).$$

Since

$$\lim_{s\to\infty} s\phi_{(n,m-l,l,j)}(s) = \begin{cases} \mu_1(1-p_1) & \text{if } (n,m-l,l,j) = (0,1,0,0) \\ \mu_2(1-p_2) & \text{if } (n,m-l,l,j) = (0,0,1,1) \\ 0, & otherwise, \end{cases}$$

we have

$$f_L(0) = \lim_{s\to\infty} sL(s) = [1,0,0,\dots,0]_{1\times(\Lambda_M-1)} \lim_{s\to\infty} s\,\phi(s) = \mu_1(1-p_1).$$

We focus our attention on computing the k^{th} order moment $\boldsymbol{m}_{(n,m-l,l,j)}(k) = E\left[T^k_{(n,m-l,l,j)}\right]$ of $\boldsymbol{T}_{(n,m-l,l,j)}(s)$ for $k = 0,1,2,\dots$. Differentiating (11) k-times with respect to s on both sides, we get

$$\boldsymbol{T}_L(s)\frac{d^k}{ds^k}\boldsymbol{\phi}(s) - k\frac{d^{k-1}}{ds^{k-1}}\boldsymbol{\phi}(s) = 0 \quad \text{for } k = 1,2,3,\dots. \tag{20}$$

Let $\overline{\boldsymbol{m}}(k)$ be the vector of moments partitioned in accordance with the orbit levels. In the partitioned form, we have

$$\overline{\boldsymbol{m}}_0(k) = [\overline{\boldsymbol{m}}_{0_1}(k), \overline{\boldsymbol{m}}_{0_2}(k), \dots, \overline{\boldsymbol{m}}_{0_m}(k), \dots, \overline{\boldsymbol{m}}_{0_M}(k)]^T_{(\Gamma_M-1)\times 1} \quad \text{for } k = 0,1,2,\dots,$$

where

$$\begin{aligned}\overline{\boldsymbol{m}}_{0_m}(k) = \ & [\overline{m}_{(0,m,0,0)}(k), \overline{m}_{(0,m-1,1,0)}(k), \overline{m}_{(0,m-1,1,1)}(k), \overline{m}_{(0,m-2,2,0)}(k), \overline{m}_{(0,m-2,2,1)}(k), \\ & \overline{m}_{(0,m-2,2,2)}(k), \dots, \overline{m}_{(0,1,m-1,0)}(k), \overline{m}_{(0,1,m-1,1)}(k), \dots, \overline{m}_{(0,1,m-1,s_{m-1})}(k), \\ & \overline{m}_{(0,0,m,0)}(k), \overline{m}_{(0,0,m,1)}(k), \dots, \overline{m}_{(0,0,m,s_m)}(k)]^T_{\gamma_m\times 1} \quad \text{for } m = 1,2,3,\dots,M.\end{aligned}$$

When $n = 1,2,3,\dots,K$,

$$\overline{\boldsymbol{m}}_n(k) = [\overline{\boldsymbol{m}}_{n_0}(k), \overline{\boldsymbol{m}}_{n_1}(k), \dots, \overline{\boldsymbol{m}}_{n_m}(k), \dots, \overline{\boldsymbol{m}}_{n_M}(k)]^T_{\Gamma_M\times 1}$$

where

$$\begin{aligned}\overline{\boldsymbol{m}}_{n_m}(k) = \ & [\overline{m}_{(n,m,0,0)}(k), \overline{m}_{(n,m-1,1,0)}(k), \overline{m}_{(n,m-1,1,1)}(k), \overline{m}_{(n,m-2,2,0)}(k), \overline{m}_{(n,m-2,2,1)}(k), \\ & \overline{m}_{(n,m-2,2,2)}(k), \dots, \overline{m}_{(n,1,m-1,0)}(k), \overline{m}_{(n,1,m-1,1)}(k), \dots, \overline{m}_{(n,1,m-1,s_{m-1})}(k), \\ & \overline{m}_{(n,0,m,0)}(k), \overline{m}_{(n,0,m,1)}(k), \dots, \overline{m}_{(n,0,m,s_m)}(k)]^T_{\gamma_m\times 1}, \quad m = 0,1,2,\dots,M,\end{aligned}$$

and for $k = 0,1,2,\dots$,

$$\overline{\boldsymbol{m}}(k) = [\overline{\boldsymbol{m}}_0(k), \overline{\boldsymbol{m}}_1(k), \dots, \overline{\boldsymbol{m}}_K(k)]^T_{(\Lambda_M-1)\times 1}.$$

Making use of $\overline{\boldsymbol{m}}(k) = (-1)^k \frac{d^k}{ds^k}\phi(s)|_{s=0}$ and $T_L(0) = \bar{Q}$ in (20) yields

$$\bar{Q}\,\overline{\boldsymbol{m}}(k) + k\,\overline{\boldsymbol{m}}(k-1) = 0 \ \text{ for } k = 1,2,3,\dots,$$

with $\overline{\boldsymbol{m}}(0) = [1,1,1,\dots,1]^T_{(\Lambda_M-1)\times 1}$.

Thus, the k^{th} order conditional moments, $\overline{\boldsymbol{m}}(k)$, $k = 1,2,3,\dots$, of the busy period satisfy the following recurrence block tri-diagonal system:

$$\bar{Q}\overline{\boldsymbol{m}}(k) = -k\overline{\boldsymbol{m}}(k-1), \quad k = 1,2,3,\dots, \tag{21}$$

whence

$$\overline{\boldsymbol{m}}(k) = -k(\bar{Q})^{-1}\overline{\boldsymbol{m}}(k-1)\ ,\ k = 1,2,3,\dots\ .$$

Hence the unconditional k^{th} order moments are obtained as

$$E(L^k) = [1,0,0,\ldots,0]_{1\times(\Lambda_M-1)}\bar{\boldsymbol{m}}(k),\ k = 1,2,3,\ldots\ . \tag{22}$$

The first two unconditional moments of the busy period of our retrial queueing network system are obtained as

$$E(L) = \bar{\boldsymbol{m}}(1) = -[1,0,0,\ldots,0]_{1\times(\Lambda_M-1)}(\bar{Q})^{-1}\ [1,1,1,\ldots,1]_{(\Lambda_M-1)\times 1},$$

and

$$E(L^2) = [1,0,0,\ldots,0]_{1\times(\Lambda_M-1)}\bar{\boldsymbol{m}}(2)$$
$$= 2[1,0,0,\ldots,0](\bar{Q})^{-2}\ [1,1,\ldots,1]^T_{\Lambda_M-1\times 1}$$

where we have used $\bar{\boldsymbol{m}}(2) = 2\ (\bar{Q})^{-2}\ [1,1,\ldots,1]^T_{\Lambda_M-1\times 1}$.

Thus the variance, $Var(L)$, of the busy period L is obtained as $Var(L) = E(L^2) - (E(L))^2$.

We now demonstrate a variety of numerical examples to illustrate the trend of $E(L)$ and $Var(L)$. In Figures 6 and 7, respectively, $E(L)$ and $Var(L)$ are sketched versus λ for different values of p and fixed parametric values $\nu = 5$, $\theta = 5$, $\mu_1 = 9$, $\mu_2 = 8$, $p_1 = p_2 = 0.5$, $J_1 = 2$, $J_2 = 3$, $M = 6$ and $K = 10$. Figures 6 and 7 show that both $E(L)$ and $Var(L)$ are increasing functions of λ. On the other hand, for $\lambda = 7$, $\theta = 5$, $\mu_1 = 9$, $\mu_2 = 8$, $p_1 = p_2 = 0.5$, $J_1 = 2$, $J_2 = 3$, $M = 6$ and $K = 10$, the opposite trends are seen in both $E(L)$ and $Var(L)$ versus ν as displayed, respectively, in Figures 8 and 9. In the case of non-vacation system, it is evident that, both $E(L)$ and $Var(L)$ are seen to be less than any of the vacation service systems under discussion.

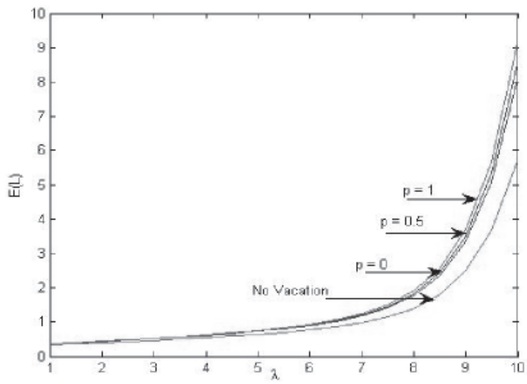

Figure 6. $E(L)$ versus λ for ν =5, θ =5, M=6, K=10, p_1=p_2=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

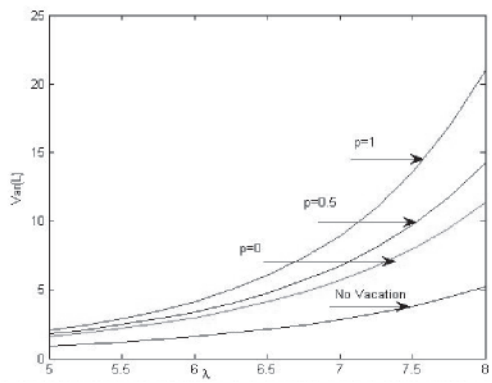

Figure 7. $Var(L)$ versus λ for ν =5, θ =5, M=6, K=10, p_1=p_2=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

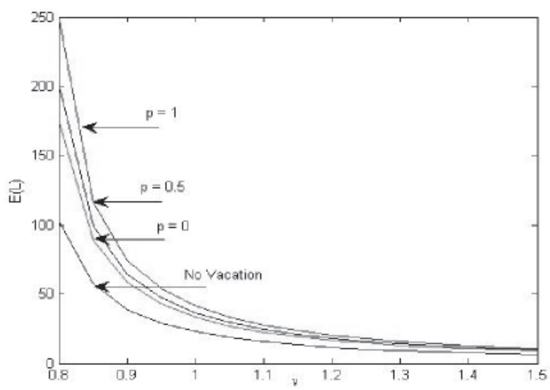

Figure 8. $E(L)$ versus ν for λ =7, θ =5, M=6, K=10, p_1=p_2=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

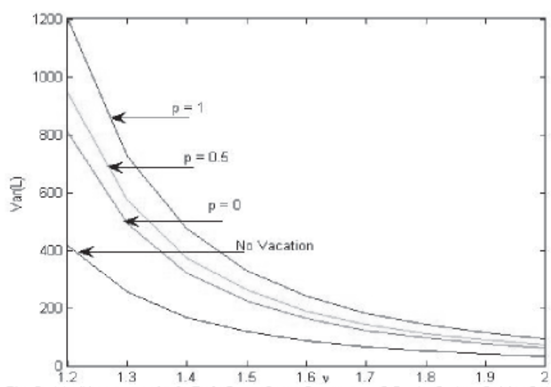

Figure 9. $Var(L)$ versus ν for λ =7, θ =5, M=6, K=10, p_1=p_2=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

Now we show, in Figures 10 and 11, the effects of λ and ν on $E(L)$ for varying orbit size $K = 5, 10$ and 15. From Figure 10, it is clear that $E(L)$ increases for increasing values of λ and K. The trend of $E(L)$ versus ν is depicted in Figure 11 and we notice that the curves corresponding to $K = 5, 10$ and 15 are graphically indistinguishable in the displayed domain. However, $E(L)$ decreases sharply before attaining its equilibrium for large values of ν. We have also observed from our numerical experience, though it is not being reported here, that the trends of $Var(L)$ versus λ, and ν, respectively, behave very similar to $E(L)$ against λ and ν for $K = 5, 10$ and 15. We depict, in Figures 12 and 13, the trends of $E(L)$ and $Var(L)$ versus θ for $\lambda = 7$, $\nu = 5$, $\mu_1 = 9$, $\mu_2 = 8$, $p_1 = p_2 = p = 0.5$, $J_1 = 2$, $J_2 = 3$ and $M = 6$ for varying orbit size $K = 5, 10$ and 15. As is evident from the figures, all three curves corresponding to $K = 5, 10$ and 15 are decreasing sharply before attaining their equilibrium for large values of θ.

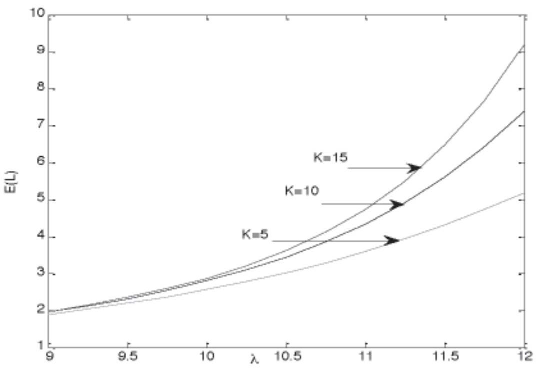

Figure 10. $E(L)$ versus λ for $\nu = 5$, $\theta = 5$, $M=6$, $p_1 = p_2 = p = 0.5$, $\mu_1 = 9$, $\mu_2 = 8$, $J_1 = 2$, and $J_2 = 3$.

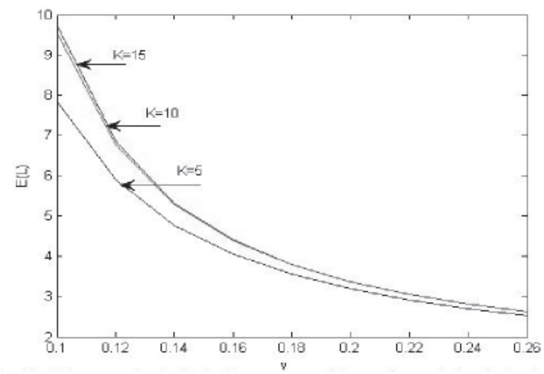

Figure 11. $E(L)$ versus ν for $\lambda = 7$, $\theta = 5$, $M=6$, $p_1 = p_2 = p = 0.5$, $\mu_1 = 9$, $\mu_2 = 8$, $J_1 = 2$, and $J_2 = 3$.

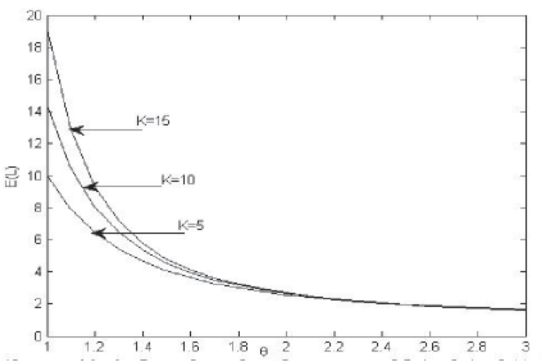

Figure 12. $E(L)$ versus θ for $\lambda = 7$, $\nu = 5$, $M=6$, $p_1 = p_2 = p = 0.5$, $\mu_1 = 9$, $\mu_2 = 8$, $J_1 = 2$, and $J_2 = 3$.

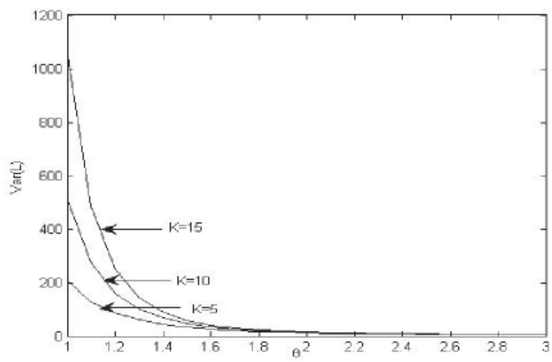

Figure 13. $Var(L)$ versus θ for $\lambda = 7$, $\nu = 5$, $M=6$, $p_1 = p_2 = p = 0.5$, $\mu_1 = 9$, $\mu_2 = 8$. $J_1 = 2$, and $J_2 = 3$.

Next, we display the behaviour of the mean, $E(L)$, and the variance, $Var(L)$, of the busy period for the system under discussion in Figures 14-17 for several choices of the number of servers (J_1, J_2) of I/O and CPU queues with orbit capacity $K = 10$. Figures 14 and 16 exhibit that both $E(L)$ and $Var(L)$ increase with increasing values of λ as is to be expected. In Figures 15 and 17, it can be observed that both measures $E(L)$ and $Var(L)$ decrease for increasing values of ν.

Finally, we have also observed from our numerical experience that the trends of two descriptors $E(L)$ and $Var(L)$ versus λ, θ and ν for three different levels of the service rates $(\mu_1,\ \mu_2)$ behave very similar to $E(L)$ and $Var(L)$ for the several choice of the number of servers (J_1, J_2) of I/O and CPU queues with orbit capacity $K = 10$.

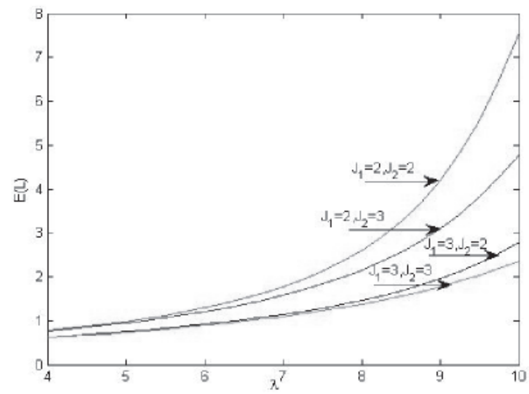

Figure 14. $E(L)$ versus λ for ν =5, θ =5, M=6, K=10, p_1=p_2=p=0.5, μ_1=9, and μ_2=8.

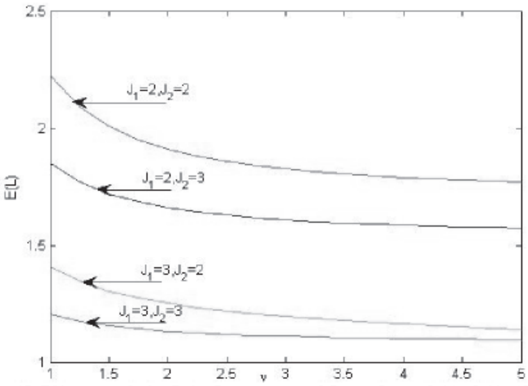

Figure 15. $E(L)$ versus ν for λ =7, θ =5, M=6, K=10, p_1=p_2=p=0.5, μ_1=9, and μ_2=8.

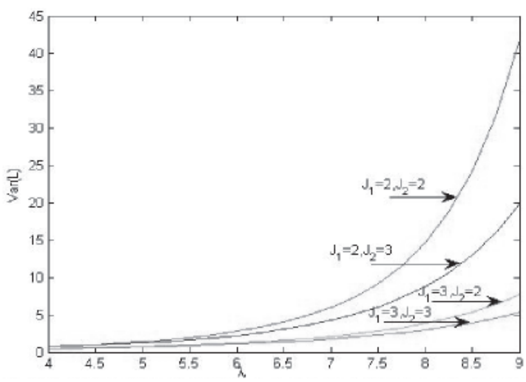

Figure 16. $Var(L)$ versus λ for ν =5, θ =5, M=6, K=10, p_1=p_2=p=0.5, μ_1=9, and μ_2=8.

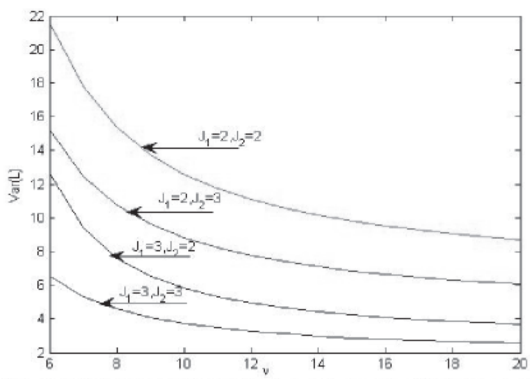

Figure 17. $Var(L)$ versus ν for λ =7, θ =5, M=6, K=10, p_1=p_2=p=0.5, μ_1=9, and μ_2=8.

7. Waiting Time Analysis

In this section, we analyze the waiting time of a tagged program (packets) for our retrial queueing system. The waiting time W is defined as the sojourn time of a tagged program in the finite orbit/retrial group. In the classical retrial queues, it is typically assumed that the programs in the retrial group operate under a random order policy. This assumption makes the analysis intractable because we have to consider not only the system state at the arrival epoch of the tagged program, but also the possibility that the programs arriving at a later time will compete for entering into the inner multiprocessor network. In general, the stationary distribution of W is analytically intractable. However, we develop an algorithmic procedure for the recursive computation of the LSTs and moments of the waiting time, W, of a tagged orbit program for the FOC retrial queueing network system.

In our model, it is clear that if the number of programs in the inner multiprocessor network is less than M, an arriving primary program enters into the I/O queue. On the other hand, if the number of programs in the inner multiprocessor network is equal to M and the number of programs in the orbit is less than K, an arriving primary program joins the group of orbiting programs. In order to study the waiting time of the tagged program in the orbit, we must observe the time duration starting from the arrival epoch of the tagged program upto the time when it enters into the inner multiprocessor network.

First, we derive the LST of the conditional waiting time distribution of the tagged program given the system state and the status of the tagged program immediately on its arrival. For this purpose, we introduce the following definitions and notations:

$W_{(n,m-l,l,j)}$: the residual waiting time of the tagged program given that the system is in state

$$(n, m-l, l, j), 1 \leq n \leq K.$$

$\widetilde{W}_{(n,m-l,l,j)}(s) = E[e^{-sW_{(n,m-l,l,j)}}]$, $Re(s) \geq 0$, $\forall\ (n, m-l, l, j) \in \Omega$: the conditional LST of the random variable $W_{(n,m-l,l,j)}$ given that $(n, m-l, l, j) \in \Omega$.

$\widehat{W}(s) = E(e^{-sW})$, $Re(s) \geq 0$: the LST of the random variable W.

The following vectors comprise the above transforms according to the orbit levels:

For $n = 1, 2, 3, \ldots, K$,

$$\widetilde{\boldsymbol{W}}_n(s) = [\widetilde{W}_{(n,0,0,0)}(s), \widetilde{W}_{(n,1,0,0)}(s), \widetilde{W}_{(n,0,1,0)}(s), \widetilde{W}_{(n,0,1,1)}(s), \ldots, \widetilde{W}_{(n,0,M,s_M)}(s)]^T_{\Gamma_M \times 1}$$

and

$$\widetilde{\boldsymbol{W}}(s) = [\widetilde{\boldsymbol{W}}_1(s), \widetilde{\boldsymbol{W}}_2(s), \ldots, \widetilde{\boldsymbol{W}}_n(s), \ldots, \widetilde{\boldsymbol{W}}_K(s)]^T_{K\Gamma_M \times 1}.$$

The following theorem gives a system of linear equations for the LSTs $\widetilde{W}_{(n,m-l,l,j)}(s)$.

Theorem 2. *The LSTs* $\{\widetilde{W}_{(n,m-l,l,j)}(s);\ (n, m-l, l, j) \in \Omega,\ 1 \leq n \leq K\}$ *satisfy the following block tri-diagonal system:*

$$\boldsymbol{T}_W(s)\boldsymbol{W}(s) = \boldsymbol{\Delta}_{K\Gamma_M \times 1} \tag{23}$$

where

$$\boldsymbol{T}_W(s) = \widehat{\boldsymbol{Q}} - s\boldsymbol{I}_{K\Gamma_M \times K\Gamma_M} \tag{24}$$

$$\boldsymbol{\Delta}_{K\Gamma_M \times 1} = -\nu[(\underbrace{1, 1, \ldots, 1}_{\Gamma_{M-1}}, \underbrace{0, 0, \ldots, 0}_{\gamma_M}), 2(\underbrace{1, 1, \ldots, 1}_{\Gamma_{M-1}}, \underbrace{0, 0, \ldots, 0}_{\gamma_M}), \ldots, K(\underbrace{1, 1, \ldots, 1}_{\Gamma_{M-1}}, \underbrace{0, 0, \ldots, 0}_{\gamma_M})]^T_{K\Gamma_M \times 1}.$$

with

$$\widehat{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{A}_{1,1} & \boldsymbol{A}_{0,1} & & & & \\ \widehat{\boldsymbol{A}}_{2,2} & \boldsymbol{A}_{1,2} & \boldsymbol{A}_{0,2} & & & \\ & \widehat{\boldsymbol{A}}_{2,3} & \boldsymbol{A}_{1,3} & \boldsymbol{A}_{0,3} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \widehat{\boldsymbol{A}}_{2,K-1} & \boldsymbol{A}_{1,K-1} & \boldsymbol{A}_{0,K-1} \\ & & & & \widehat{\boldsymbol{A}}_{2,K} & \boldsymbol{A}_{0,K} + \boldsymbol{A}_{1,K} \end{bmatrix}_{K\Gamma_M \times K\Gamma_M} \tag{25}$$

in which $\widehat{\boldsymbol{A}}_{2,n} = \frac{(n-1)}{n}\boldsymbol{A}_{2,n}$ *for* $n = 2, 3, 4, \ldots, K$,

Proof. Employing the first-step analysis, we obtain the following system of equations:

When $n = 1, 2, \ldots, K-1,\ j < l$,

$$\widetilde{W}_{(n,m-l,l,j)}(s) = \left(\frac{\lambda(1-\delta_{mM})}{s + \lambda + (J_2 - j)\theta + r_{m-l}\mu_1 + j\mu_2 + n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n,m-l+1,l,j)}(s)$$

$$+\left(\frac{\lambda\delta_{mM}}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n+1,m-l,l,j)}(s)$$
$$+\left(\frac{(J_2-j)\theta}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n,m-l,l,j+1)}(s)$$
$$+\left(\frac{r_{m-l}p_1\mu_1}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n,m-l-1,l+1,j)}(s)$$
$$+\left(\frac{r_{m-l}(1-p_1)\mu_1}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n,m-l-1,l,j)}(s)$$
$$+\left(\frac{jpp_2\mu_2}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n,m-l+1,l-1,j-1)}(s)$$
$$+\left(\frac{j(1-p)p_2\mu_2}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n,m-l+1,l-1,j)}(s)$$
$$+\left(\frac{jp(1-p_2)\mu_2}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n,m-l,l-1,j-1)}(s)$$
$$+\left(\frac{j(1-p)(1-p_2)\mu_2}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n,m-l,l-1,j)}(s)$$
$$+\left(\frac{(n-1)\nu(1-\delta_{mM})}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n-1,m-l+1,l,j)}(s)$$
$$+\left(\frac{\nu(1-\delta_{mM})}{s+\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})}\right)$$
$$\text{for } 0\le j\le s_l,\ 0\le l\le m,\ 0\le m\le M. \quad (26)$$

When $n=1,2,3,\dots,K-1,\ j=l$,

$$\widetilde{W}_{(n,m-l,l,l)}(s)=\left(\frac{\lambda(1-\delta_{mM})}{s+\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n,m-l+1,l,l)}(s)$$
$$+\left(\frac{\lambda\delta_{mM}}{s+\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n+1,m-l,l,l)}(s)$$
$$+\left(\frac{r_{m-l}p_1\mu_1}{s+\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n,m-l-1,l+1,l)}(s)$$
$$+\left(\frac{r_{m-l}(1-p_1)\mu_1}{s+\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n,m-l-1,l,l)}(s)$$
$$+\left(\frac{lp_2\mu_2}{s+\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n,m-l+1,l-1,l-1)}(s)$$
$$+\left(\frac{l(1-p_2)\mu_2}{s+\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n,m-l,l-1,l-1)}(s)$$
$$+\left(\frac{(n-1)\nu(1-\delta_{mM})}{s+\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})}\right)\widetilde{W}_{(n-1,m-l+1,l,l)}(s)$$

$$+\left(\frac{\nu(1-\delta_{mM})}{s+\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})}\right)$$

$$\text{for } 0 \le l \le m,\ 0 \le m \le M. \qquad (27)$$

When $n = K, j < l$,

$$\widetilde{W}_{(K,m-l,l,j)}(s) = \left(\frac{\lambda(1-\delta_{mM})}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K,m-l+1,l,j)}(s)$$
$$+\left(\frac{(J_2-j)\theta}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K,m-l,l,j+1)}(s)$$
$$+\left(\frac{r_{m-l}p_1\mu_1}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K,m-l-1,l+1,j)}(s)$$
$$+\left(\frac{r_{m-l}(1-p_1)\mu_1}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K,m-l-1,l,j)}(s)$$
$$+\left(\frac{jpp_2\mu_2}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K,m-l+1,l-1,j-1)}(s)$$
$$+\left(\frac{j(1-p)p_2\mu_2}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K,m-l+1,l-1,j)}(s)$$
$$+\left(\frac{jp(1-p_2)\mu_2}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K,m-l,l-1,j-1)}(s)$$
$$+\left(\frac{j(1-p)\mu_2(1-p_2)}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K,m-l,l-1,j)}(s)$$
$$+\left(\frac{(K-1)\nu(1-\delta_{mM})}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K-1,m-l+1,l,j)}(s)$$
$$+\left(\frac{\nu(1-\delta_{mM})}{s+\lambda(1-\delta_{mM})+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+K\nu(1-\delta_{mM})}\right)$$

$$\text{for } 0 \le j \le s_l,\ 0 \le l \le m,\ m \le M. \qquad (28)$$

When $n = K,\ j = l$,

$$\widetilde{W}_{(K,m-l,l,l)}(s) = \left(\frac{\lambda(1-\delta_{mM})}{s+\lambda(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K,m-l+1,l,l)}(s)$$
$$+\left(\frac{r_{m-l}p_1\mu_1}{s+\lambda(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K,m-l-1,l+1,l)}(s)$$
$$+\left(\frac{r_{m-l}(1-p_1)\mu_1}{s+\lambda(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K,m-l-1,l,l)}(s)$$
$$+\left(\frac{lp_2\mu_2}{s+\lambda(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K,m-l+1,l-1,l-1)}(s)$$
$$+\left(\frac{l(1-p_2)\mu_2}{s+\lambda(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K,m-l,l-1,l-1)}(s)$$

$$+\left(\frac{(K-1)\nu(1-\delta_{mM})}{s+\lambda(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2+K\nu(1-\delta_{mM})}\right)\widetilde{W}_{(K-1,m-l+1,l,l)}(s)$$
$$+\left(\frac{\nu(1-\delta_{mM})}{s+\lambda(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2+K\nu(1-\delta_{mM})}\right)$$
$$\text{for} \quad 0\le l\le m,\ 0\le m\le M. \qquad (29)$$

The contribution of repeated attempts made by programs from the orbit is explained as follows: In (26)-(29), the last terms on the right-hand side correspond to the case where the tagged program from the orbit should immediately enter into the inner multiprocessor network. In contrast, if any non-tagged program from the orbit applies for entry into the inner multiprocessor network and succeeds, we obtain the last but one term on the right-hand side according to the random order re-attempt. By expressing the system of equations (26)-(29) in matrix form, we obtain the expression (23).

Note that, if upon arrival, the tagged primary program should join the orbital programs and wait in the orbit if it finds the system at any state in the set $E_W=\{(n,M-l,l,j);\ 0\le n\le K-1, 0\le j\le s_l,$ $0\le l\le M\}$. The steady state probability vector $\mathbf{\Pi}_W$ of dimension $1\times K\Gamma_M$ is defined as

$$\mathbf{\Pi}_W=[\mathbf{\Pi}_{W_0},\mathbf{\Pi}_{W_1},\dots,\mathbf{\Pi}_{W_n},\dots,\mathbf{\Pi}_{W_{K-1}}]_{1\times K\Gamma_M}, \qquad (30)$$

where

$$\mathbf{\Pi}_{W_n}=[\underbrace{0,0,\dots,0}_{\Gamma_{M-1}},\underbrace{Y(n,M,0,0),Y(n,M-1,1,0),Y(n,M-1,1,1),\dots,Y(n,0,M,s_M)}_{\gamma_M}]_{1\times\Gamma_M},$$

for $n=0,1,2,\dots,K-1$.

According to the PASTA property, we obtain the unconditional version of the LST of the waiting time W as

$$\widehat{W}(s)=1-\mathbf{\Pi}_W e_W+\mathbf{\Pi}_W\widetilde{W}(s), \qquad (31)$$

where $\boldsymbol{e}_W=[1,1,1,\dots,1]^T_{K\Gamma_M\times 1}$

It is noticed that the contribution

$$P(W=0)=1-\mathbf{\Pi}_W e_W,$$

represents the steady-state probability of no waiting time which occurs either when the arriving tagged primary program with rate λ finds the number of programs in the inner multiprocessor network is less than M or when it finds the number of programs in the system (i.e., in both orbit and inner multiprocessor network) is $K+M$ and becomes a lost program. The contribution $\Pi_W\widetilde{W}(s)$ represents the LST of the waiting time of the tagged program with density function $w_c(t)$ given $W>0$ such that

$$\int_{0+}^{\infty} w_c(t)dt=\mathbf{\Pi}_W e_W^T.$$

We have that

$$\lim_{s\to\infty} s\widetilde{W}_{(n,m-l,l,j)}(s)=\begin{cases}\nu & \text{if } 1\le n\le K, \quad 0\le j\le s_l, \quad 0\le l\le m, \quad 0\le m\le M-1.\\ 0, & otherwise.\end{cases}$$

We now focus on the moments of the waiting time, W, of the tagged program in the orbit. To this end, let $\overline{W}_{(n,m-l,l,j)}(k)=E[W^{(k)}_{(n,m-l,l,j)}]$, be the k^{th} order moment of $W_{(n,m-l,l,j)}$, $k=0,1,2,\dots$, and $\overline{W}(k)=E(W^k)$ be the k^{th} order moments of W.

The following vectors comprise the above moments partitioned according to the orbit levels:

For $k=0,1,2,\dots$,

$$\overline{\boldsymbol{W}}_n(k)=[\overline{W}_{(n,0,0,0)}(k),\overline{W}_{(n,1,0,0)}(k),\overline{W}_{(n,0,1,0)}(k),\overline{W}_{(n,0,1,1)}(k),\dots,\overline{W}_{(n,0,M,s_M)}(k)]^T_{\Gamma_M\times 1}$$

and

$$\overline{\boldsymbol{W}}(k)=[\overline{\boldsymbol{W}}_1(k),\overline{\boldsymbol{W}}_2(k),\dots,\overline{\boldsymbol{W}}_n(k),\dots,\overline{\boldsymbol{W}}_K(k)]^T_{K\Gamma_M\times 1}.$$

Differentiating the expression (23) k-times with respect to s, we find that

$$\boldsymbol{T}_W(s)\frac{d^k}{ds^k}\widetilde{\boldsymbol{W}}(s)-k\frac{d^{k-1}}{ds^{k-1}}\widetilde{\boldsymbol{W}}(s)=0.$$

Making use of $\overline{\boldsymbol{W}}(k)=(-1)^k\frac{d^k}{ds^k}\widetilde{\boldsymbol{W}}(s)|_{s=0}$ and $\boldsymbol{T}_W(0)=\widehat{\boldsymbol{Q}}$ in the above yields the conditional k^{th} order vector of moments $\overline{\boldsymbol{W}}(k)$, $k=0,1,2,...$, of the waiting time of the tagged program in the orbit as the solution of the following block tri-diagonal system:

$$\widehat{Q}\overline{\boldsymbol{W}}(k)=-k\overline{\boldsymbol{W}}(k-1)\ \text{ for }\ k=1,2,3,\dots, \tag{32}$$

with

$$\overline{\boldsymbol{W}}(0)=[1,1,1,\dots,1]^T_{K\Gamma_M\times 1}.$$

The unconditional k^{th} order moments $E(W^k)=\widehat{W}(k)$, $k=1,2,3,\dots$, of the waiting time of the tagged program are determined as

$$\widehat{W}(k)=E(W^k)=\boldsymbol{\Pi}_W\overline{\boldsymbol{W}}(k),\ k=0,1,2,\dots. \tag{33}$$

In particular, the first two unconditional moments of waiting time of the tagged program for FOC can be obtained by using the block recurrent system (32) and the probability vector (30) as

$$E(W)=\boldsymbol{\Pi}_W\overline{\boldsymbol{W}}(1)$$
$$E(W)=-\boldsymbol{\Pi}_W(\widehat{\boldsymbol{Q}})^{-1}[1,1,1,\dots,]^T_{K\Gamma_M\times 1}$$
$$E(W^2)=\boldsymbol{\Pi}_W\overline{W}(2)=2\boldsymbol{\Pi}_W(\widehat{\boldsymbol{Q}})^{-2}[1,1,1,\dots,]^T_{K\Gamma_M\times 1}.$$

Finally, the variance, $Var(W)$, of the waiting time W is obtained as $Var(W)=E(W^2)-E(W)^2$.

We now examine the impact of the primary program arrival rate λ, the vacation rate θ, the retrial rate ν, the vacation parameter p, orbit capacity K and the number of service channels (J_1, J_2) on two measures $E(W)$ and $Var(W)$. In all our numerical examples, we take $p_1=p_2=0.5$, $\mu_1=9, \mu_2=8$ and $M=6$. Figures 18 and 19, respectively, reveal the trends of $E(W)$ and $Var(W)$ against λ for $p=0,0.5,1$, $\nu=5$, $\theta=5$, $J_1=2$, $J_2=3$ and $K=10$. It is observed that both $E(W)$ and $Var(W)$ initially increase rapidly and then start decreasing monotonically with increasing values of λ. Moreover, both measures $E(W)$ and $Var(W)$ demonstrate a surprising phenomenon of possessing a maximum as a function of λ for all three vacation scheduling service systems and non-vacation system. Such a phenomenon is also noticed in Falin and Artalejo [15], Almasi et al. [3] and Dragieva [12] for non-vacation retrial queueing systems. Besides, it can be seen that both descriptors $E(W)$ and $Var(W)$ possess the highest maximum in the case of 1-limited service vacation system.

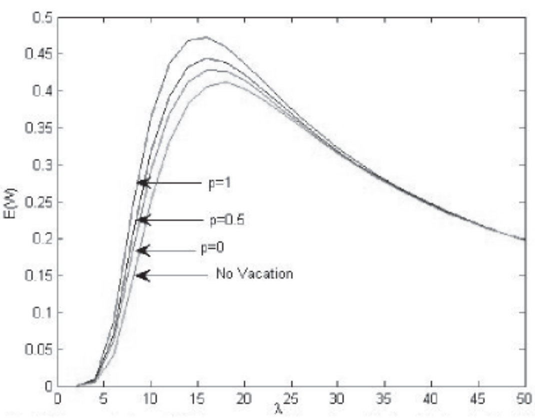

Figure 18. $E(W)$ versus λ for $\nu=5$, $\theta=5$, M=6, K=10, p_1=p_2=0.5, μ_1=9, μ_2=8. J_1=2, and J_2=3.

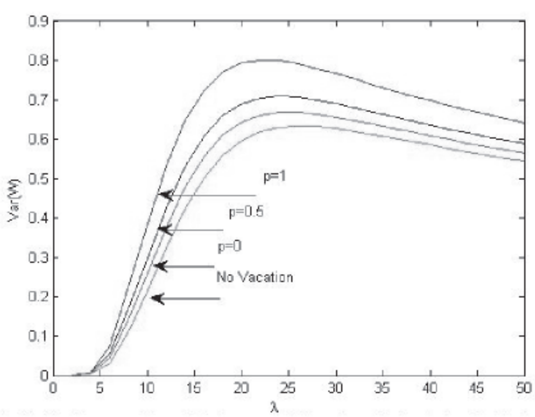

Figure 19. $Var(W)$ versus λ for $\nu=5$, $\theta=5$, M=6, K=10, p_1=p_2=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

Next, we depict $E(W)$ and $Var(W)$, respectively, in Figures 20 and 21 as a function of ν. These figures show the effect of not only the retrial rate ν but also the vacation parameter p. For $\lambda = 7$, $\mu_1 = 9$, $\mu_2 = 8$, $\theta = 5$, $p_1 = p_2 = 0.5$, $p = 0$, 0.5, 1, $J_1 = 2$, $J_2 = 3$, $M = 6$ and $K = 10$, the figures reveal that $E(W)$ and $Var(W)$ are monotonically decreasing rapidly to zero as ν tends to a large value. However, for fixed value of ν, the measures $E(W)$ and $Var(W)$ increase when the value of vacation parameter p increases. Further, for the non-vacation system, both $E(W)$ and $Var(W)$ have lower values than any of the vacation service systems under investigation.

We analyze the behaviour of $E(W)$ and $Var(W)$ for $p = p_1 = p_2 = 0.5$, $\mu_1 = 9$, $\mu_2 = 8$, $\nu = 5$, $\theta = 5$, $J_1 = 2$, $J_2 = 3$, $M = 6$ and three different orbit capacity $K = 5$, 10 and 15. In Figures 22 and 23, we plot $E(W)$ and $Var(W)$ versus λ. As before, the figures display the same surprising phenomenon of possessing a maximum as a function of λ. Further, it is also noticed that both $E(W)$ and $Var(W)$ possess the highest maximum for the largest orbit capacity. Figures 24 and 25 reveal the trend of $E(W)$ and $Var(W)$ versus θ for $p = p_1 = p_2 = 0.5$, $\lambda = 7, \nu = 5$, $\mu_1 = 9$, $\mu_2 = 8$, $J_1 = 2$, $J_2 = 3$, $M = 6$ and for three different orbit capacity $K = 5, 10$ and 15. It is seen from the figures that both $E(W)$ and $Var(W)$ are decreasing for increasing values of θ as is to be expected. However, for fixed value of θ, the descriptors $E(W)$ and $Var(W)$ increase when the orbit capacity K

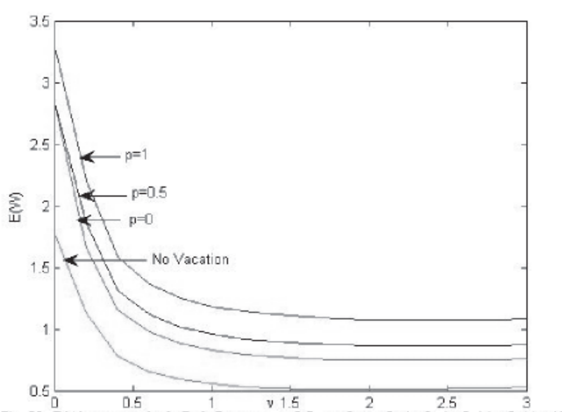

Figure 20. $E(W)$ versus ν for $\lambda = 7$, $\theta = 5$, $M=6$, $K=10$, $p_1=p_2=0.5$, $\mu_1=9$, $\mu_2=8$, $J_1=2$, and $J_2=3$.

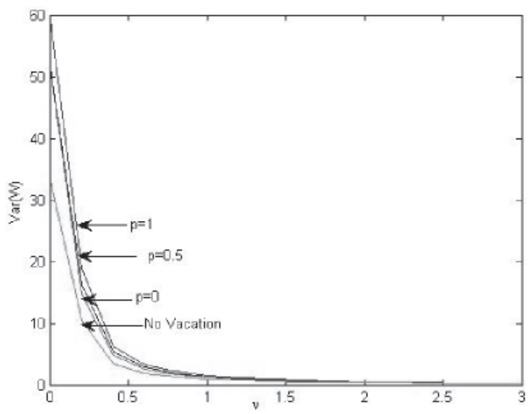

Figure 21. $Var(W)$ versus ν for $\lambda = 7$, $\theta = 5$, $M=6$, $K=10$, $p_1=p_2=0.5$, $\mu_1=9$, $\mu_2=8$, $J_1=2$, and $J_2=3$.

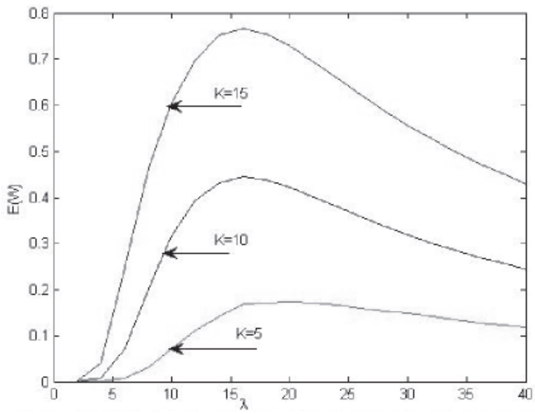

Figure 22. $E(W)$ versus λ for $\nu = 5$, $\theta = 5$, $M=6$, $p_1=p_2=p=0.5$, $\mu_1=9$, $\mu_2=8$, $J_1=2$, and $J_2=3$.

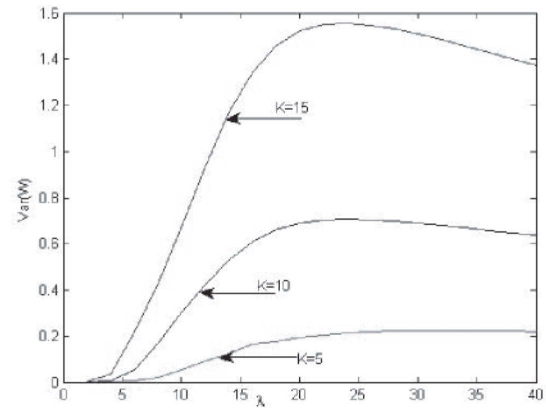

Figure 23. $Var(W)$ versus λ for $\nu = 5$, $\theta = 5$, $M=6$, $p_1=p_2=p=0.5$, $\mu_1=9$, $\mu_2=8$, $J_1=2$, and $J_2=3$.

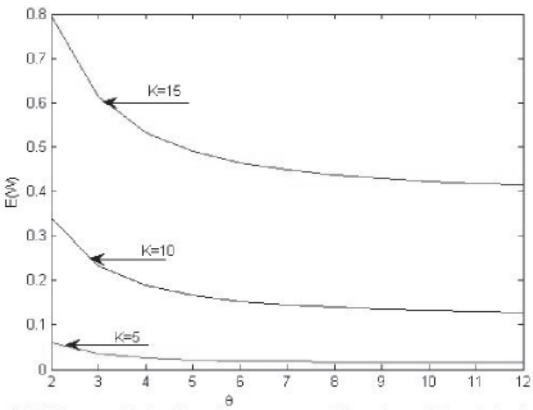

Figure 24. $E(W)$ versus θ for λ =7, ν =5, M=6, p_1 =p_2=p=0.5, μ_1=9, μ_2= 8, J_1=2, and J_2=3.

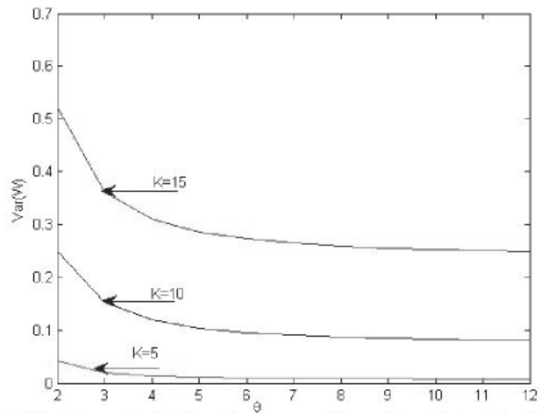

Figure 25. $Var(W)$ versus θ for λ =7, ν =5, M=6, p_1 =p_2=p=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

increases. We have also examined the behaviour of $E(W)$ and $Var(W)$ against ν for $p = p_1 = p_2 = 0.5$, $\lambda = 7$, $\theta = 5$, $\mu_1 = 9$, $\mu_2 = 8$, $J_1 = 2$, $J_2 = 3$, $M = 6$ and $K = 5, 10$ and 15. From our numerical experience, not reported here, similar conclusions can be inferred for both descriptors, i.e., $E(W)$ and $Var(W)$ decrease with increasing retrial rate ν but increase with increasing values of K.

Finally, we investigate the behaviour of $E(W)$ and $Var(W)$ against λ, θ and ν for different combinations of the number of service channels (J_1, J_2) in the I/O and CPU queues. By taking $\nu = 5$, $\theta = 5$, $p = p_1 = p_2 = 0.5$, $\mu_1 = 9$, $\mu_2 = 8$, $M = 6$ and $K = 10$, We depict $E(W)$ and $Var(W)$, respectively, in Figures 26 and 27 as a function of λ for different combinations of the number of service channels (J_1, J_2) in the I/O and CPU queue. From the figures, we infer that both measures $E(W)$ and $Var(W)$ attain a maximum as a function of λ. Further, it is interesting to observe that both $E(W)$ and $Var(W)$ possess the highest maximum when the total number of service channels (J_1, J_2) are small in the I/O and CPU queues. We also examined the behaviour of $E(W)$ and $Var(W)$ versus θ. Our numerical results, though it is not being presented here, indicated that both descriptors $E(W)$ and $Var(W)$ decrease before attaining their limiting values for increasing values of θ and for fixed θ, they decrease when we increase the total number of service channels (J_1, J_2) in the I/O and CPU queues. Similar behaviours can also be noticed for $E(W)$ and $Var(W)$ versus ν for different combinations of the number of service channels (J_1, J_2) in the I/O and CPU queues.

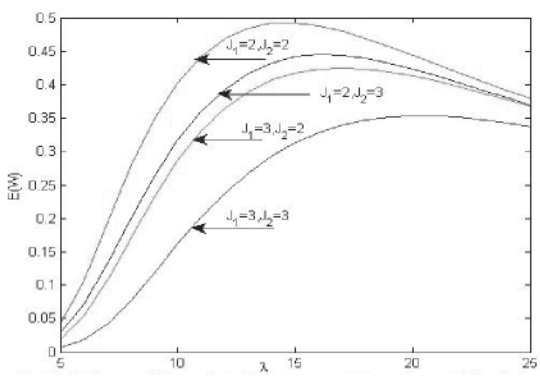

Figure 26. $E(W)$ versus λ for ν =5, θ =5, M=6, K=10, p_1=p_2=p=0.5, μ_1=9, and μ_2=8.

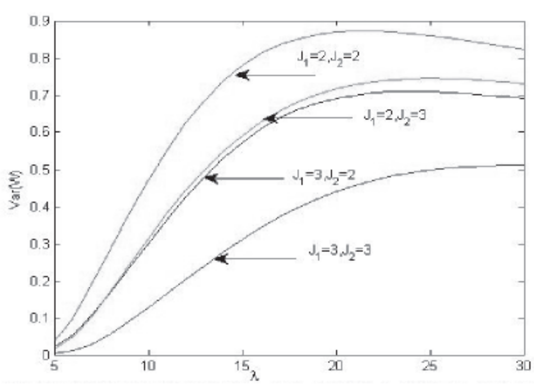

Figure 27. $Var(W)$ versus λ for ν=5, θ =5, M=6, K=10, p_1=p_2=p =0.5, μ_1=9, and μ_2=8.

8. Number of Retrials

In this section, we consider the random variable, ψ, of the number of retrials made by the tagged program from the orbit until it enters into the inner multiprocessor network. This descriptor is the discrete counter part of the waiting time, W, of a tagged program studied in the previous section and complements the present analysis. The number of retrials ψ is an important quantity in itself because in real multiprogramming computer system, it determines the additional load on the control devices. In this sense both W and ψ are excellent performance measures for the multiprogramming-multiprocessor retrial queueing system. In order to discuss the number of retrials made by a tagged program for our queueing network system, we introduce the following definitions and notations:

$\psi_{(n,m-l,l,j)}$: the number of retrials that a tagged program will make from the orbit given that the system state is $(n,m-l,l,j)\in\Omega$, $1\le n\le K$.

$\tilde{\psi}_{(n,m-l,l,j)}(z)=E[z^{\psi_{(n,m-l,l,j)}}]=\sum_{k=0}^{\infty}P(\psi_{(n,m-l,l,j)}=k)\ z^k, |z|\le 1,$

: the conditional probability generating function of the random variable $\psi_{(n,m-l,l,j)}$.

$\hat{\psi}(z)$=$E(z^{\psi})$, $|z|\le 1$: the probability generating functions of the random variable ψ.

Next we introduce the vectors which comprise the above generating functions partitioned according to the orbit levels : For $n=1,2,3,\dots,K$,

$$\tilde{\boldsymbol{\psi}}_n(z)=[\tilde{\psi}_{(n,0,0,0)}(z),\tilde{\psi}_{(n,1,0,0)}(z),\tilde{\psi}_{(n,0,1,0)}(z),\tilde{\psi}_{(n,0,1,1)}(z),\dots,\tilde{\psi}_{(n,0,M,s_M)}(z)]^T_{\Gamma_M\times 1}$$

and

$$\tilde{\boldsymbol{\psi}}(z)=[\tilde{\boldsymbol{\psi}}_1(z),\tilde{\boldsymbol{\psi}}_2(z),\tilde{\boldsymbol{\psi}}_3(z),\dots,\tilde{\boldsymbol{\psi}}_K(z)]^T_{K\Gamma_M\times 1}.$$

The following theorem establishes a relationship for the generating functions $\tilde{\psi}_{(n,m-l,l,j)}(z)$:

Theorem 3. *The generating functions* $\{\tilde{\psi}_{(n,m-l,l,j)}(z);\ (n,m-l,l,j)\ \in\ \Omega,\ 1\le n\le K,$ $0\le j\le s_l,\ 0\le l\le m,\ 0\le m\le M\}$ *satisfy the following block recurrent system:*

$$\boldsymbol{T}_{\psi}(z)\ \tilde{\boldsymbol{\psi}}(z)=\boldsymbol{\Delta}(z) \tag{34}$$

where

$$\boldsymbol{T}_{\psi}(z)=\widehat{\boldsymbol{Q}}+(1-z)\boldsymbol{V},\quad \boldsymbol{\Delta}(z)=z\boldsymbol{\Delta}_{K\Gamma_M\times 1} \tag{35}$$

with

$$\boldsymbol{V}\ =\ -\nu\ diag[\boldsymbol{I}_{01},\boldsymbol{I}_{02},\boldsymbol{I}_{03},\dots,\boldsymbol{I}_{0K}]_{K\Gamma_M\times K\Gamma_M} \tag{36}$$

in which

$$\mathbf{I}_{on}=n\begin{bmatrix}\mathbf{0}_{\Gamma_{M-1}\times\Gamma_{M-1}} & \mathbf{0}_{\Gamma_{M-1}\times\gamma_M}\\ \mathbf{0}_{\gamma_M\times\Gamma_{M-1}} & \boldsymbol{I}_{\gamma_M\times\gamma_M}\end{bmatrix}_{\Gamma_M\times\Gamma_M}\quad for\ \ n=1,2,\dots,K.$$

Proof. Adopting the first-step argument, we have the following system of equations for $\tilde{\psi}_{(n,m-l,l,j)}(z)$:

When $n=1,2,3,\dots,K-1,\ j<l,$

$$\begin{aligned}\tilde{\psi}_{(n,m-l,l,j)}(z)=&\left(\frac{\lambda(1-\delta_{mM})}{\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l+1,l,j)}(z)\\ &+\left(\frac{\lambda\delta_{mM}}{\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n+1,m-l,l,j)}(z)\\ &+\left(\frac{(J_2-j)\theta}{\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l,l,j+1)}(z)\\ &+\left(\frac{r_{m-l}p_1\mu_1}{\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l-1,l+1,j)}(z)\end{aligned}$$

$$+\left(\frac{r_{m-l}(1-p_1)\mu_1}{\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l-1,l,j)}(z)$$

$$+\left(\frac{jpp_2\mu_2}{\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l+1,l-1,j-1)}(z)$$

$$+\left(\frac{j(1-p)p_2\mu_2}{\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l+1,l-1,j)}(z)$$

$$+\left(\frac{jp(1-p_2)\mu_2}{\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l,l-1,j-1)}(z)$$

$$+\left(\frac{j(1-p)(1-p_2)\mu_2}{\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l,l-1,j)}(z)$$

$$+\left(\frac{(n-1)\nu(1-\delta_{mM})}{\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n-1,m-l+1,l,j)}(z)$$

$$+\left(\frac{z\nu(1-\delta_{mM})}{\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)$$

$$+\left(\frac{z\nu\delta_{mM}}{\lambda+(J_2-j)\theta+r_{m-l}\mu_1+j\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l,l,j)}(z)$$

$$\text{for } 0\le j\le s_l,\ 0\le l\le m,\ 0\le m\le M. \tag{37}$$

When $n=1,2,\dots,K-1,\ j=l$,

$$\tilde{\psi}_{(n,m-l,l,l)}(z)=\left(\frac{\lambda(1-\delta_{mM})}{\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l+1,l,l)}(z)$$

$$+\left(\frac{\lambda\delta_{mM}}{\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n+1,m-l,l,l)}(z)$$

$$+\left(\frac{r_{m-l}p_1\mu_1}{\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l-1,l+1,l)}(z)$$

$$+\left(\frac{r_{m-l}(1-p_1)\mu_1}{\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l-1,l,l)}(z)$$

$$+\left(\frac{lp_2\mu_2}{\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l+1,l-1,l-1)}(z)$$

$$+\left(\frac{l(1-p_2)\mu_2}{\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l,l-1,l-1)}(z)$$

$$+\left(\frac{(n-1)\nu(1-\delta_{mM})}{\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n-1,m-l+1,l,l)}(z)$$

$$+\left(\frac{z\nu(1-\delta_{mM})}{\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)$$

$$+\left(\frac{z\nu\delta_{mM}}{\lambda+r_{m-l}\mu_1+l\mu_2+n\nu(1-\delta_{mM})+\nu\delta_{mM}}\right)\tilde{\psi}_{(n,m-l,l,l)}(z)$$

$$\text{for } 0\le l\le m,\ 0\le m\le M. \tag{38}$$

When $n = K,\ j < l$,

$$\begin{aligned}
\tilde{\psi}_{(k,m-l,l,j)}(z) = &\left(\frac{\lambda(1-\delta_{mM})}{\lambda(1-\delta_{mM}) + (J_2 - j)\theta + r_{m-l}\mu_1 + j\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K,m-l+1,l,j)}(z) \\
&+\left(\frac{(J_2 - j)\theta}{\lambda(1-\delta_{mM}) + (J_2 - j)\theta + r_{m-l}\mu_1 + j\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K,m-l,l,j+1)}(z) \\
&+\left(\frac{r_{m-l}p_1\mu_1}{\lambda(1-\delta_{mM}) + (J_2 - j)\theta + r_{m-l}\mu_1 + j\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K,m-l-1,l+1,j)}(z) \\
&+\left(\frac{r_{m-l}(1-p_1)\mu_1}{\lambda(1-\delta_{mM}) + (J_2 - j)\theta + r_{m-l}\mu_1 + j\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K,m-l-1,l,j)}(z) \\
&+\left(\frac{jpp_2\mu_2}{\lambda(1-\delta_{mM}) + (J_2 - j)\theta + r_{m-l}\mu_1 + j\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K,m-l+1,l-1,j-1)}(z) \\
&+\left(\frac{j(1-p)p_2\mu_2}{\lambda(1-\delta_{mM}) + (J_2 - j)\theta + r_{m-l}\mu_1 + j\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K,m-l+1,l-1,j)}(z) \\
&+\left(\frac{jp(1-p_2)\mu_2}{\lambda(1-\delta_{mM}) + (J_2 - j)\theta + r_{m-l}\mu_1 + j\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K,m-l,l-1,j-1)}(z) \\
&+\left(\frac{j(1-p)(1-p_2)\mu_2}{\lambda(1-\delta_{mM}) + (J_2 - j)\theta + r_{m-l}\mu_1 + j\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K,m-l,l-1,j)}(z) \\
&+\left(\frac{(K-1)\nu(1-\delta_{mM})}{\lambda(1-\delta_{mM}) + (J_2 - j)\theta + r_{m-l}\mu_1 + j\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K-1,m-l+1,l,j)}(z) \\
&+\left(\frac{z\nu(1-\delta_{mM})}{\lambda(1-\delta_{mM}) + (J_2 - j)\theta + r_{m-l}\mu_1 + j\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right) \\
&+\left(\frac{z\nu\delta_{mM}}{\lambda(1-\delta_{mM}) + (J_2 - j)\theta + r_{m-l}\mu_1 + j\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K,m-l,l,j)}(z)
\end{aligned}$$

$$\text{for } 0 \le j \le s_l,\ 0 \le l \le m,\ 0 \le m \le M. \tag{39}$$

When $n = K,\ j = l$,

$$\begin{aligned}
\tilde{\psi}_{(k,m-l,l,l)}(z) = &\left(\frac{\lambda(1-\delta_{mM})}{\lambda(1-\delta_{mM}) + r_{m-l}\mu_1 + l\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K,m-l+1,l,l)}(z) \\
&+\left(\frac{r_{m-l}p_1\mu_1}{\lambda(1-\delta_{mM}) + r_{m-l}\mu_1 + l\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K,m-l-1,l+1,l)}(z) \\
&+\left(\frac{r_{m-l}(1-p_1)\mu_1}{\lambda(1-\delta_{mM}) + r_{m-l}\mu_1 + l\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K,m-l-1,l,l)}(z) \\
&+\left(\frac{lp_2\mu_2}{\lambda(1-\delta_{mM}) + r_{m-l}\mu_1 + l\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K,m-l+1,l-1,l-1)}(z) \\
&+\left(\frac{l(1-p_2)\mu_2}{\lambda(1-\delta_{mM}) + r_{m-l}\mu_1 + l\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K,m-l,l-1,l-1)}(z) \\
&+\left(\frac{(K-1)\nu(1-\delta_{mM})}{\lambda(1-\delta_{mM}) + r_{m-l}\mu_1 + l\mu_2 + K\nu(1-\delta_{mM}) + \nu\delta_{mM}}\right)\tilde{\psi}_{(K-1,m-l+1,l,l)}(z)
\end{aligned}$$

$$+\left(\frac{zv(1-\delta_{mM})}{\lambda(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2+Kv(1-\delta_{mM})+v\delta_{mM}}\right)$$
$$+\left(\frac{zv\delta_{mM}}{\lambda(1-\delta_{mM})+r_{m-l}\mu_1+l\mu_2+Kv(1-\delta_{mM})+v\delta_{mM}}\right)\tilde{\psi}_{(K,m-l,l,l)}(z)$$
$$\text{for } 0 \le l \le m,\ 0 \le m \le M. \qquad (40)$$

In the above equations (37)-(40), the last three terms of the right-hand side correspond to retrials made by the orbital programs. The incidence of a retrial may be made either by a non-tagged program or by the tagged program. Due to the random order repeated attempt, the terms containing $(n-1)v(1-\delta_{mM})$ and $zv(1-\delta_{mM})$, respectively, correspond to the retrials made by the non-tagged program and by the tagged program, provided the number of programs in the inner multiprocessor network is less than its maximum capacity M. On the other hand, if the inner multiprocessor network is fully occupied by M programs, the retrials made by the orbit programs will be vain retrials. Hence the vain retrial made by a non-tagged program neither affects the event under study nor modifies the current system state, but we have to count the vain retrials made by the tagged program. Thus the term with $zv\delta_{mM}$ corresponds to the vain retrial made by the tagged program.

It is to be noted that by setting $s=0$ in (26)-(29), the left-hand sides of waiting time equations agree with (37)-(40) except the term containing $-v(1-z)$ which should be added due to vain retrials at the main diagonal elements of $\widehat{\boldsymbol{Q}}$ when the inner multiprocessor network is fully occupied by M programs. After the routine matrix formulation, the system of equations (37)-(40) leads to (34).

The unconditional version of the probability generating function $\tilde{\psi}(z)$ of the number of repeated attempts made by the tagged program from the orbit is given as

$$\hat{\psi}(z) = 1-\boldsymbol{\Pi}_W e_{K\Gamma_M} + \boldsymbol{\Pi}_W \widetilde{\boldsymbol{\psi}}(z),$$

where $1-\boldsymbol{\Pi}_W \boldsymbol{e}_{K\Gamma_M}$ represents the probability that the tagged program upon arrival enters immediately into the inner multiprocessor network.

We now compute recursively, the k^{th} order factorial moment of the random variable ψ, the number of retrials made by the tagged program.

Let $\overline{\psi}_{(n,m-l,l,j)}(k) \;=\; E[\psi_{(n,m-l,l,j)}\ (\psi_{(n,m-l,l,j)}-1)\cdots(\psi_{(n,m-l,l,j)}-k+1)\,],\ k=1,2,3,\ldots,$

be the k^{th} factorial moment of the random variable $\psi_{(n,m-l,l,j)}$ and $\tilde{\psi}_{(n,m-l,l,j)}(0)=1$.

The moments are partitioned according to the orbit levels as, $k=0,1,2,\ldots,$

$$\overline{\boldsymbol{\psi}}_n(k) = [\overline{\psi}_{(n,0,0,0)},\overline{\psi}_{(n,1,0,0)},\overline{\psi}_{(n,0,1,0)},\overline{\psi}_{(n,0,1,1)},\ldots,\overline{\psi}_{(n,0,M,S_M)}]^T_{\Gamma_M\times 1},\ 1\le n\le K,$$

and

$$\overline{\boldsymbol{\psi}}(k) \;= [\overline{\boldsymbol{\psi}}_1(k),\overline{\boldsymbol{\psi}}_2(k),\ldots,\overline{\boldsymbol{\psi}}_n(k),\ldots,\overline{\boldsymbol{\psi}}_K(k)]^T_{K\Gamma_M\times 1}.$$

Differentiating the expression (34) k-times with respect to z, we get

$$\boldsymbol{T}_\psi(z)\frac{d^k}{dz^k}\widetilde{\boldsymbol{\psi}}(z) - k\,\boldsymbol{V}\frac{d^{k-1}}{dx^{k-1}}\widetilde{\boldsymbol{\psi}}(z) = \delta_{k1}\boldsymbol{\Delta}_{K\Gamma_M\times 1}.$$

By setting $z=1$ and making use of $\overline{\boldsymbol{\psi}}(k) = \frac{d^k}{dz^k}\widetilde{\boldsymbol{\psi}}(z)|_{z=1}$, $k=1,2,3,\ldots,$ in the above equation, we get the following block recurrent system:

$$\widehat{\boldsymbol{Q}}\overline{\boldsymbol{\psi}}(k) = \delta_{k1}\boldsymbol{\Delta}_{K\Gamma_M\times 1} + k\,\boldsymbol{V}\,\overline{\boldsymbol{\psi}}(k-1),\ \text{with } \overline{\boldsymbol{\psi}}(0) = \boldsymbol{e}_{K\Gamma_M}\ \text{for } k=1,2,3,\ldots. \qquad (41)$$

Hence, the unconditional k^{th} order factorial moments $E[\psi\ (\psi-1)(\psi-2)\cdots(\psi-k+1)]$ of the random variable ψ are determined as

$$E[\psi(\psi-1)(\psi-2)\cdots(\psi-k+1)] = \boldsymbol{\Pi}_W\overline{\boldsymbol{\psi}}(k),\ k=1,2,3,\ldots, \qquad (42)$$

where $\boldsymbol{\Pi}_W$ is given in (30).

In particular, by using the block recurrent relation (41) and the probability (30), the first two unconditional moments, $E(\psi)$, and, $E(\psi^2)$, of the number of retrials made by the tagged program can be obtained as

$$E(\psi) = \mathbf{\Pi}_W \overline{\boldsymbol{\psi}}(1) = \mathbf{\Pi}_W (\widehat{\boldsymbol{Q}})^{-1} [\boldsymbol{\Delta}_{K\Gamma_M \times 1} + \boldsymbol{V}\, \boldsymbol{e}_{K\Gamma_M}]$$

and

$$E(\psi^2) = \mathbf{\Pi}_W [I + 2(\widehat{\boldsymbol{Q}})^{-1} \boldsymbol{V}] \widetilde{\boldsymbol{\psi}}(1).$$

We now present the numerical results of the mean, $E(\psi)$, and the variance, $Var(\psi)$, of the number of retrials ψ made by the tagged program from the orbit. For all our numerical investigation, we have fixed $p_1 = p_2 = 0.5$, $\mu_1 = 9$, $\mu_2 = 8$ and $M = 6$. Figures 28 and 29, respectively, demonstrate the behaviour of $E(\psi)$ and $Var(\psi)$ versus λ for $p = 0,\ 0.5,\ 1$, $\nu = 5$, $\theta = 5$, $J_1 = 2$, $J_2 = 3$, and $K = 10$. Here also, the figures exhibit a similar surprising phenomenon of our retrial queueing network system attaining a maximum of $E(\psi)$ and $Var(\psi)$ as a function of λ as noticed earlier in the analysis of waiting time. Moreover, it is observed that both measures $E(\psi)$ and $Var(\psi)$ possess the highest maximum in the case of 1-limited service vacation system.

We next plot $E(\psi)$ and $Var(\psi)$, respectively, in Figures 30 and 31 as a function of ν for $\lambda = 7$, $\theta = 5$, $p = 0,\ 0.5, 1$, $J_1 = 2$, $J_2 = 3$, and $K = 10$. As a result, both descriptors $E(\psi)$ and $Var(\psi)$ increase monotonically with increasing values of ν and the curve corresponding to non-vacation system is lower than the curves of the exhaustive, Bernoulli scheduling and 1-limited service vacation systems.

Figures 32 and 33 report numerical examples, respectively, to show the influence of λ on $E(\psi)$ and $Var(\psi)$ for $\nu = 5$, $\theta = 5$, $p = 0.5$, $J_1 = 2$, $J_2 = 3$, and for three different orbit capacities $K = 5,\ 10,\ 15$. As in the case of waiting time analysis, the maximum attainment of $E(\psi)$ and $Var(\psi)$ as a function of λ are addressed in Figures 32 and 33. Further, our numerical experience, not reported here, indicated that, for fixed λ and θ, the measures $E(\psi)$ and $Var(\psi)$ always increase for increasing values of ν and they increase further when the orbit capacity K increases. This fact is as expected since increase in both ν and K causes more congestion and consequently more reattempts. Next, we study the behaviour of $E(\psi)$ and $Var(\psi)$ as a function of θ by fixing $\lambda = 7$, $\nu = 5$, $\mu_1 = 9$, $\mu_2 = 8$, $p_1 = p_2 = p = 0.5$, $J_1 = 2$, $J_2 = 3$, and M =6 for three different orbit capacity K = 5, 10 and 15. In Figures 34 and 35, it is observed that both descriptors $E(\psi)$ and $Var(\psi)$ decrease for increasing values of θ whereas they increase when the orbit size K increases for fixed values of θ. It can also be seen that $E(\psi)$ and $Var(\psi)$ behave very similar to E(W) and $Var(W)$ against θ for different orbit size K.

Finally, we plot $E(\psi)$ and $Var(\psi)$, respectively, in Figures 36 and 37 as a function of λ for the different combinations of the number of service channels (J_1, J_2) in the I/O and CPU queues. For $\nu = 5$, $\theta = 5$, $p = 0.5$, and $K = 10$, Figures 36 and 37 exhibit that the descriptors $E(\psi)$ and $Var(\psi)$ behave very similar to $E(W)$ and $Var(W)$ versus λ as reported in Figures 26 and 27. We have also performed some numerical illustrations for the descriptors $E(\psi)$ and $Var(\psi)$ as functions of ν and θ for the different combinations of the number of service channels (J_1, J_2) in the I/O and CPU queues. From our numerical results, though it is not being reported here, it can be concluded that the trends of $E(\psi)$ and $Var(\psi)$ are very similar to $E(W)$ and $Var(W)$ versus ν and θ, respectively, as stated in the waiting time analysis.

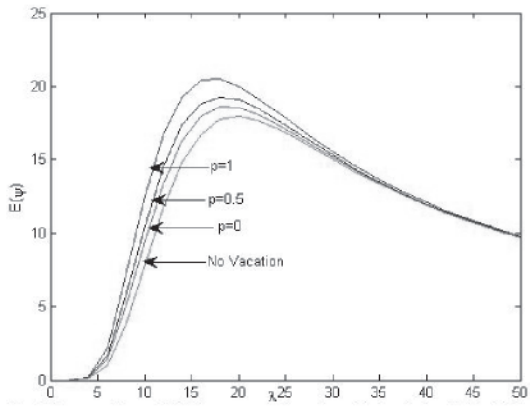

Figure 28. $E(\psi)$ versus λ for ν =5, θ =5, M=6, K=10, p_1=p_2=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

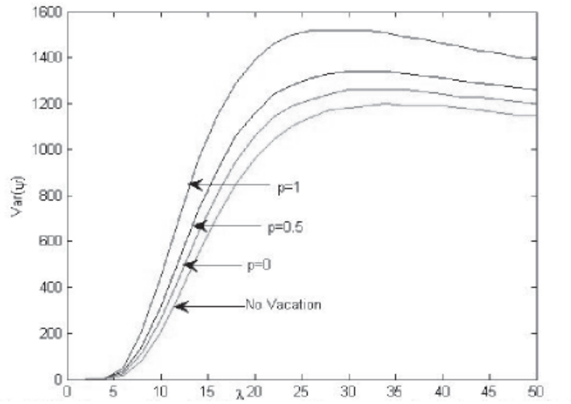

Figure 29. $Var(\psi)$ versus λ for ν =5, θ =5, M=6, K=10, p_1=p_2=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

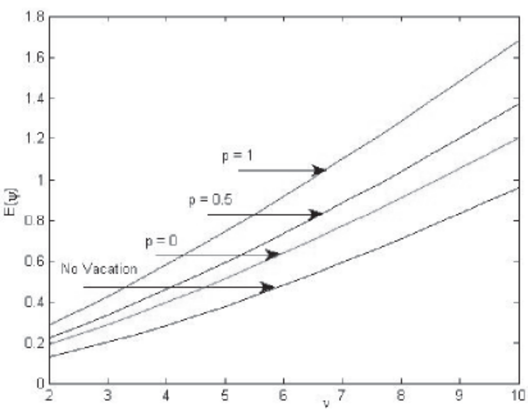

Figure 30. $E(\psi)$ versus ν for λ =7, θ =5, M=6, K=10, p_1=p_2=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

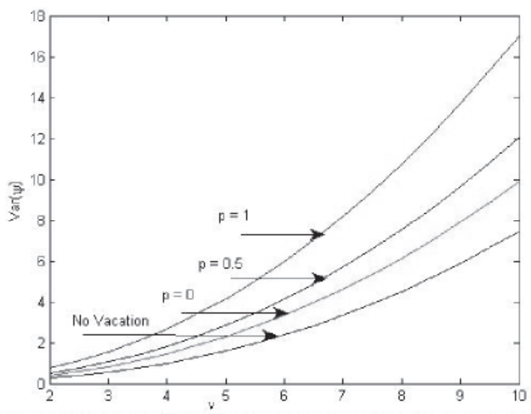

Figure 31. $Var(\psi)$ versus ν for λ =7, θ =5, M=6, K=10, p_1=p_2=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

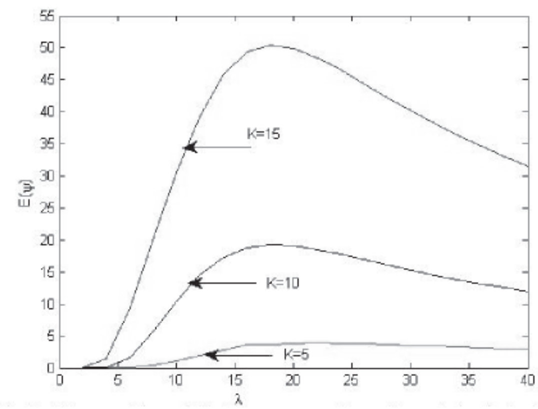

Figure 32. $E(\psi)$ versus λ for ν =5, θ =5, M=6, p_1=p_2 =p=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

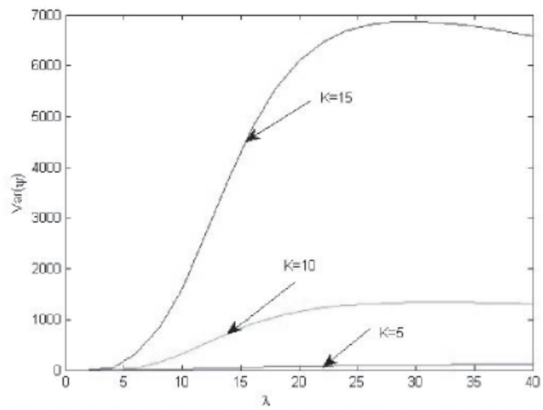

Figure 33. $Var(\psi)$ versus λ for ν =5, θ =5, M=6, p_1=p_2=p=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

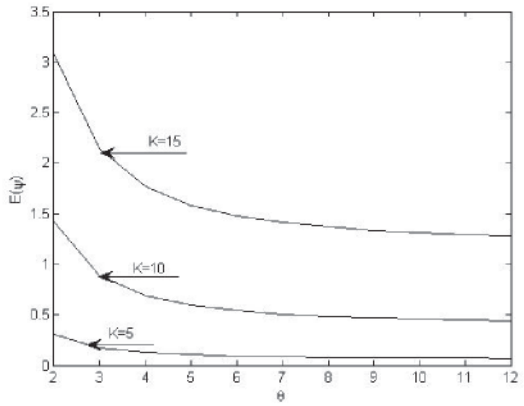

Figure 34. $E(\psi)$ versus θ for λ =7, ν =5, M=6, p_1= p_2 =p=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

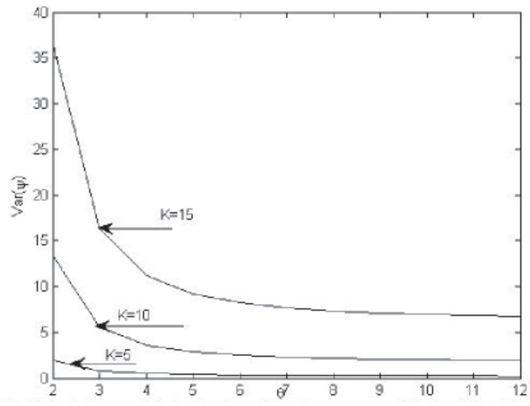

Figure 35. $Var(\psi)$ versus θ for λ =7, ν =5, M=6, p_1=p_2=p=0.5, μ_1=9, μ_2=8, J_1=2, and J_2=3.

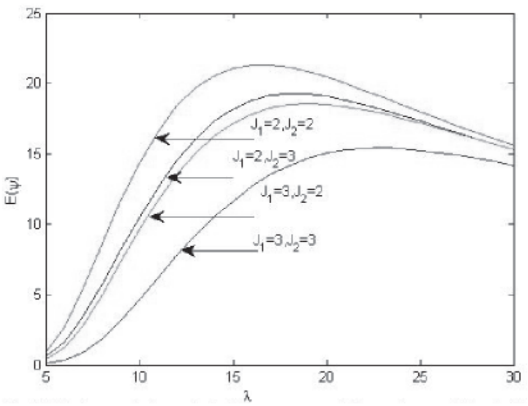

Figure 36. $E(\psi)$ versus λ for ν =5, θ =5, M=6, K=10, p_1=p_2=p=0.5, μ_1=9, and μ_2=8.

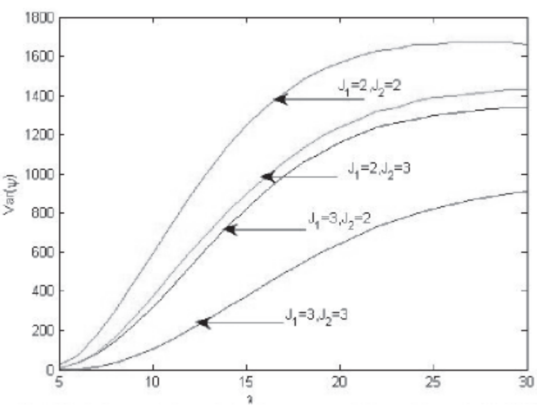

Figure 37. $Var(\psi)$ versus λ for ν =5, θ =5, M=6, K=10, p_1=p_2=p=0.5, μ_1=9, and μ_2=8.

9. Conclusion and Future Work

In this article, we have analyzed numerically a Markovian retrial queue with finite orbit capacity for a multiprogramming-multiprocessor computer network system in which the service channels of CPU queue avail vacations under Bernoulli schedule. The system is formulated as a level-dependent QBD process and the stationary distribution of the number of programs has been obtained. Using the matrix analytical methods, various system performance measures such as the mean and variance of the number of programs in the retrial group, the system busy period, waiting time and the number of retrial made by a tagged program from the orbit are determined. We have also identified the optimal retrial rate for chosen parametric values for specific probability descriptor of the system. Extensive numerical illustrations have been presented to show insight into the performance of the retrial queuing system under discussion. For further work, we plan to study an algorithmic analysis of the classical retrial queue with unlimited orbit capacity for a multiprogramming-multiprocessor computer networks and investigate the ergodic condition of the system. Further, we will also pay our attention to obtain the numerical solution for the stationary distribution of the number of programs in the system using the direct-truncation method. By numerical illustration, the behavior of the various system performance measures and the quantities of interest will be discussed.

Acknowledgments

The authors would like to thank the anonymous reviewers and the guest editors for the careful reading of the manuscript and the valuable suggestions which helped to improve the quality of this paper.

References

[1] Adiri, I., Hofri, M., & Yadin, M.(1973). A multiprogramming queue. *Journal of ACM*, 20(4), 589-603.

[2] Allen, A. O. (2006). Probability, Statistics and Queuing Theory with Computer Applications (2nd ed.). New Delhi, India: Academic Press (An imprint of Elsevier).

[3] Almasi, B., Roszik, J., & Sztrik, J. (2005). Homogeneous finite-source retrial queues with server subject to breakdowns and repairs. *Mathematical and Computer Modelling*, 42, 673-682.

[4] Artalejo, J. R. (2010). Accessible bibliography on retrial queues: Progress in 2000-2009. *Mathematical and Computer Modelling*, 51, 1071-1081.

[5] Artalejo, J. R., & Gomez-Corral, A. (2008). Retrial Queueing Systems: A computational approach. Berlin: Springer.

[6] Artalejo, J. R., Gomez-Corral, A., & Neuts, M. F. (2001). Analysis of multiserver queues with constant retrial rate. *European Journal of Operational Research*, 135,569-581.

[7] Avi-Itzhak, B., & Halfin, S. (1987). Server sharing with a limited number of service positions and symmetric queues. *Journal of Applied Probability*, 24(4), 990-1000.

[8] Avi-Itzhak, B., & Halfin, S. (1988). Response times in $M/M/1$ time-sharing schemes with limited number of service positions. *Journal of Applied Probability*, 25(3), 579-595.

[9] Avi-Itzhak, B., & Heyman, D. P. (1973). Approximate queueing models for multiprogramming computer systems. *Operations Research*, 21, 1212-1230.

[10] Brandwajn, A. (1977). A queueing model of multiprogrammed computer systems under full load conditions. *Journal of ACM*, 24(2), 222-240.

[11] Daduna, H. (1986). Cycle times in two-stage closed queueing networks: Applications to multiprogrammed computer systems with virtual memory. *Operations Research*, 34(2), 281-288.

[12] Dragieva, V. I. (2013). A finite source retrial queues: Number of retrials. *Communications in Statistics-Theory and Method*, 42, 812-829.

[13] Elhafsi, E. H., & Molle, M. (2007). On the solution to QBD processes with finite state space, *Stochastic Analysis and Applications*, 25, 763-779.

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

[15] Falin, G. I., & Artalejo, J. R. (1998). A finite source retrial queues. *European Journal of Operational Research*, 108, 409-424.

[16] Falin, G. I., & Templeton, J. G. C. (1997). Retrial Queues. London: Chapman and Hall.

[17] Gaver, D. P. (1967). Probability models for multiprogramming computer systems. *Journal of ACM*, 14(3), 423-438.

[18] Gaver, D. P., & Humfeld, G. (1976). Multitype multiprogramming models. *Acta Informatica*, 7, 111-121.

[19] Gaver, D. P., Jacobs, P. A., & Latouche, G. (1984). Finite birth and-death models in randomly changing environments, *Advances in Applied probability*, 16: 715-731.

[20] Gaver, D. P., & Shedler, G. S. (1973). Processor utilization inmultiprogramming systems via diffusion approximations. *Operations Research*, 21(2), 569-576.

[21] Gelenbe, E., & Mitrani, I. (2010). Analysis and synthesis of Computer Systems (2nd ed.). London: Imperial College Press.

[22] Hofri, M. (1978). A generating-function analysis of multiprogramming queues. *International Journal of Computer Information Sciences*, 7(2), 121-155.

[23] Kameda, H. (1986). Effects of job loading policies for multiprogramming systems in processing a job stream. *ACM Transactions on Computer Systems*, 4(1), 71-106.

[24] Keilson, J., & Servi, L. D.(1986). Oscillating random walk models for G1/G/1 vacation systems with Bernoulli schedules. *Journal of Applied Probability*, 23, 790–802.

[25] Konheim, A. G., & Reiser, M. (1976). A queuing model with finite waiting room and blocking. *Journal of ACM*, 23(2), 328-341.

[26] Konheim,A. G., & Reiser, M. (1978). Finite capacity queuing systems with applications in computer modeling. *SIAM Journal of Computing*, 7(2), 210-229.

[27] Krishna Kumar, B., & Raja, J. (2006). On multiserver feedback retrial queues with balking and control retrial rate. *Annals of Operations Research*, 141(1), 211-232.

[28] Krishna Kumar, B., Thanikachalam, A., Kanakasabapathi, V., & Rukmani, R. (2016). Performance analysis of a multiprogramming-multiprocessor retrail queueing system with orderly reattempts, *Annals of Operations Research*, 247(1), 319-364.

[29] Kulkarni, V. G., & Liang, H. M. (1997). Retrial queues revisited. In J. H. Dshalalow (Ed.), Frontiers in queueing: Models and applications in science and engineering (pp. 19-34). New York, Boca Raton: CRC Press.

[30] Latouche, G. (1981). Algorithmic analysis of a multiprogramming-multiprocessor computer system. *Journal of ACM*, 28(4), 662-679.

[31] Latouche, G., & Ramaswami, V. (1999). Introduction to Matrix Analytic Method in Stochastic Modelling. Philadelphia: ASA-SIAM.

[32] Lewis, P. V. W., & Shedler, G. S. (1971). A cycle-queue model of system overhead in multiprogrammed computer systems. *Journal of ACM*, 18(2), 199-220.

[33] Rege, K. M., & Sengupta, B. (1985). Sojourn time distribution in a multiprogrammed computer system. *AT & T Tech Journal,* 64, 1077-1090.

[34] Servi, L. D. (2002). Algorithmic solution to two-dimensional birth-death process with application to capacity planning, *Telecommunication Systems,* 21(2), 205-212.