Analysis of Thread Scheduling With Multiple Processors Under A Markov Chain Model

D. Shukla[1] Saurabh Jain[2] and Shweta Ojha[3]

Abstract

This paper presents a Markov chain model based study in the environment of multi-level queue scheduling with the multiple processors, assuming the random movement of scheduler over various processes and queues. Each processor assumes random selection of threads from different queues. In particular, the discussion incorporates only three processors along with three queues and the procedure of thread scheduling is examined in light of Markov chain model. A simulation study is incorporated to support the findings.

Keywords : Process scheduling, Markov chain model, State of system, Process queue, Multi-level queue scheduling, Transition probability matrix, Central Processing Unit (CPU).

1. Introduction

The scheduling is a methodology of managing multiple queues of processes in order to minimize delay and to optimize performance of the system in the environment where queues of processes exist with servers. A scheduler is an OS module whose primary objective is to optimize the system performance according to the criteria set by the system designers. Scheduler refers to a set of policies and mechanism, built into the operating system, that governs the order in which work require to be done by the computer.

A process and a thread differ to each other in terms of their execution priority. An application can be implemented as a number of threads that cooperate and execute concurrently in the same address space. On uni-processor, thread can be used as a program structuring aid and to overlap I/O with processing (when one thread is waiting for I/O, another thread may be executed of the same program). The switching cost for thread is less than the switching cost over process. The real advantage of thread appears in multi-processor systems where threads can be used to exploit the parallelism in applications and due to which significant interactions among threads occur. A combination of thread management and process scheduling together can improve upon the performance of the system. Some popular multi-processor thread schedulings are:

(i) Load Sharing (ii) Gang (Group) Scheduling (iii) Dedicated Processor Assignment (iv) Dynamic Scheduling.

A multi-level queue scheduling algorithm partitions the ready queue into separate queues. Processes are permanently assigned to one queue, generally based on

[1] Department of Maths and Statistics, Dr. H.S. Gour University, Sagar (M.P.) – 470003. E-mail:diwakarshukla @rediffmail.com

[2] Department of Computer Science and Applications, Dr. H.S. Gour University, Sagar (M.P.)- 470003 .E-mail : iamsaurabh_4@yahoo.co.in

[3] Department. of Computer Science and Applications, Dr. H.S. Gour University, Sagar (M.P.) - 470003. E-mail : meshweta1 @ gmail.com

some property of the process such as memory size, process priority or process type. Each queue has its own scheduling algorithm.

In this paper, a Markov chain model is used to examine the scheduler's transition behavior among threads and multi-processors in the multi-level queue environment. Scheduler picks up threads from processes and randomly allocates to processors. The focus is on to analyze the transition probabilities in thread scheduling and to simulate the movement mechanism of the scheduler procedure under the assumed probabilistic environment in the form of a model.

A. Motivation

Naldi [4] has applied Markov chain model technique to interpret the flow of internet traffic among various network operators. Shukla et al. [9], [10] used the same to explain the pattern of information flow in Space Division switches and Knockout switches. Shukla and Jain [7], [8] utilized the Markov Chain model technique for the study of scheduler transition mechanism in the multilevel queue scheduling of an operating system. The fundamental basics of this technique are described in detail due to Medhi [3]. Deriving an idea from all these contributions, this paper also incorporates the use of Markov Chain model to study the scheduler transition behavior the thread scheduling procedure.

B. A Review

The DRR algorithm suggested by Shreedar and Varghese [6] has shown a betterment over queuing pattern and DRRA algorithm is an efficient version {15}. Bennet and Zhang [1] suggested an improved fair weight queuing algorithm. A detailed description of round robin routing is due to Liu and Towsley [2] whereas Nelson and Towsley [5] provided a methodology to evaluate the performance of parallel processing systems. The useful contribution over a variety of scheduling schemes and their relative comparisons are due to Silberschatz and Galvin [11], Stalling [12], Tanenbaum and Woodhull [13], Zhang [14], Arco et al. [15] etc.

2. Markov Chain Model

Assume that there are three parallel processors; P_1, P_2, P_3 and three queues Q_1, Q_2, Q_3 in a processing system. The queue Q_k has a process R_k, each with three threads (k=1,2,3). Define a Markov chain $\{X^{(n)}, n^3 0\}$ where X is position of scheduler over states at the nth jump (or transition). Each thread is a state (3+3+3=9) and processors P_k are additional three states. In all, to assume X jumps over 12 states (t_1, t_2, t_3, …, t_9 for threads and P_1, P_2, P_3 for processors) randomly among n transitions.

The transition diagram for the case of three queues in each R_k is given in fig. 2.0

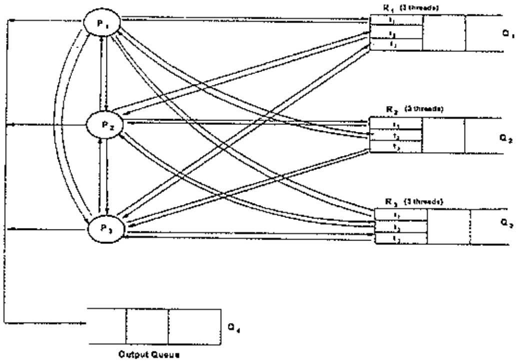

Figure 2 : The Transition Diagram

All the three threads of process R_k could be processed in any of three processors. There is a single queue (Q_4) used to exit the fully processed R_k through P_k. Following are some further assumptions:

(i) The transition of threads from R_k to P_k occurs but no transition from P_k to R_k occurs

(ii) The $X^{(n)}$ denotes the transition of ith thread to jth processor in nth step of transition. The corresponding unit-step transition probability matrix is in fig. 9.0.

3. Calculation Of Transition Probabilities

Let priority of the queue Q_k is a_k and $\sum_{k=1}^{3} a_k = 1$. This supports for multilevel queue scheduling such that each queue has a process with three threads. Suppose initial priorities of threads are b_i so that $P[X^{(0)}=t_i] = b_i$: $\sum_{i=1}^{9} b_i = 1$, i=1,2,3….9.

Remark 3.0: The state probabilities of processors (as per fig. 9.0) after the first transition:

$$P\left[X^{(1)} = P_1\right]_I = \sum_{i=1}^{9} P\left[X^{(0)} = t_i\right] P\left[X^{(1)} = P_1 / X^{(0)} = t_i\right]$$

$$= b_1 s_{1,10} + b_2 s_{2,10} + \ldots\ldots.. + b_9 s_{9,10}$$

$$= \sum_{i=1}^{9} b_i s_{i,10}$$

$$P\left[X^{(1)} = P_2\right]_I = \sum_{i=1}^{9} b_i s_{i,11}$$

$$P\left[X^{(1)} = P_3\right]_I = \sum_{i=1}^{9} b_i s_{i,12}$$

Remark 3.1: The state probabilities of processors after the second transition:

$$P\left[X^{(2)} = P_1\right]_I = \sum_{j=10}^{12}\left[\sum_{i=1}^{9} b_i s_{i,j}\left(s_{j,10}\right)\right]$$

$$P\left[X^{(2)} = P_2\right]_I = \sum_{j=10}^{12}\left[\sum_{i=1}^{9} b_i s_{i,j}\left(s_{j,11}\right)\right]$$

$$P\left[X^{(2)} = P_3\right]_I = \sum_{j=10}^{12}\left[\sum_{i=1}^{9} b_i s_{i,j}\left(s_{j,12}\right)\right]$$

Remark 3.2: Generalized probabilities of processors after in transitions:

$$P\left[X^{(n)} = P_1\right] = \sum_{m=10}^{12} \ldots\ldots\ldots \sum_{l=10}^{12} \sum_{k=10}^{12}\left[\sum_{j=10}^{12}\left\{\sum_{i=1}^{9} b_i s_{i,j}\left(s_{j,k}\right)\right\}\right] s_{k,l} \ldots\ldots s_{m,10}$$

$$P\left[X^{(n)} = P_2\right] = \sum_{m=10}^{12} \ldots\ldots\ldots \sum_{l=10}^{12} \sum_{k=10}^{12}\left[\sum_{j=10}^{12}\left\{\sum_{i=1}^{9} b_i s_{i,j}\left(s_{j,k}\right)\right\}\right] s_{k,l} \ldots\ldots s_{m,11}$$

$$P\left[X^{(n)} = P_3\right] = \sum_{m=10}^{12} \ldots\ldots\ldots \sum_{l=10}^{12} \sum_{k=10}^{12}\left[\sum_{j=10}^{12}\left\{\sum_{i=1}^{9} b_i s_{i,j}\left(s_{j,k}\right)\right\}\right] s_{k,l} \ldots\ldots s_{m,12}$$

4. Calculation Of Thread Probabilities

Remark 4.0: State probabilities of processors after the first transition:

$$P\left[X^{(1)} = P_1\right]_{II} = \sum_{i=1}^{9} P\left[X^{(0)} = t_i\right] P\left[X^{1} = P_1 / X^{0} = t_i\right]$$

$$= b_1 s_{1,10} + b_2 s_{2,10} + \ldots\ldots.. + b_9 s_{9,10} = \sum_{i=1}^{9} b_i s_{i,10}$$

$$P\left[X^{(1)} = P_2\right]_{II} = \sum_{i=1}^{9} b_i s_{i,11}$$

$$P\left[X^{(1)} = P_3\right]_{II} = \sum_{i=1}^{9} b_i s_{i,12}$$

Remark 4.1: State probabilities of threads after the second transition:

$$P\left[X^{(2)} = t_1\right] = \sum_{j=10}^{12}\left[\sum_{i=1}^{9} b_i s_{i,j}\left(s_{j,1}\right)\right]$$

$$P\left[X^{(2)} = t_2\right] = \sum_{j=10}^{12}\left[\sum_{i=1}^{9} b_i s_{i,j}\left(s_{j,2}\right)\right]$$

$$P\left[X^{(2)} = t_3\right] = \sum_{j=10}^{12}\left[\sum_{i=1}^{9} b_i s_{i,j}\left(s_{j,3}\right)\right]$$

$$P\left[X^{(2)} = t_9\right] = \sum_{j=10}^{12}\left[\sum_{i=1}^{9} b_i s_{i,j}\left(s_{j,9}\right)\right]$$

Remark 4.2: State probabilities of processors after the third transition:

$$P\left[X^{(3)} = P_1\right]_{II} = \sum_{k=1}^{9}\left[\sum_{j=10}^{12}\left(\sum_{i=1}^{9} b_i s_{i,j}\right)\right] s_{k,10}$$

$$P\left[X^{(3)} = P_2\right]_{II} = \sum_{k=1}^{9}\left[\sum_{j=10}^{12}\left(\sum_{i=1}^{9} b_i s_{i,j}\right)\right] s_{k,11}$$

$$P\left[X^{(3)} = P_3\right]_{II} = \sum_{k=1}^{9}\left[\sum_{j=10}^{12}\left(\sum_{i=1}^{9} b_i s_{i,j}\right)\right] s_{k,12}$$

Remark 4.3: Generalized expressions for processors state after the n transitions:

$$P\left[X^{(n)} = P_1\right]_{II} = \sum_{q=1}^{9} \ldots\ldots \sum_{l=10}^{12}\left[\sum_{k=1}^{9}\left\{\sum_{j=10}^{12}\left(\sum_{i=1}^{9} b_i s_{i,j} s_{j,k}\right)\right\} s_{k,l}\right] \ldots\ldots s_{q,1}$$

$$P\left[X^{(n)} = P_2\right]_{II} = \sum_{q=1}^{9} \ldots\ldots \sum_{l=10}^{12}\left[\sum_{k=1}^{9}\left\{\sum_{j=10}^{12}\left(\sum_{i=1}^{9} b_i s_{i,j} s_{j,k}\right)\right\} s_{k,l}\right] \ldots\ldots s_{q,2}$$

$$P\left[X^{(n)} = P_3\right]_{II} = \sum_{q=1}^{9} \ldots\ldots \sum_{l=10}^{12}\left[\sum_{k=1}^{9}\left\{\sum_{j=10}^{12}\left(\sum_{i=1}^{9} b_i s_{i,j} s_{j,k}\right)\right\} s_{k,l}\right] \ldots\ldots s_{q,3}$$

Remark 4.4: Generalized expressions of thread state after the n transitions:

$$P\left[X^{(n+1)} = t_1\right] = \sum_{w=10}^{12} \ldots\ldots \sum_{l=10}^{12}\left[\sum_{k=1}^{9}\left\{\sum_{j=10}^{12}\left(\sum_{i=1}^{9} b_i s_{i,j} s_{j,k}\right)\right\} s_{k,l}\right] \ldots s_{w,1}$$

$$P\left[X^{(n+1)} = t_1\right] = \sum_{w=10}^{12} \ldots\ldots \sum_{l=10}^{12}\left[\sum_{k=1}^{9}\left\{\sum_{j=10}^{12}\left(\sum_{i=1}^{9} b_i s_{i,j} s_{j,k}\right)\right\} s_{k,l}\right] \ldots s_{w,2}$$

$$P\left[X^{(n+1)} = t_3\right] = \sum_{w=10}^{12} \ldots\ldots \sum_{l=10}^{12}\left[\sum_{k=1}^{9}\left\{\sum_{j=10}^{12}\left(\sum_{i=1}^{9} b_i s_{i,j} s_{j,k}\right)\right\} s_{k,l}\right] \ldots s_{w,3}$$

$$P\left[X^{(n+1)} = t_9\right] = \sum_{w=10}^{12} \ldots \sum_{l=10}^{12}\left[\sum_{k=1}^{9}\left\{\sum_{j=10}^{12}\left(\sum_{i=1}^{9} b_i s_{i,j} s_{j,k}\right)\right\} s_{k,l}\right] \ldots s_{w,9}$$

5. Processor and Thread Indices

Define three kinds of processor and thread indices:

(a) Processor Index $I\left[P_k^{(n)}\right]$

$$I\left[P_k^{(n)}\right]=\frac{P\left[X^{(n)}=P_k\right]_I}{\sum_{i=1}^{9}P\left[X^{(n)}=P_k\right]_I}$$

(b) Processor-Thread Index $T\left[P_k^{(n)}\right]$

$$T\left[P_k^{(n)}\right]=\frac{P\left[X^{(n)}=P_k\right]_{II}}{\sum_{i=1}^{9}P\left[X^{(n)}=P_k\right]_{II}}$$

(c) Thread index $T\left[t_i^{(n)}\right]$

$$T\left[t_i^{(n)}\right]=\frac{P\left[X^{(n)}=t_i\right]}{\sum_{i=1}^{9}P\left[X^{(n)}=t_i\right]}$$

6. Simulation Study

Consider following three data sets on which graphs are obtained through simulation.

Data set I:

$P\left[X^{(0)}=t_1\right]=0.1, P\left[X^{(0)}=t_2\right]=0.1, P\left[X^{(0)}=t_3\right]=0.1,$

$P\left[X^{(0)}=t_4\right]=0.1, P\left[X^{(0)}=t_5\right]=0.1, P\left[X^{(0)}=t_6\right]=0.1,$

$P\left[X^{(0)}=t_7\right]=0.1, P\left[X^{(0)}=t_8\right]=0.1, P\left[X^{(0)}=t_9\right]=0.2$

$X^{(n+1)}$

$X^{(n)}$	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	P_1	P_2	P_3
t_1	0	0	0	0	0	0	0	0	0	0.3	0.3	0.4
t_2	0	0	0	0	0	0	0	0	0	0.3	0.3	0.4
t_3	0	0	0	0	0	0	0	0	0	0.3	0.3	0.4
t_4	0	0	0	0	0	0	0	0	0	0.3	0.3	0.4
t_5	0	0	0	0	0	0	0	0	0	0.3	0.3	0.4
t_6	0	0	0	0	0	0	0	0	0	0.3	0.3	0.4
t_7	0	0	0	0	0	0	0	0	0	0.3	0.3	0.4
t_8	0	0	0	0	0	0	0	0	0	0.3	0.3	0.4
t_9	0	0	0	0	0	0	0	0	0	0.3	0.3	0.4
P_1	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.087
P_2	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.087
P_3	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.087

Figure 3 : Transition Probability Matrix For Data Set I

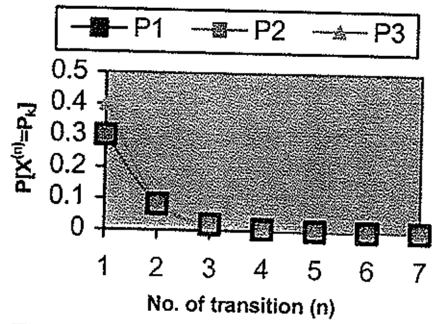

Figure 4 [A] : Processor to Processor

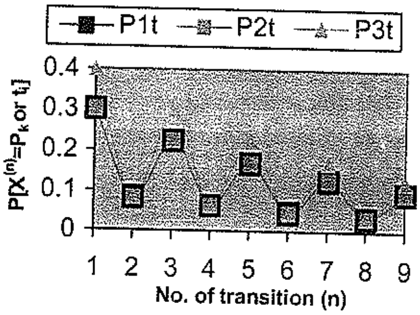

Figure 4 [B] : Processor to Thread

Data set II:

$P\left[X^{(0)}=t_1\right]=0.1, P\left[X^{(0)}=t_2\right]=0.1, P\left[X^{(0)}=t_3\right]=0.1, P\left[X^{(0)}=t_4\right]=0.1,$
$P\left[X^{(0)}=t_5\right]=0.1, P\left[X^{(0)}=t_6\right]=0.1, P\left[X^{(0)}=t_7\right]=0.1, P\left[X^{(0)}=t_8\right]=0.1,$
$P\left[X^{(0)}=t_9\right]=0.2$

$X^{(n+1)}$

$X^{(n)}$	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	P_1	P_2	P_3
t_1	0	0	0	0	0	0	0	0	0	0.25	0.35	0.4
t_2	0	0	0	0	0	0	0	0	0	0.33	0.41	0.26
t_3	0	0	0	0	0	0	0	0	0	0.29	0.35	0.36
t_4	0	0	0	0	0	0	0	0	0	0.32	0.34	0.34
t_5	0	0	0	0	0	0	0	0	0	0.4	0.38	0.22
t_6	0	0	0	0	0	0	0	0	0	0.33	0.36	0.31
t_7	0	0	0	0	0	0	0	0	0	0.27	0.37	0.36
t_8	0	0	0	0	0	0	0	0	0	0.42	0.26	0.32
t_9	0	0	0	0	0	0	0	0	0	0.43	0.3	0.27
P_1	0.079	0.081	0.082	0.078	0.079	0.082	0.083	0.084	0.078	0.084	0.082	0.10
P_2	0.082	0.081	0.078	0.079	0.081	0.069	0.079	0.081	0.082	0.078	0.072	0.13
P_3	0.083	0.082	0.068	0.069	0.072	0.065	0.069	0.075	0.078	0.081	0.082	0.17

Figure 5 : Transition Probability Matrix For Data Set II)

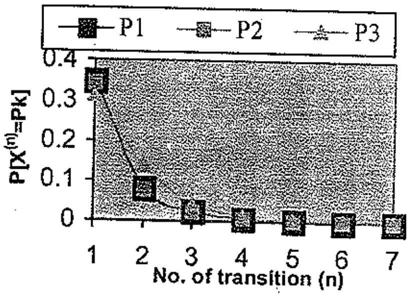

Figure 6 [A] : Processor to Processor

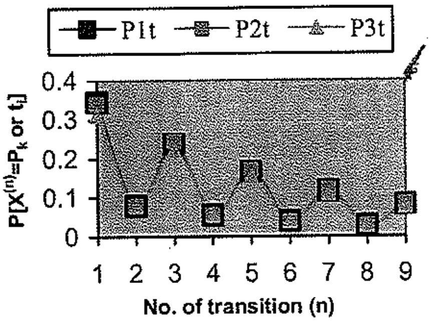

Figure 6 [B] : Processor to Thread

Data set III:

$$P[X^{(0)}=t_1]=0.1, P[X^{(0)}=t_2]=0.1, P[X^{(0)}=t_3]=0.1, P[X^{(0)}=t_4]=0.1,$$
$$P[X^{(0)}=t_5]=0.1, P[X^{(0)}=t_6]=0.1, P[X^{(0)}=t_7]=0.1, P[X^{(0)}=t_8]=0.1,$$
$$P[X^{(0)}=t_9]=0.2$$

$X^{(n+1)}$

$X^{(n)}$	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	P_1	P_2	P_3
t_1	0	0	0	0	0	0	0	0	0	0.23	0.35	0.42
t_2	0	0	0	0	0	0	0	0	0	0.35	0.38	0.27
t_3	0	0	0	0	0	0	0	0	0	0.37	0.27	0.36
t_4	0	0	0	0	0	0	0	0	0	0.26	0.32	0.42
t_5	0	0	0	0	0	0	0	0	0	0.27	0.39	0.34
t_6	0	0	0	0	0	0	0	0	0	0.43	0.37	0.2
t_7	0	0	0	0	0	0	0	0	0	0.32	0.33	0.35
t_8	0	0	0	0	0	0	0	0	0	0.33	0.336	0.31
t_9	0	0	0	0	0	0	0	0	0	0.41	0.33	0.26
P_1	0.082	0.081	0.078	0.068	0.065	0.071	0.075	0.081	0.065	0.069	0.082	0.183
P_2	0.078	0.081	0.065	0.082	0.081	0.079	0.075	0.086	0.081	0.075	0.079	0.138
P_3	0.081	0.083	0.068	0.082	0.081	0.065	0.069	0.071	0.072	0.065	0.073	0.19

Figure 7 : Transition Probability Matrix For Data Set

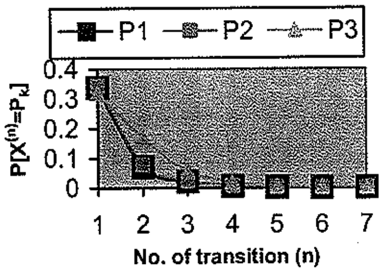

Figure 8 [A] : Processor to Processor

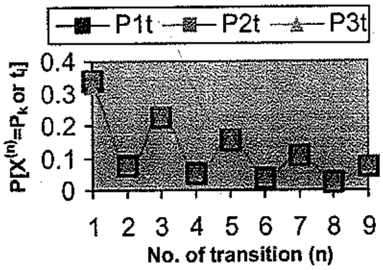

Figure 8 [B] : Processor to Thread

7. Concluding Remarks

Over the increasing transitions, the state probabilities of processors are going down. The similar pattern of downward trend is found for all three data sets. When coming to the aspect of scheduler transition over processors under thread scheduling it seems, initially the processor may be assigned priorities for scheduler but with the increase of transitions this priority nullifies. While comparing state probabilities of threads and processors together, both constantly reduces over increasing n in all three data sets. But, the processors state probability remains high over the thread state probabilities. This indicates more and more involvement of scheduler towards processor in the thread scheduling algorithm.

$X^{(n+1)}$

$X^{(n)}$	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	P_1	P_2	P_3
t_1	0	0	0	0	0	0	0	0	0	$s_{1,10}$	$s_{1,11}$	$s_{1,12}$
t_2	0	0	0	0	0	0	0	0	0	$s_{2,10}$	$s_{2,11}$	$s_{2,12}$
t_3	0	0	0	0	0	0	0	0	0	$s_{3,10}$	$s_{3,11}$	$s_{3,12}$
t_4	0	0	0	0	0	0	0	0	0	$s_{4,10}$	$s_{4,11}$	$s_{4,12}$
t_5	0	0	0	0	0	0	0	0	0	$s_{5,10}$	$s_{5,11}$	$s_{5,12}$
t_6	0	0	0	0	0	0	0	0	0	$s_{6,10}$	$s_{6,11}$	$s_{6,12}$
t_7	0	0	0	0	0	0	0	0	0	$s_{7,10}$	$s_{7,11}$	$s_{7,12}$
t_8	0	0	0	0	0	0	0	0	0	$s_{8,10}$	$s_{8,11}$	$s_{8,12}$
t_9	0	0	0	0	0	0	0	0	0	$s_{9,10}$	$s_{9,11}$	$s_{9,12}$
P_1	$s_{10,1}$	$s_{10,2}$	$s_{10,3}$	$s_{10,4}$	$s_{10,5}$	$s_{10,6}$	$s_{10,7}$	$s_{10,8}$	$s_{10,9}$	$s_{10,10}$	$s_{10,11}$	$s_{10,12}$
P_2	$s_{11,1}$	$s_{11,2}$	$s_{11,3}$	$s_{11,4}$	$s_{11,5}$	$s_{11,6}$	$s_{11,7}$	$s_{11,8}$	$s_{11,9}$	$s_{11,10}$	$s_{11,11}$	$s_{11,12}$
P_3	$s_{12,1}$	$s_{12,2}$	$s_{12,3}$	$s_{12,4}$	$s_{12,5}$	$s_{12,6}$	$s_{12,7}$	$s_{12,8}$	$s_{12,9}$	$s_{12,10}$	$s_{12,11}$	$s_{12,12}$

Figure 9 : Transition Probability Matrix Of Markov Chain Model

References

[1] J.C.R. Bennett, and H. Zhang, "*WF²Q: Worst-Case Fair Weight Fair Queuing*", INFOCOM'96, PP. 1.d.3.1-1.d.3.9, 1996.

[2] Z. Liu, and D. Towsley, "*Optimality of Round Robin scheduling policy*", Journal of Applied Probability, Vol. 26(1), PP. 68-79, 1994

[3] J. Medhi, " *Stochastic Processes*", Ed. 4, Wiley Limited (Fourth Reprint), New Delhi, 1991.

[4] M. Naldi, " *Internet Access Traffic Sharing in a multi-user environment*", Computer Networks, Vol. 38, PP. 809-824, 2002.

[5] R. Nelson and D. Towsley, " *A Performance of several priority policies for parallel processing systems* ", Journal of ACM, Vol. 40, PP. 714-740, 1993.

[6] M. Shreedhar, and G. Varghese, "*Efficient-Fair queuing using Deficit Round Robin*", SIGCOMM'95 conf., ACM, PP. 231-243, 1995.

[7] D. Shukla, and Saurabh Jain, " *A Markov chain model for multilevel queue scheduler in operating system* ", Proceedings of International Conference on Mathematics and Computer Science, ICMCS-07, PP. 522-526, 2007 a.

[8] D. Shukla, and Saurabh Jain, " *Deadlock state study in security based multilevel queue scheduling scheme in operating system* ", Proceedings of National Conference on Network Security and Management, NCNSM-07, PP. 166-175, 2007 b.

[9] D. Shukla, S.K. Gadewar and R.K. Pathak, *"A Stochastic model for Space-Division Switches in Computer Networks"*, Applied Mathematics and Computation (Elsevier Journal), Vol. 184, Issue 2, PP. 235-269, 2007.

[10] D. Shukla and S.K. Gadewar, " *Stochastic model for cell movement in a Knockout-Switch in Computer Networks* ", Jour. of High Speed Network, (IOS Press Journal) , Vol. 16(3), PP. 301-321, 2007.

[11] A. Silberschatz, and P. Galvin, *"Operating System Concepts"*, Ed.5, John Wiley and Sons (Asia), Inc, 1999.

[12] W. Stalling, "*Operating Systems*", Ed.5, Pearson Eduaction, Singopore, Indian Edition, New Delhi, 2004.

[13] A. Tanenbaum, "*Operating system* ", Ed. 8, Prentice Hall of India, New Delhi, 2000.

[14] H. Zhang, *"Service Disciplines for Guaranteed Performance in Packet-Switching Networks"*, Proceedings of IEEE, Vol. 38, No. 10, PP. 1374-1396, 1995.

[15] J.M. Arco, D. Meziat and B. Alarcos, "*Deficit Round Robin Alternated: A New scheduling Algorithm*", 5th IEEE Conference on Protocols for Multimedia Systems, PROMS-2000, PP. 302-311, (20 Author's Biography 00).

Author's Biography

Dr. Diwakar Shukla ,Professor, is a permanent faculty member in the Department of Mathematics and Statistics, Sagar University, Sagar, M.P. with over 19 years experience of teaching to U.G and P.G classes. He obtained M.Sc.(Stat.), Ph.D.(Stat.) degrees from Banaras Hindu University, Varanasi and served the Devi Ahilya University, Indore, M.P. as a Lecturer for nearly nine years and obtained M.Tech.(Computer Science) degree from there. He joined the Sasar University,Sagar, MP, as a Reader in 1998. During Ph.D work at BHU., he was recipient of Junior and Senior research fellowships of CSIR, New Delhi, through All India Fellowship Examination of 1983. Till now, he has published more than 50 research papers in national and international journals and participated in over 35 seminars/conferences at the national level. He is also the recipient of MPCOST Young Scientist Award, ISAS Young Scientist Medal, UGC Career Award and UGC Visiting Fellow to the Amerawati University, Maharashtra. He worked as a Professor, for one year, to the Lucknow University, Lucknow, U.P., and visited abroad to Sydney (Australia) and Shanghai (China) for the conference participation. He has authored one book and supervised seven Ph.D. theses in Statistics and Computer Science both; eight students are presently enrolled with him for their doctoral degree. He is a member of 10 learned bodies of Statistics and Computer Science at national level. The area of research he works for are Sampling Theory, Graph Theory, Stochastic Modeling, Computer Network Switching Systems and Operating Systems.

Mr Saurabh Jain has obtained M.C.A. degree from H.S. Gour University, Sagar,MP, in 2005. He is presently working as a Lecturer in the Department of Computer Science & Applications in the same University since 2007. He did his research in the field of CPU scheduling and Operating systems. He has authored and co-authored 6 research papers published in journals and conference proceedings. His current research interest is to analyze the scheduler's performance and algorithms both under probability models.

Mrs. Shweta Ojha received her M.C.A. degree from H.S. Gour University, Sagar in 2005. She is presently working as a Lecturer in the Department of Computer Science & Applications in the same University since 2007. She is enrolled for the doctoral degree. Her research interest is to examine the scheduler's performance under various algorithms in probabilistic environment. She has some research papers published in conferences proceedings.