ISSN 1512-1232

BEHAVIORIAL STUDY OF SINGLE LEVEL TREE NETWORK WITH THE HELP OF MARKOV CHAIN MODEL

[1]Saurabh Jain, [2]Shweta Ojha

[1] Deptt. of MCA, Shri Vaishnav Institute of Technology and Science, Indore (M.P.) e-mail: iamsaurabh_4@yahoo.co.in
[2]Deptt. of Computer Sc. and Applications, H. S. Gour University, Sagar (M.P.), 470003 e-mail: meshwetaojha1@gmail.com

Abstract

In this present era of networking, the different opportunities are being explored in order to maximize it's utilization within limited resources. It is in the benefit of the system that the same infrastructure should be utilized for more and more connections. The recent research works are going to explore all the possible options that are feasible. The existing option could be in the field of designing new scheduling algorithms. The work done by Moges and Robertazzi has developed the concept of divisible load scheduling and have shown its analogue with the Markov chain model. Working on the same line, in this paper, we are trying to predict the behavior of single level tree network with the help of Markov chain model. It will wave a path for network designer for it's application and use.

Keywords

Markov chain model, Single level tree network, Divisible jobs, Divisible load scheduling, Supportive divisible load scheduling.

1. Introduction

The pressure mounted on the system for handling the gradually increasing number of jobs, has triggered a new search for improvising the present structures and also need for entirely new techniques. Based on the concept of analogy between the divisible load scheduling and Markov chain model [1], this paper has been derived to highlight the possibility of supportive divisible load scheduling concept and its behavioral analogy with the Markov chain model is also established. Some conceptual model and their descriptions has been used in order to deal with the topic. A common load is being shared by different processors and their links. The sharing is affected according to the structure of the system[2,3]. By this, it is obvious that there is not any precedence relation among the data. Applications including grid computing, parallel and distributed processor network scheduling, meta computing, data intensive computing and very many other high end computing and processing devices. It will be quite a useful tool in multimedia and computer utility applications in now-a-days products.

Optimal load allocation for different network topologies are included in linear daisy chain, bus, tree and complex network using recursive equation sets presented in [4]. In [5], the integration of monitory cost optimization and divisible load theory is presented. The scheduling policy research includes independent task scheduling [6], Multi-round algorithm [7] in fixed communication. The transition labeling method has been devised differently from the commonly used methods in queuing theory model diagram and other related works, irrespective of the fact that most of the Markov chain models are one dimensional. Distributed processing and it's cost related issues [8] have been greatly explained by J. Blazewicz and M. Drozdowski.

An equivalent continuous time Markov chain model [9,10] for various network topologies and load scheduling policies currently modeled by divisible load theory are produced. The idea for introducing this unification between divisible load theory and Markov chain model is that they have

ISSN 1512-1232

a number of commonality between them. Basically, both the theories are linear in nature. That means each of them can be solved in theory by solving the associated linear set of equations. Many optimal divisible load schedules for various network topologies have Markov chain analogue. It helps in explaining the similarity between queuing theory and divisible load theory. However, the stochastic nature of the Markov chain model shows a strange similarity with the deterministic natured divisible load theory. This new equivalence provides a new and powerful modeling tool. On the basis of this analogue, a new concept of supportive divisible load scheduling policy is proposed.

Markov chain model is an arch but useful technique which has again drawn the attention of many researchers . D. Shukla and S. Jain has [11] developed a Markov chain model for Multi-level queue scheduling. The space division switches of the computer network area has also been studied on the basis of stochastic model approach[12]. The predictive nature of the Markov chain model has been further utilized for analyzing the different users behavior in the internet environment. D. Shukla, S. Jain and R. Singhai has studied the behavior of Round Robin scheduling scheme with the help of Markov chain model [16]. A data model approach has been designed by D. Shukla, S. Jain and S. Ojha for the behavioral study of Multi-Level queue with the help of Markov chain model [17].

2. Model Definition

Initially, the divisible load scheduling theory is described with the help of definitions and notations. The network structure used in this paper is basically the linear daisy chain with homogenous and heterogeneous links. The model consists of both a homogenous and heterogeneous processors with the initial processor termed as root processor.

The total processing load carrying to root processor is randomly divided into different equal/unequal fractions of load to be given to each processor over the network. The root processor keeps some of the load to itself and sends out rest of the load to remaining processors on the network. It is assumed that, the processors on the network are of diverse type depending on their specific criteria to whether or not compute and communicate at the same time. Basically we are focusing on two cases, with front end processors and without front end processors.

2.1 Symbols used in Model

In this paper, the following symbols and their definitions will be used:

α_i : The fraction of load that is assigned to the processor i by the load originating processor.

Z_i : A constant that is inversely proportion to the speed of link i in the network.

W_i: A constant that is inversely proportional to the computation speed of processor i in the network.

T_{cm} : Communication intensity constant. This is the time that the link takes to transmit the entire processing load over a link when $Z_i=1$. The entire load can be transmitted over the i^{th} link in time $Z_i Tcm$.

T_{cp} : Computation intensity constant. This is the time that the i^{th} processor takes to process the entire processing load when $W_i=1$. The complete load can be processed on the i^{th} processor in time $W_i Tcp$.

T_i : It is the total time taken by i^{th} processor, starting from the beginning of the processor at t=0 and the time when i^{th} processor completes the computation process. This symbol includes the computation time, communication time and the idle time of the processor.
i.e. ($T_i = \alpha_i W_i Tcp + \alpha_i Z_i Tcm$ + idle time)

T_f : Processing finish time of total processing load, assuming load is delivered to the originating processor by t=0
and $T_f = \max(T_i)$

Y_i : A constant that is inversely proportional to the speed of the link from i^{th} processor to the supportive processor.

α_s : The fraction of load of the supportive processor, assigned to it by the i^{th} processor through which it is interacting at the time t. It varies with the processors.

Compulsory assumption: The load originating at the root processor is supposed to be normalized as unit load.

3. Behavioral Similarity between The Markov Chain Model and Single Level Tree Network

We are going to consider the same concept of sharing the load of the main processors of the network by a supportive processor by giving way to the existence of a new supportive divisible load scheduling concept. In order to prove it's possibility, here we are designing a model of a single level tree network with N+1 processors and having N links between them. It can be shown by the figure below:

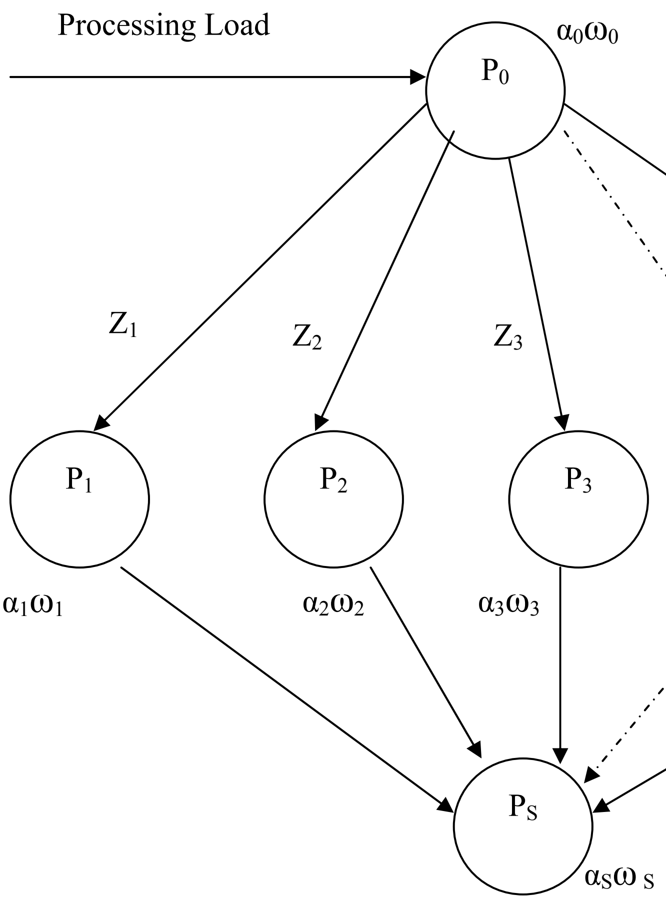

Fig: Load Distribution in Single Level Tree with heterogeneous links without Front End and Support Processor

The initial node processor takes the upcoming load and is distributed sequentially, the remaining load to its child processor at the lower level. Each processor in the network is assumed to have no front end processor. That means, the root processor will first finish communicating all of the load to be transmitted to the lower level before it starts computing it's own load. The terminal processor starts computing it's own fraction of load. Other processors start computing only after receiving their respective fraction of load- that is also called as staggered start.

ISSN 1512-1232

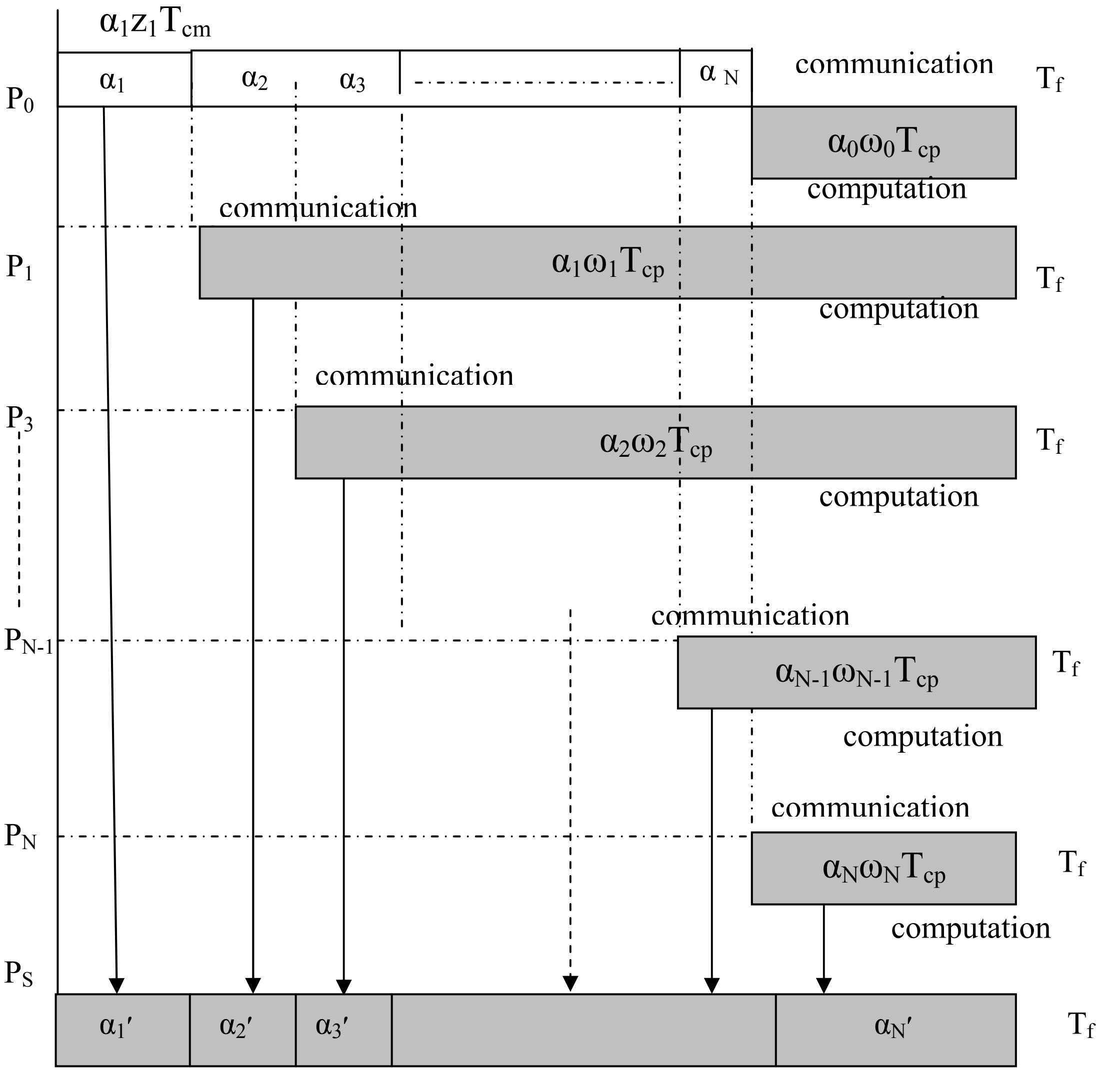

Fig: Timing Diagram for Single Level Tree without Front End Processor with Heterogeneous links and Support Processor

The equation for minimum finish time can be written as:

$$\alpha_0\,\omega_0 T_{cp} = \alpha_N \omega_N T_{cp} + \alpha_1'(Z_s T_{cm} + \omega_s T_{cp})$$

$$\alpha_1\,\omega_1 T_{cp} = \alpha_2 z_2 T_{cm} + \alpha_2 \omega_2 T_{cp} + \alpha_2'(Z_S T_{cm} + \omega_s T_{cp})$$

$$\alpha_2\,\omega_2 T_{cp} = \alpha_3 z_3 T_{cm} + \alpha_3 \omega_3 T_{cp} + \alpha_3'(Z_S T_{cm} + \omega_s T_{cp})$$

Similarly, the N^{th} term equation will be

$$\alpha_{N-1}\,\omega_{N-1} T_{cp} = \alpha_N z_N T_{cm} + \alpha_N \omega_N T_{cp} + \alpha_N'(Z_S T_{cm} + \omega_s T_{cp})$$

and by taking $(Z_S T_{cm} + \omega_s T_{cp}) = C$, the equation will become

$$\alpha_{N-1}\,\omega_{N-1} T_{cp} = \alpha_N z_N T_{cm} + \alpha_N \omega_N T_{cp} + \alpha_N' C$$

ISSN 1512-1232

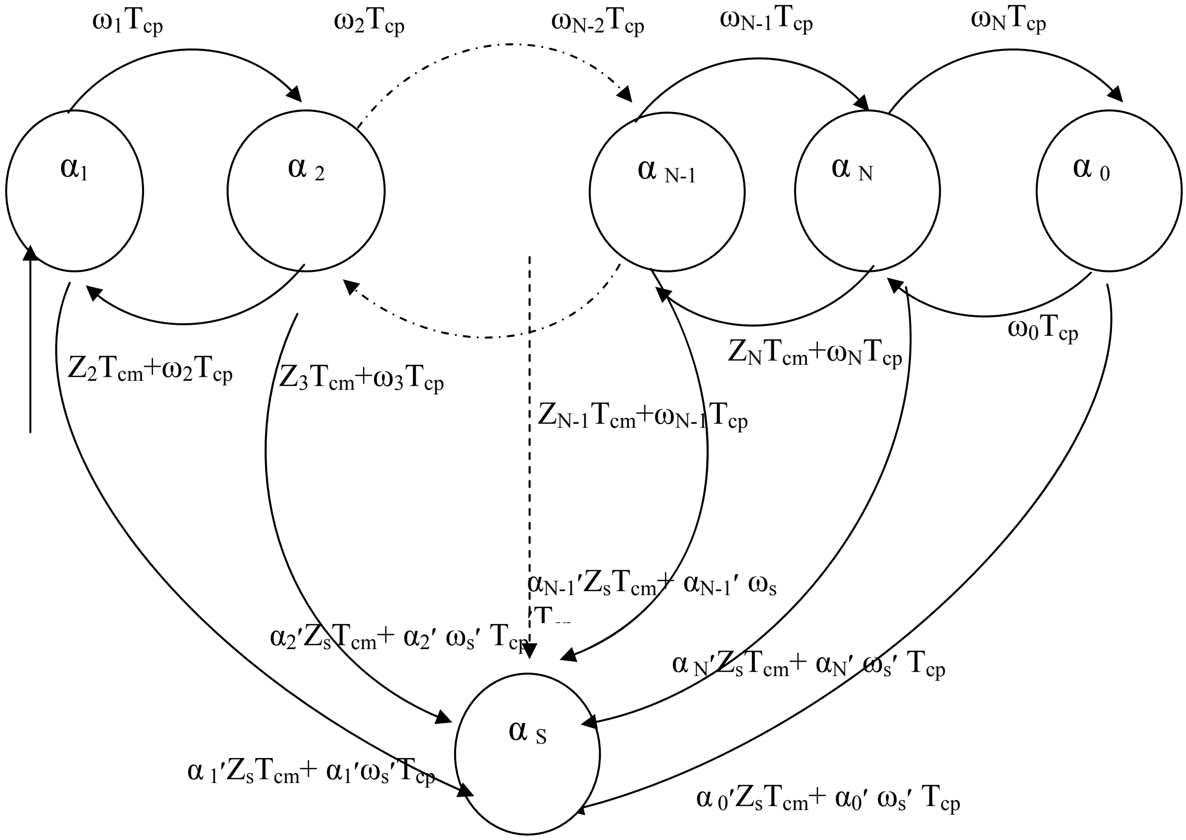

Fig: Markov chain model for Single Level Tree network without front end processor, heterogeneous link and with a support processor

For a different kind of tree network with N+1 processor, N links and one support processor with homogeneous link is considered. Here again, the root processor P_0 keeps a fraction of load for itself to compute and distribute the remaining load to it's child processors at the next lower level sequentially. Due to the presence of front end processors, the processor starts computing it's fraction of load at the moment it finishes receiving it's data load.

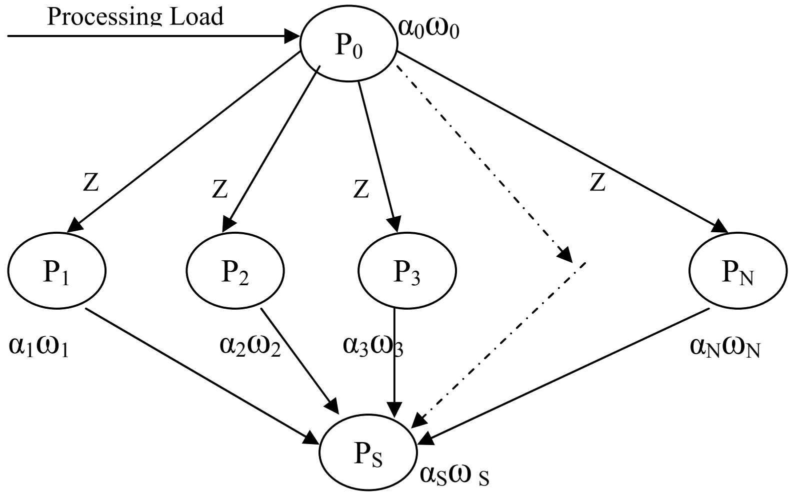

Fig: Load Distribution in Single Level Tree with Front End Processor and Homogeneous link and a Support Processor

ISSN 1512-1232

The timing diagram showing the process of load distribution for a single level tree network with front end processor and homogeneous links is shown below:

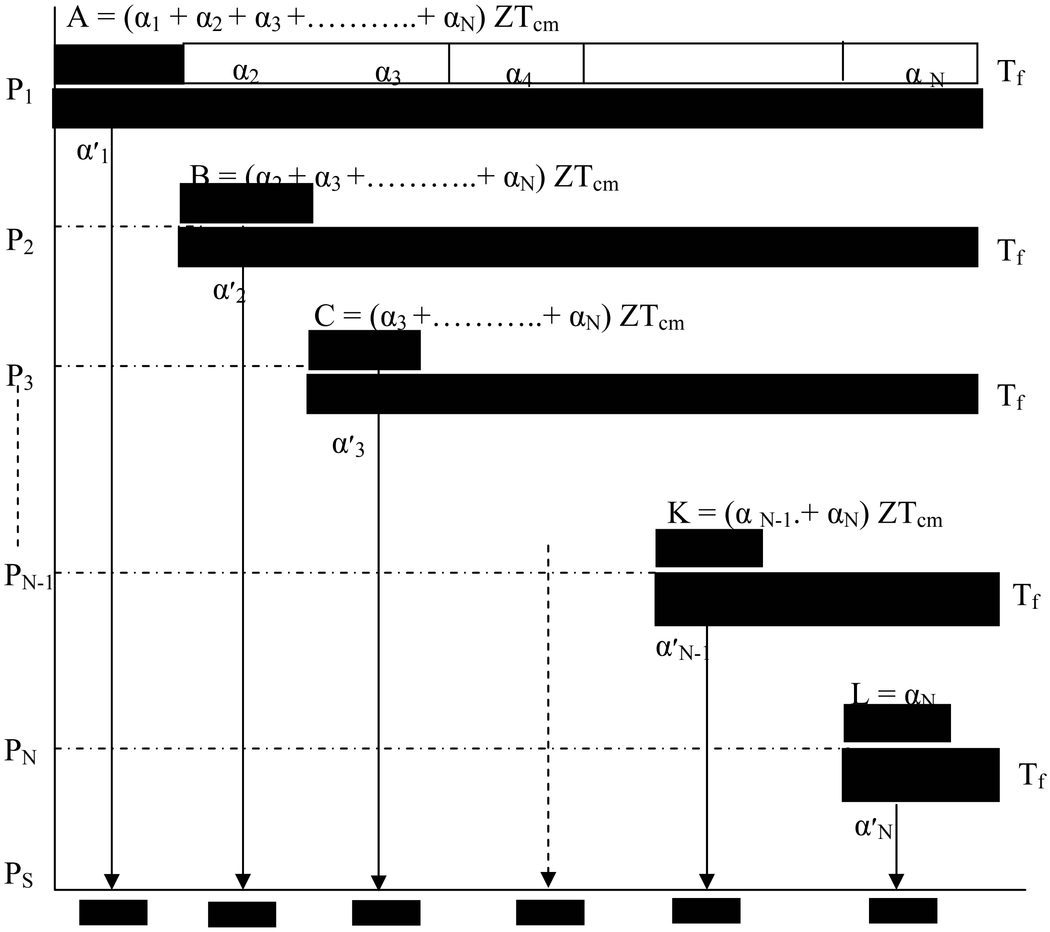

Fig: Timing Diagram for Single Level Tree Network with homogeneous links and support processor

Here again ,the equations for the minimum finish time can be written as:

$$\alpha_0\,\omega_0 T_{cp} = \alpha_1 zT_{cm} + \alpha_1\omega_1 T_{cp} + \alpha_1'(ZT_{cm} + \omega_s T_{cp})$$
$$\alpha_1\,\omega_1 T_{cp} = \alpha_2 zT_{cm} + \alpha_2\omega_2 T_{cp} + \alpha_2'(ZT_{cm} + \omega_s T_{cp})$$

⋮

further, considering $(ZT_{cm} + \omega_s T_{cp})$=C the equations are taking the same form as in case with linear daisy chain network, i.e.

$$\alpha_{N-1}\omega_{N-1}T_{cp} = \alpha_N zT_{cm} + \alpha_N\omega_N T_{cp} + \alpha_{N-1}'\,C$$

ISSN 1512-1232

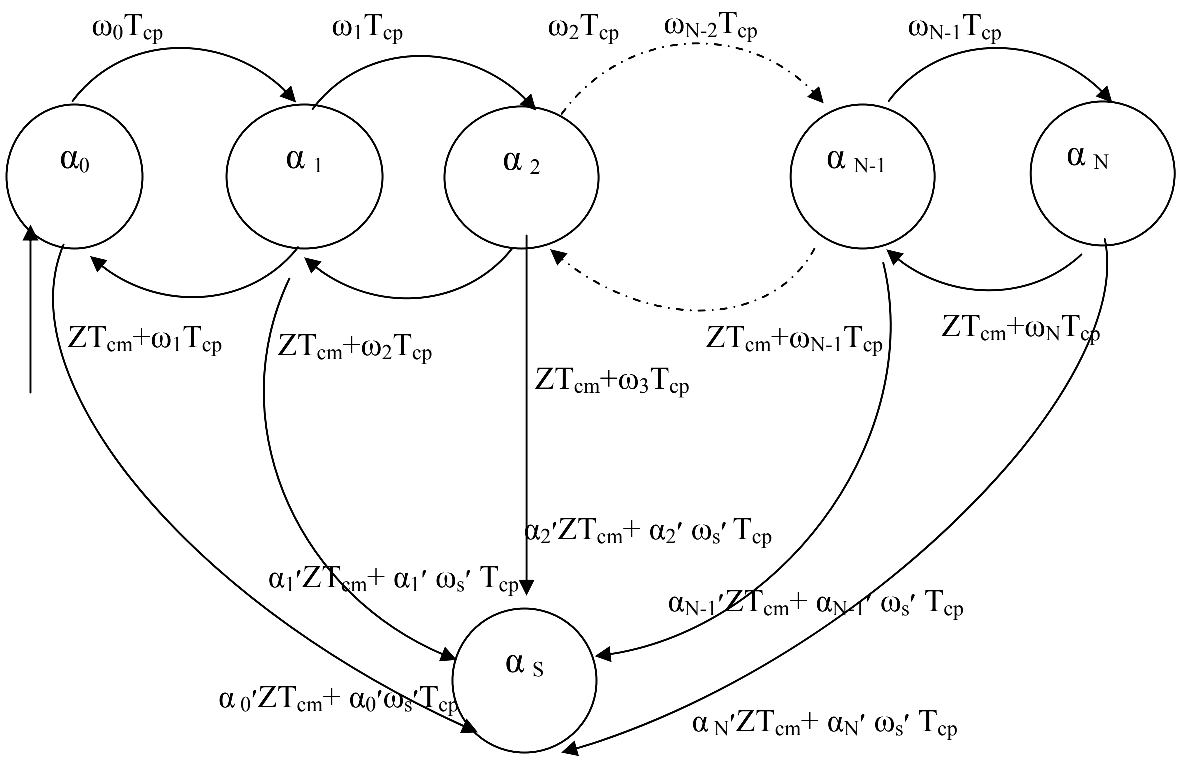

Fig: Markov chain model for Single Level Tree network with cut-through switching, front end processor, homogeneous link and with a support processor

4. Conclusion

This model has been designed in order to parameterize the overall efficiency and behaviour of supportive divisible load scheduling scheme on the background of Markov chain model. The whole model is being described with the help of some local balance equations. This work is focused on single level tree network structure on varying conditions of homogenous and heterogeneous links. Further the same concept could be applied on two level and multilevel tree network and on their different variants. Through this concept, it has been concluded that this supported divisible load scheduling theory has a large number of similarity with Markov chain model. In the parallel, it has also been concluded that not every parameterized supportive divisible load scheduling strategy model has a corresponding Markov chain model. But wherever such similarity persists, it will prove to be a great help in deriving the specific features for the model. Some of the model, even after being entirely different in their network topologies like single level tree network in both homogeneous and heterogeneous links, share strange similarity in behavior in terms of timing diagram or their Markov chain model analogue.

ISSN 1512-1232

References

1. M.A. Moges and T.G. Robertazzi, Divisible Load Scheduling and Markov Chain Models, Preprint submitted to Elsevier Science, (2006).
2. V. Bharadwaj, D. Ghose, V. Mani, An efficient load distribution strategy for a distributed linear network of processors with communication delays. *Computer and Mathematics with Applications*, 29 95-112 (1995).
3. S. Bataineh and T.G. Robertazzi, Bus oriented load sharing for a network of sensor driven processors. *IEEE Transactions on Systems, Man and Cybernetics*, 21 1202-1205 (1991).
4. Y.C. Cheng and T.G. Robertazzi, Distributed computation for a tree network with communication delays. *IEEE Transactions on Aerospace and Electronic Systems*, 26 511-516 (1990).
5. J. Sohn, T.G. Robertazzi and S. Luryi, Optimizing Computing Costs Using Divisible Load Analysis. *IEEE Transactions on Parallel and Distributed Systems*, 9 225-234 (1998).
6. O. Beaumont, A. Legrand, and Y. Robert, Optimal algorithms for scheduling divisible workloads on heterogeneous systems. *12th Heterogeneous Computing Workshops HCW'2003*, (2003).
7. Y. Yang, H. Casanova, UMR: A Multi-Round Algorithm for Scheduling Divisible Workloads. *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'03)*, Nice, France, (2003).
8. J. Blazewicz and M. Drozdowski, Distributed Processing of Distributed Jobs with Communication Startup Costs. *Discrete Applied Mathematics*, 76 21-41, (1997).
9. J.R. Jackson, Networks of Waiting Lines. *Operations Research*, 5 518-521 (1957).
10. T.G. Robertazzi, Computer Networks and Systems: Queueing theory and Performance Evaluation. *3rd edition*, Springer-Verlag, (2000).
11. D. Shukla, and Saurabh Jain: A Markov chain model for multi-level queue scheduler in operating system, *Proc. International Conference on Mathematics and Computer Science, ICMCS-07*, 2007, pp. 522-526.
12. D. Shukla, S. Gadewar, and R.K. Pathak, *A Stochastic model for space-division switches in computer networks*, Applied mathematics and Computation (Elsevier Journal), 184(2), 2007, pp. 235-269.
13. D. Shukla, and Sanjay Thakur, *Index based Internet Traffic Sharing Analysis of users by a Markov chain probability model*, Accepted for publishing Journal of Computer science (JCS), 4(3), 2010.
14. D. Shukla, S. Thakur, V. Tiwari, and A. Deshmukh, *Share Loss Analysis of Internet Traffic Distribution in Computer Networks*, International Journal of Computer Science and Security (IJCSS), 3(5), 2009, pp. 414-427.
15. D. Shukla, and Sanjay Thakur, *State Probability Analysis of Users in Internet between two Operators*, International Journal of Advanced Networking and Applications (IJANA), 1(1), 2009, pp. 90-95.
16. D. Shukla, S. Jain and R. Singhai, *A Markov Chain Model for the Analysis of Round-Robin Scheduling Scheme*, International Journal of Advanced Networking and Applications (IJANA), 1(1), 2009.
17. D. Shukla, S. Jain and S. Ojha, *Data model based analysis of multi-level queue scheduling using Markov Chain Model,* Accepted for publication in International Journal of Applied Computer Science and Mathematics, July Issues, 2010.

Article received: 2010-12-28