DETERMINATION OF THE VALUE OF CONVERGENCE PARAMETER IN A PROCEDURE OF CALCULATING THE UPPER BOUNDARY OF THROUGHPUT FOR PACKET SWITCH

T. Tashev[1]. A. Bakanov[2], R.Tasheva[3]

[1]*Institute of Information and Communication Technologies, Bulgarian Academy of* Sciences, ttashev@iit.bas.bg
[2]*Institute of Psychology, Russian Academy of* Science, *arsb2000@pochta.ru*
[3] *Technical University of Sofia, Bulgaria*, rpt@tu-sofia.bg

Abstract. The task which includes computing of non-conflict schedule for crossbar switch node of packet commutation is NP-hard. After checking their efficiency with throughput modelling by uniform load traffic, the check for non-uniform traffic is required. In the presented paper we propose a procedure of calculating the upper boundary of throughput for crossbar packet switch node. The results of the computer simulation of switch throughput for hotspot load by grid-cluster of IICT-BAS are presented. Simulation utilizes PIM-algorithm for non-conflict schedule, specified by apparatus of Generalized Nets. It is shown that the usage of the suggested procedure permit to evaluate influence of the step of simulation on the precision of boundary values for throughput.

Key words: Computer Technologies, Modeling, Network nodes, Crossbar switch

1. INTRODUCTION

Modern digital information systems are built on the principle of exchange of information of discrete portions, called packets. Communication nodes in these systems are called router and switch. A crossbar switch node routes traffic from the input to output where a message packet is transmitted from the source to the destination. The randomly incoming traffic must be controlled and scheduled to eliminate conflict at the crossbar switch. The goal of the traffic-scheduling for the crossbar switches is to maximize the throughput of packet through a switch and to minimize packet blocking probability and packet waiting time [1]. This is assured by the algorithm for calculation of non-conflict schedule which is running in the control unit of the switch (Scheduler – Fig.1) [2].

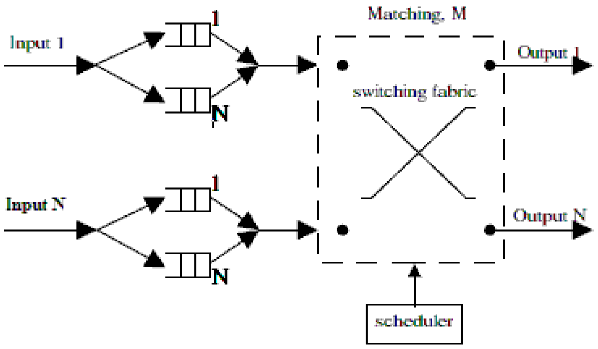

Fig. 1 Architecture of crossbar switch node

The problem of calculating of non-conflict schedule is NP-complete [3]. Increasing data volumes [4] and increasing the speed of transmission lines of communication require new, more efficient algorithms for the calculation of the conflict-free schedule. The efficiency of these algorithms can be verified with a formal or simulation tools.

The efficiency of the algorithms for switches in the first place can be evaluated by using bandwidth output channels (throughput). The incoming traffic may be uniform or non-uniform. The study of an algorithm's throughput begins with modeling of the switch throughput under uniform load traffic (uniform i.i.d Bernoulli traffic). The next step is to investigate the algorithm's throughput under non-uniform traffic [5].

In the previous paper [6] we proposed a procedure for calculating the upper boundary of throughput for crossbar switch node using the results of simulation throughput. If throughput of crossbar node increases to a certain limit (monotonically), the procedure provides one unique solution. This solution can be determined by the value of step of convergence, which value is calculated on the result of the simulations with a certain error.

In this paper we will show that under certain conditions the value of the step of convergence gets a precise meaning. In this case, comparing this value with the simulation results, we will obtain directly the computational error of the procedure.

2. CONDITIONS FOR THE SIMULATION

The load traffic may be various types [5] presenting by traffic matrix $\boldsymbol{T}$. Traffic matrix $\boldsymbol{T}$ for switching fabric with size $N \times N$ is defined according to Gupta [7]. A conflict situation is created when in any row of the $\boldsymbol{T}$ matrix contains more than one nonzero element. This corresponds to the case when one source declares connection with more than one receiver. If any column of the $\boldsymbol{T}$ matrix hosts more than one nonzero element, this also indicates a conflict situation. Avoiding conflicts is related to the switch node efficiency. In order to obtain a non-conflict schedule it is necessary to compute a sequence of non-conflict matrices $\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_r$ such that their sum is equal to the traffic matrix T. Each row and column of every matrix $\boldsymbol{Q}_i$, $i = 1, 2, \ldots, r$ has no more than one element equal to 1 and the rest of elements are equal to 0.

We utilize a family of patterns of $\boldsymbol{T}$ for non-uniform traffic simulation based on the hotspot (Chao) model. This model is given by: $\lambda_{ii} = 0{,}5\rho$ for $i = j$ and $\lambda_{ij} = 0{,}5\rho/(N\text{-}1)$ otherwise, $i, j \in 1, \ldots, N$, where ρ is the load intensity of each input (i.i.d. Bermoulli) [9].

The first matrix type $\boldsymbol{T}$ in the family of patterns is called $Chao_1$. Its optimal schedule requires $2(N\text{-}1)$ switches of crossbar matrix for $N \times N$ switch node. In the general case, the i-th matrix type is denoted by $Chao_i$. Its optimal schedule requires $2i(N\text{-}1)$ switches of crossbar matrix for $N \times N$ switch node [8]. This matrix type is shown in Figure 2.

$$T=\underset{2\times 2}{\begin{bmatrix}1&1\\1&1\end{bmatrix}}\ \underset{3\times 3}{\begin{bmatrix}2&1&1\\1&2&1\\1&1&2\end{bmatrix}}\cdots\underset{k\times k}{\begin{bmatrix}k\text{-}1&\dots&1\\ \vdots&\ddots&\vdots\\1&\dots&k\text{-}1\end{bmatrix}}\cdots\qquad T=\underset{2\times 2}{\begin{bmatrix}i&i\\i&i\end{bmatrix}}\ \underset{3\times 3}{\begin{bmatrix}i{*}2&i&i\\i&i{*}2&i\\i&i&i{*}2\end{bmatrix}}\cdots\underset{k\times k}{\begin{bmatrix}i{*}(k\text{-}1)&\dots&i\\ \vdots&\ddots&\vdots\\i&\dots&i{*}(k\text{-}1)\end{bmatrix}}\cdots$$

Fig. 2 Matrices of types $Chao_1$ and $Chao_i$.

For description of switch algorithms many authors use different formal apparatus such as cellular automata, neural networks, queue theory [10], matrices-masks [11] etc. We use the apparatus of Generalized nets (GN) [12]. GN is a contemporary formal tool created to make detailed representation of connections between the structure and timing correspondence in parallel processes [13].

Our simulations is **using the known PIM**-algorithm [14] **for non-conflict schedule specified by the apparatus of Generalized nets. The utilized GN-model for PIM-**algorithm is specified in [8]. The transition from a GN-model to executive program is performed as in [15]. The source code has been compiled by means of the grid-structure BG01-IPP of the Institute of information and communication technologies - Bulgarian Academy of Sciences (http:// www.grid.bas.bg) and the resulting code is executed in the grid-structure.

3. PROCEDURE OF CALCULATING THE UPPER BOUNDARY

We give an informal description of the procedure for determining the convergence of the throughput of the switch to a specific upper boundary on the results of computational experiments for a given family of patterns for load traffic and given algorithm. Approbation of the procedure is done with the patterns for hotspot model of incoming traffic and model of PIM-algorithm [6].

Step 1. Let the obtained results of computational experiments to study throughput are ranked heuristically to trend towards convergence (graphic curves) to some upper bound. The initial evaluation of the required number of graphs is at least 5 (from $Pattern_1$ - $Chao_1$). In our example, we have seven curves (templates). In the figures below, $Chao_i$ is denoted as Ci for i=1,2,… .

We decide if there is any reason to declare the existence of an upper boundary. If "Yes", then we estimate the range of possible values (for $i \to \infty$ and $N \to \infty$). According to this data (7 ready) - 77% more than the boundary, and at least 80% of its maximum throughput. We define the dimension of N for the next steps keeping in mind the following considerations - simulation results should contain data that already have a "sustainable" monotonic increase efficiency.

In our case - Results for Figure 3 – we estimate that N must be greater than 20 (20x20). The maximum value is determined by the available computing power.

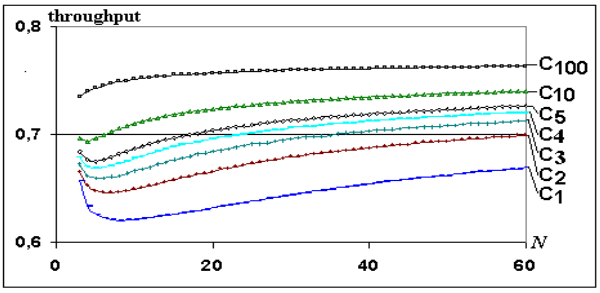

Fig. 3 Throughput of PIM-algorithm with Chao-traffic

Step 2. We carry out simulations for 4 patterns of incoming traffic using the selected permanent "step" m for index i in patterns $Chao_i$. (if the results are ready, gather "together"). 4 patterns - the minimum number of patterns, and $m \in \{2,3,4,\dots\}$.

In [6] we used a value of 10 for the "step" m. Now we will use a value of 5 for the m. This means that you will simulate the throughput of the switch with patterns for incoming traffic $Chao_1$, $Chao_5$, $Chao_{25}$, $Chao_{125}$, $Chao_{625}$, … We get results for C1, C5, C25, C125 – which are shown in Figure 4. The dimension N - from 3x3 to 70x70. 10 000 simulations - the first, second and third patterns, for last ($Chao_{125}$) - 1000.

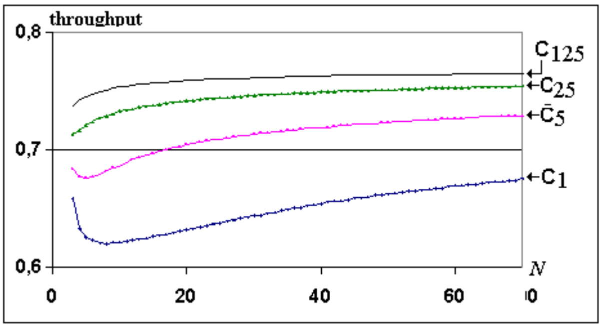

Fig. 4 Throughput for $Chao_1$, $Chao_5$, $Chao_{25}$, $Chao_{125}$

Step 3. Calculation of the difference between throughput for neighboring patterns. We build a curved line for differences as shown in Figure 5. We have 3 curves : Δ_1 = throughput ($Chao_5$) - throughput ($Chao_1$), Δ_2 = throughput ($Chao_{25}$) - throughput ($Chao_5$), Δ_3 = throughput ($Chao_{125}$) - throughput ($Chao_{25}$).

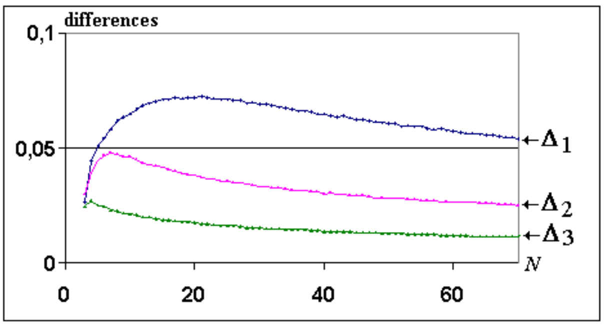

Fig. 5 Differences between throughput

Step 4. We calculate the convergence parameter **δ** - ratio of the difference. We build a curved line for results - shown in Figure 6. We have $\delta_1 = \Delta_1 / \Delta_2$, $\delta_2 = \Delta_2 / \Delta_3$.

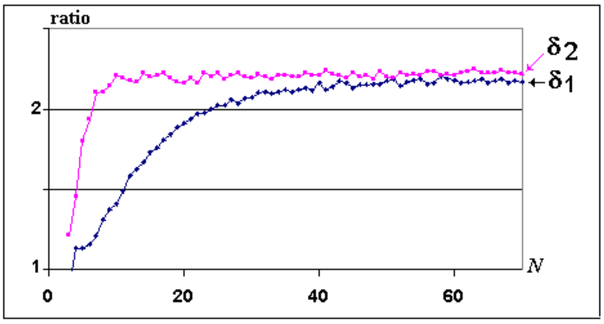

Fig. 6 Ratio (δ_1, δ_2) between differences

The convergence parameter δ_1 is increased to the value of ≈ 2,18. The convergence parameter δ_2 tends to the value of ≈ 2,22. Do these two curves converge to the same value? May be. Error for δ_1 and δ_2 : ± 0,02 are shown in Figure 7.

We need some extra data. The value of the step *m* is 5 - the next pattern for simulation will be a $Chao_{625}$. We calculate the ratio of the difference $\delta_3 = \Delta_3 / \Delta_4 = \Delta_3 /$ (throughput ($Chao_{625}$) - throughput ($Chao_{125}$)). The result is shown in Figure 8. There are 1000 simulations for $Chao_{625}$. The convergence parameter δ_3 tends to the value of ≈ 2,23. Error for δ_3 : ± 0,04.

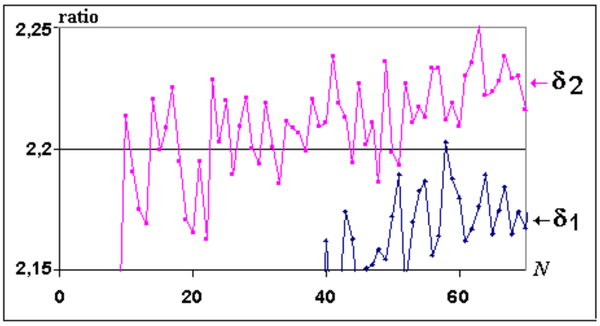

Fig. 7 Ratio (δ_1, δ_2) in a more detailed presentation

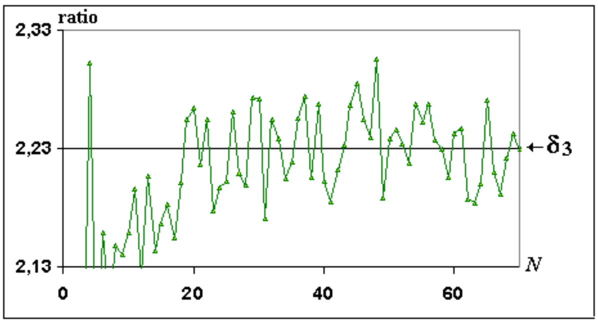

Fig. 8 Ratio (δ_3) between differences

Using the results of the simulations we assume that the parameter of convergence δ has a value of 2.23 with error ± 0,03.

We know from theory that an infinite number series of the form $1/a + 1/a^2 + 1/a^3 + ... + 1/a^i + ...$, where $a > 1$, converges as $i \to \infty$ to the value of $1/(a-1)$. In our case we assume that the parameter of convergence **δ** forms a series with a = 2.23.

Consequently, the coefficient of convergence *Sn* for our results is $Sn = 1 / (2{,}23-1) = 0{,}813(00813) \approx 0{,}813$.

Step 5. We accept what the true border capacity with unlimited buffer increase and for each N> 20 (see Step 1) converges to the value at 0,8130 greater than the difference between this pattern and the former, which we want to use for the calculation.

We compute values of the upper boundary using the pattern with the highest number - $Chao_{625}$:

Boundary-throughput (*N*) = throughput $Chao_{625}$ (*N*) + $Sn \,.\, \Delta_4 (N)$,

where Δ_4=throughput($Chao_{625}$) - throughput ($Chao_{125}$).

The result is shown in Figure 9. The most important throughput = 0,77415 for *N* =70 (0,77399 for *N* = 60).

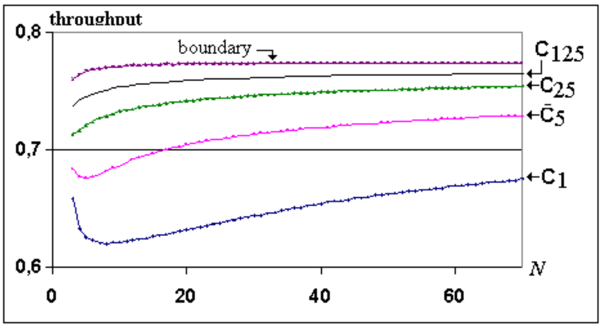

Fig. 9 Upper boundary of throughput

Step 6. We consider the growth trend in Figure 9 throughput values of the border. Keeping in mind the largest value for the throughput 0.77415 obtained in Step 5, we estimate the value as $N \to \infty$ at 0.775.

Step 7. Primary analysis of errors.

Relative error {Boundary-throughput(N)= throughput $Chao_{625}$ (*N*) + $Sn.\, \Delta_4 (N)$ }

The results of the simulations show that the relative error of the throughput at the used patterns is approximately equal. Let's designate it with c RelErr.

In this case the assessment of the relative error for boundary is :

Relative error {Boundary-throughput(N) = throughput $Chao_{625}$ (*N*) + $Sn.\, \Delta_4 (N)$} = RelErr + 4. RelErr + 2. RelErr

Relative error is more dependent (in these terms) on the accuracy of the coefficient of "convergence."

4. ACCURACY OF THE OBTAINED BOUNDARY

During the calculation of the boundary we used the curve of the pattern Chao625 and the value of the **parameter of convergence** δ_3 =2,23 (***Sn***=0,8130). The boundary obtained if the curve of the pattern Chao125 with δ_2=2,22 (***Sn***=0,8196)is used is showed on Figure 10. The curve presented in that case is above this for Chao625. Which of the results is more accurate?

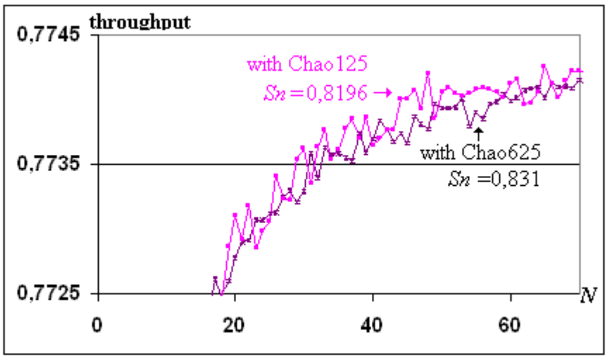

Fig. 10 Upper boundary of throughput, *m*=5

Let's calculate the boundary using the curve of the pattern $Chao_{125}$ and the value of the **parameter of convergence** δ_3. The obtained result is showed on Figure 11. The correlation of the curves is better than on Figure 10. Is this enough to accept that the value of δ_3 is the most accurate one? We will be using the data from [6] for the boundary obtained using step *m*=10. The comparison of the curves is showed on Figure 12. Having in mind that comparison we accept that the value of δ_3 is more accurate.

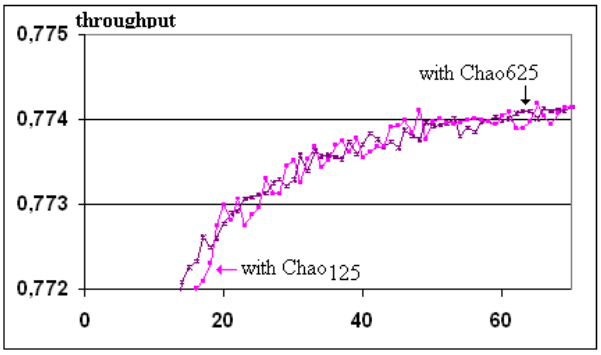

Fig. 11 Upper boundary , *m*=5, *Sn*=0,813

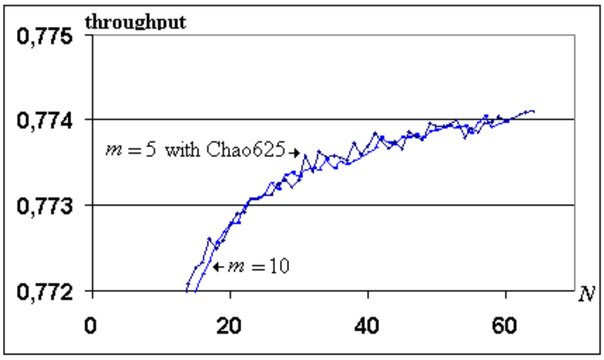

Fig. 12 Upper boundary, *m*=10,*m*=5 ($Chao_{625}$)

Besides, we can answer the question for the accurate value of the convergence parameter **δ**.

Proposition: The accurate value of the convergence parameter **δ** equals square root равна----- of the step for pattern choice *m* (from Step 2): $\boldsymbol{\delta = m^{1/2}}$.

Reasoning: The results of the simulations. In [6] at *m*=10 is calculated δ=3,15 ± 0,02 . $10^{1/2}$≈3,162277. Here we obtain at *m*=5 δ=2,23 ± 0,03. $5^{1/2}$≈2,236068.

Consequence 1: The differences between the obtained during the simulations values of δ_i and the value $\boldsymbol{m^{1/2}}$ are equal to the absolute error δ. This is an independent way to evaluate the resultant errors.

Consequence 2: If $\boldsymbol{m^{1/2}}$ is the most accurate value of δ, we decrease the error of the boundary calculation twice.

Consequence 3: It is possible to choose the step *m*, accordingly to the calculation resources available for the simulation implementation.

Proofs for the proposition validity can be received through new simulations – at bigger dimensions of *N* and different combinations for incoming traffic. This is an object for further investigations.

5. CONCLUSION

The results of the implemented modeling show that in the proposed procedure the value of the parameter of convergence equals the square root of the step chosen for the pattern simulations of the incoming traffic. As a consequence we can evaluate the error of the simulation upper boundary for switch throughput. It is shown that the usage of the suggested procedure permit to evaluate influence of the step of simulation on the precision of boundary values for throughput.

The suggested procedure is better to prove using simpler model of incoming traffic (for example uniform) and also try on the example of other - more complicated (for example , unbalanced model) – traffic models.

REFERENCES

1. Ермаков А.С. Модель балансировки коммутаци-онных процессов. Proc. of Int. Workshop "DCCN-2007", Sofia, Bulgaria, 2007. Moscow, Technosfera, 2007, pp. 171-176.
2. Deb S., Shah D., Shakkottai S. Fast Matching Algorithms for Repetitive Optimization: An Application to Switch Scheduling. Information Sciences and Systems, 2006 40th Annual Conference on, 22-24 March 2006, pp. 1266-1271.
3. Chen T, Mavor J, Denyer Ph, Renshaw D. Traffic routing algorithm for serial superchip system customisation // IEE Proc.- E.part, Jan 1990, Vol. 137, No.1, pp. 65-73.
4. Атанасова Т. E-Home - агрегиране на данни за повишаване на енергийна ефективност. Съвместен научен семинар ИИКТ-БАН, ИМИ-БАН, КТП, *„Моделиране и управление на информационните процеси"* Сборник трудове, КТП, София, 2010, стр. 69-75.
5. Chang H.J, Qu G., Zheng S.Q. Performance of CTC(N) Switch under Various Traffic Models. Springer, Lecture Notes in Electrical Engineering, Volume 126, 2012, pp. 785-793.
6. Ташев Т., Баканова Н., Ташева Р. Исследование верхней границы пропускной способности коммутаци-онного узла при входящем трафике типа "горячей точки". International Journal "Information Technologies & Knowledge", Vol.7, No. 2, 2013, pp. 182-189.
7. Gupta P., McKeown N. Designing and Implementing a Fast Crossbar Sheduler. IEEE Micro, Jan-Feb 1999, pp. 20-28.
8. Tashev T., Monov V.. Modeling of the Hotspot Load Traffic for Crossbar Switch Node by Means of Generalized

Nets. Proc. of Intelligent Systems (IS), 2012 6th IEEE International Conference, 6-8 Sept. 2012 , Sofia, Bulgaria. pp. 187-191
9 Chao-Lin Yu, Cheng-Shang Chang, Duan-Shin Lee.CR Switch: A Load-Balanced Switch With Contention and Reservation. IEEE/ACM Transactions on Networking, vol. 17 no.5, October 2007. pp.1659–1671.
10. Chao H.J, Lui B. High prformance sitches and routers. John Wiley & Sons, 2007.
11. Kolchakov K.H An Approach for Performance Improvement of Class of Algorithms for Synthesis of Non-conflict Schedule in the Switch Nodes. Proc. of the 11th Int. Conf. CompSysTech'10, 17-18 June 2010, Sofia, Bulgaria. ACM Press, ICPS Vol. 471, 2010, pp. 235-239.
12. Atanassov K. Generalized Nets and System Theory. Acad.Press "Prof. M. Drinov", Sofia, Bulgaria, 1997.
13. Гочев, П. Гочева. .NET реализация на Обобщени мрежи. Обектно-ориентиран подход. Издателски център на ВУ КТП, София, 2012. 114 стр.
14. Anderson T., Owicki S., Saxe J., and Thacker C., High speed switch scheduling for local area networks. ACM Trans. Comput. Syst., vol. 11, no.4, Nov. 1993, pp. 319-352.
15. Tashev T., Vorobiov V. Generalized Net Model for Non-Conflict Switch in Communication Node. Proc. of Int. Workshop "DCCN'2007", Sept 10-12 2007, Moscow, pp. 158-163.
16. Tashev T., Monov V. Large-Scale Simulation of Uniform Load Traffic for Modeling of Throughput on a Crossbar Switch Node. 8-th Int.Conf. on "Large-Scale Scientific Computations" 6-10 June 2011, Sozopol, Bulgaria. Springer, Lecture Notes in Computer Science (LNCS) Volume 7116 (2012). pp.630-637.