COMPUTER SIMULATION OF THE THROUGHPUT OF CROSSBAR SWITCH WITH MODIFIED CHANG'S MODEL FOR LOAD TRAFFIC

T. Tashev[1], V. Monov[1], M. Marinov[2]
[1] Institute of information and communication technologies – B.A.S., Sofia, Bolgaria
[2] Technical University – Sofia, Sofia, Bolgaria

Abstract: *Achieving maximum throughput (THR) of crossbar switch node is obtained by calculating the non-conflict schedule to switch incoming packets. In this paper the results from a numerical simulation of the THR obtained using the grid-cluster of IICT-BAS (ww.grid.bas.bg) are presented. Our simulation employs MiMa-algorithm specified by apparatus of Generalized Nets. We utilize family of patterns of our own design for non-symmetric traffic (based on the Chang-model). The main result includes determining of the upper bound of the THR of the MiMa-algorithm as the maximal possible throughput value equal to 100%.*

Key words: *Computer and Communication Networks, Crossbar Switch Node, Throughput modeling, Generalized Nets, Algorithms.*

КОМПЬЮТЕРНАЯ СИМУЛЯЦИЯ ПРОПУСКНОЙ СПОСОБНОСТИ КОММУТАТОРА С МАТРИЧНЫМ ПЕРЕКЛЮЧАТЕЛЕМ ДЛЯ ВХОДЯЩЕГО ТРАФИКА ТИПА МОДИФИЦИРОВАННОЙ МОДЕЛИ ЧАНГ-А

Т. Ташев[1], В. Монов[1], М. Маринов[2]
[1] Институт информационных и коммуникационных технологий Болгарской Академии Наук, София, Болгария
Технический университет – София, София, Болгария

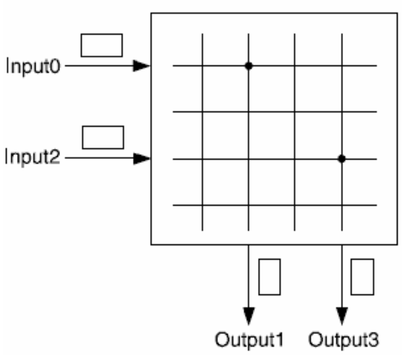
Резюме: Достижение максимальной пропускной способности (ПС) коммутационного узла с матричным переключателем получается путем вычисления бесконфликтного расписания для коммутации входящих пакетов. В данной работе представлены результаты численного моделирования ПС, полученые с использованием вычислительной сети IICT-BAS (ww.grid.bas.bg). Наше моделирование использует MiMa-алгоритм, специфицированный аппаратом Обобщенных сетей. Для несимметричного входящего трафика мы применили семейство шаблонов на основе Чанг-модели. Основной результат состоит в определения максимальной ПС при MiMa-алгоритме как стремящееся к значению равному 100%.

Ключевые слова: Компьютерные и коммуникационные сети, Пакетный коммутатор, Оценка производительности, Обобщенные сети, Алгоритмы.

1. Introduction

Crossbar switch node is a device which maximizes the speed of data transfer using parallel existing paths between the input and output lines in the commutation field of the node (as shown in Figure 1) [1]. This is obtained by means of a non-conflict commutation schedule calculated by the control block (Scheduler) of the switch node (as shown in Figure 2) [2]. From a mathematical point of view the calculation of such a schedule is NP-complete [3]. The existing classical solutions (PIM-algorithm [4], iSLIP-algorithm [5]) partly solved the problem. New more effective algorithms for schedule calculation are needed and they have to be checked for efficiency. As examples we can mention the CTC(N) [6] and RR/LQF [2] algorithms.

The efficiency of the switch performance is firstly evaluated by the throughput (THR) provided by the node for uniform and non-uniform load traffic [1, 2, 6]. The modelling of the THR should be performed for large area of the switch field (from 2x2 to at least 128x128 input/output lines). Such simulations and the necessary computations are typically carried out by using grid-computer structures. In our previous works we used this approach [7].



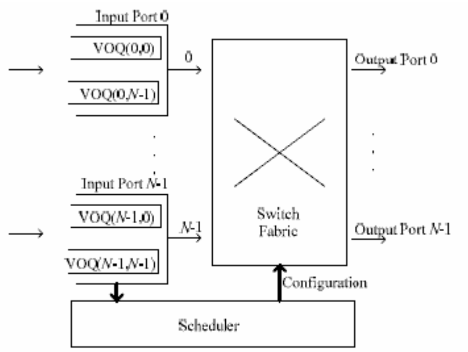


Fig.1 Crossbar switching field. Fig.2 An input-queued switch with a scheduler.

In this paper we determine the THR of a crossbar switch (with input buffering and Virtual Output Queues – Figure 2) by means of a computer simulation up to 130x130 lines. We utilize family of patterns of our own design for non-symmetric traffic (based on the Chang-model [8]). Our simulation employs MiMa-algorithm (developed by the first author) specified by apparatus of Generalized Nets (GN). Generalized Nets [9] are a formal tools allowing a representation of connections in serial and parallel processes [10, 11].

The main results of the paper include determining of an upper bound of the throughput for MiMa-algorithm under the specified family of patterns. The bound of the throughput of the MiMa-algorithm approaches the maximal possible throughput value equal to 100%.

2. MiMa-algorithm for computing a conflict-free schedule

The requests for packet transmission through switching $n \times n$ line switch node is presented by an $n \times n$ matrix $\boldsymbol{T}$, named traffic matrix (n is integer). Matrix $\boldsymbol{T}$ describes unidirectional packet flow - from input lines to output lines. Every element t_{ij} ($t_{ij} \in \{0, 1, 2, \ldots\}$) of the traffic matrix represents a request for a packet from input i to output j. For example $t_{ij} = p$ means that p packets from the $i-$th input line have to be send to $j-$th output line of the switch node [5].

A conflict situation arises when in any row of the matrix $\boldsymbol{T}$ the number of requests is more than one. This corresponds to a case when source declares packets for more than one receiver. If any column of a matrix $\boldsymbol{T}$ contain more than one element different from zero, this indicates also a conflict situation. Avoiding conflicts is closely related to the effectiveness of the switch node. In order to obtain the non-conflict schedule is necessary to compute the sequence of conflict-free matrices Q_1, . . . , Q_m such that their sum will be equal to the traffic matrix $\boldsymbol{T}$. Each column and row of matrices Q_i, $i = 1,2,\ldots,m$ has no more than 1 element equal to one and the rest of elements are equal to 0.

We will give a brief description of the MiMa-algorithm (compute the matrix Q_1).

Step 1. Initially, matrix $\boldsymbol{T}$ is introduced (t_{ij} $_{:}I,j\in$ { 1, 2, …, n}, n=const).

Step 2. A vector-column, which consists of the number of conflicts in each row (row conflict weights) is calculated.

Step 3. If there is no requests (vector-column contain only 0-elements) then go to **Step 9**. Else - continue.

Step 4. A vector-row, which consists of the number of conflicts in each column (column conflict weights), is calculated too.

Step 5. In the vector-row we choose the maximal element which determines the column with the most conflicts.

Step 6. In the vector-column we choose the maximal element which determines the row with the most conflicts.

Step 7. If there is a request in the place of intersection of the column and row with most conflicts,

then: we take this request as an element of the non-conflict matrix Q_1. Temporary records zero(0)-weight for these input and output lines. Go to **Step 2**.

else: (If there is no request) we choose the element in the vector-column which is closest in value to the maximal element. The element in the vector row remains the same.

Step 8. We check if there is a request in the intersection, etc. (like **Step 7**, we omit details).

As a result for the chosen column of $\boldsymbol{T}$ we will have a request selected for commutation (if such a request exists at all). The row and column containing the selected request are excluded from the computation of Q_1. Go to **Step 2** (The next elements of Q_1 are computed by repeating the above procedure).

Step 9. Stop.

As a result the first matrix Q_1 may consists of elements (requests) with maximal weight of conflicts in T. The next non-conflict matrices Q_2, . . . ,Q_m are computed analogously. The last matrix Q_m will contain only the non-conflict requests in matrix $\boldsymbol{T}$.

Our model is developed for packet switch node with an equal number of inputs and outputs. Its graphic form is shown on Figure 3. At the first moment of the current modeling time, one token enters into place l_1 (start). This token represents requests for sending a packet (all packets have the same size). It has an initial characteristic : "ch_0 = (pr_1ch_0, pr_2ch_0) = < n, $\boldsymbol{T}$>" (the number of the input/output lines, noted by n and the traffic matrix $\boldsymbol{T}$). The end of the MiMa-algorithm is indicated by receiving a token in the place l_{22} (stop). At this moment the place l_{20} contains the tokens of the final non-conflict schedule (the tokens who represent the solutions $Q_1, Q_2, \ldots, Q_m$).

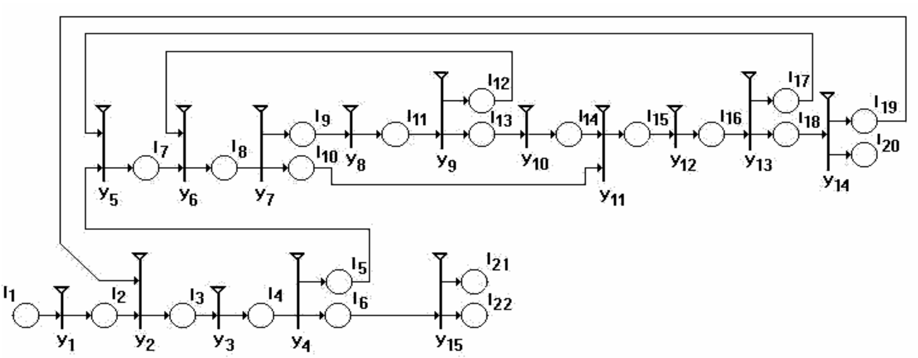


Fig.3 Graphical form of GM-model of MiMa-algorithm

The model has possibilities to provide information about the number of switching in crossbar matrix, as well as about the average number of packets transmitted by one switch. Analysis of the model proves receiving a non-conflict schedule.

3. Models for load traffic

For simulations we will use a modified pattern model for Chang's [8] load traffic. Our basic model includes a family of patterns denoted below as Cg-i , i=1,2,… Requests represent packets with the same size. The index *i* shows values of element in the traffic matrix. The optimal schedule for Cg-1 requires k_{opt} = 1.(*n-1)* switchings of the crossbar matrix for *n* x *n* switch. The optimal schedule for Cg-i requires $k_{\text{opt}} = i.(n-1)$ switchings of the crossbar matrix for *n* x *n* switch. The throughput is computed by dividing the result of optimal solution by the result of the simulated solution: if the Mima-algorithm gives the schedule with k_r switchings, then the throughput will be k_{opt} / k_r. This model ensures 100% load intensity of the input lines and 100% workflow of the output lines. The traffic matrices for Cg-1 and traffic matrices for Cg-i are shown in Figure 4. The generation of these patterns does not depend on the type of hardware and software simulation tools.

$$T^1_{(2\times2)}=\begin{bmatrix}0&1\\1&0\end{bmatrix}\; T^1_{(3\times3)}=\begin{bmatrix}0&1&1\\1&0&1\\1&1&0\end{bmatrix}\;\cdots\; T^1_{(k\times k)}=\begin{bmatrix}0&\cdots&1\\\vdots&\ddots&\vdots\\1&\cdots&0\end{bmatrix}$$

$$T^i_{(2\times2)}=\begin{bmatrix}0&i\\i&0\end{bmatrix}\; T^i_{(3\times3)}=\begin{bmatrix}0&i&i\\i&0&i\\i&i&0\end{bmatrix}\;\cdots\; T^i_{(k\times k)}=\begin{bmatrix}0&\cdots&i\\\vdots&\ddots&\vdots\\i&\cdots&0\end{bmatrix}$$

Fig.4 The basic family of patterns for Chang's load traffic: Cg-1 and Cg-i

The MiMa-algorithm is deterministic. For one matrix ***T****(k*x*k)* it gives one solution $Q_1, Q_2, \ldots, Q_m$ and THR receives quantized values for k from 3 to n. To achieve more precision of simulations in this paper we propose a modification of the family of patterns Chao_i, as it is shown below : for $n = 3$ we have three traffic matrices $\boldsymbol{T}^1$ (see Figure 5 for model Cm-1), for $n = k$ we have k traffic matrices $\boldsymbol{T}^i$ (see Figure 5 for Cm-i). The resulting throughput is the average for n runs for each size $n \times n$.

$$T^1_{(3\times3)}=\begin{bmatrix}0&1&1\\1&0&1\\1&1&0\end{bmatrix}\Rightarrow\begin{bmatrix}0&1&1\\1&0&1\\1&1&0\end{bmatrix},\begin{bmatrix}1&0&1\\1&1&0\\0&1&1\end{bmatrix},\begin{bmatrix}1&1&0\\0&1&1\\1&0&1\end{bmatrix}$$

$$T^i_{(k\times k)}\Rightarrow\begin{bmatrix}0&i&\cdots&i\\i&0&\cdots&i\\\vdots&&\ddots&\vdots\\i&\cdots&i&0\end{bmatrix},\ldots,\begin{bmatrix}i&i&\cdots&0\\0&i&\cdots&i\\\vdots&&\ddots&\vdots\\i&\cdots&0&i\end{bmatrix}$$

Fig.5 The family of patterns for modified Chang's load traffic: Cm-1 and Cm-i

4. Computer simulation

The transition from the formal model to the executive program is carried out as in [12]. The source codes have been created using the program packet Vfort [13] - free access from Institute of mathematical modeling of Russian Academy of Sciences. The source codes have been compiled with means of the grid-cluster BG01-IPP of the Institute IICT- BAS (www.grid.bas.bg). The resulting code is executed locally in the grid-cluster. The operation system is Scientific Linux release 6.5 (Carbon), kernel 2.6.32-431.20.3.e16.x86 64. We used the following grid-resources: up to 16 CPU (2 blades), 32 threads, 2GB RAM. The main restriction is the time for execution (< 72 hours).

Figure 6 shows the results from computer simulation of the MiMa-algorithm with input data by basic Chang and modified Chang pattern. Sizes of the crossbar matrix used for these simulations range from 3x3 to 130x130. Smoothing of the THR is clearly visible. The curve of calculation time is not only smoothed, but also the time is significantly reduced. Thus, we conclude that our modification of the basic family of patterns enables us to obtain more precise results in the simulations of MiMa-algorithm with respect to the THR and the time for execution.

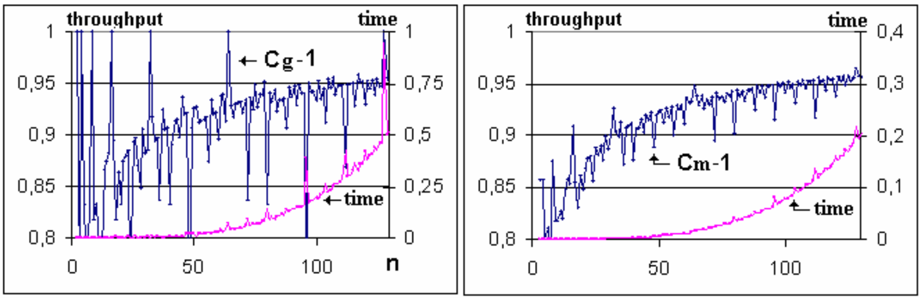


Fig.6 THR and time for basic Chang's load traffic Cg-1 and modified Cm-1

Figure 7 shows the results from computer simulation of the MiMa-algorithm with input data Cm-1, Cm-10, Cm-100. Sizes of the crossbar matrix used for these simulations range from 3x3 to 130x130. To simplify the notations in the figures, the modified pattern is denoted as Cm-j for j=1,10,100… It is shown that the time of execution increases linearly with increasing of the pattern index j. The THR also increases.

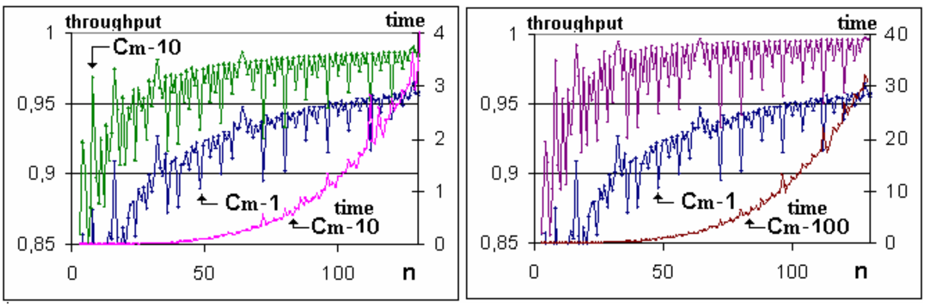


Fig.7 THR and time for modified Chang's load traffic Cm-10 and Cm-100.

This rise the question of whether the 100% THR is a bound for the family of curves for Cm-j when j tends to infinity. In order to answer this question, further simulations are needed.

Figure 8 shows results from computer simulation of the MiMa-algorithm with input patterns Cm-1000 and Cm-10000. A main restriction is the time for execution. The grid-time used for this simulation varies from 58 hours up to 70 hours for Pattern j=10000. The dimension n varies from 3×3 to 100×100 and n simulations (runs) for each size (n×n) of pattern Cm-10000 are executed.

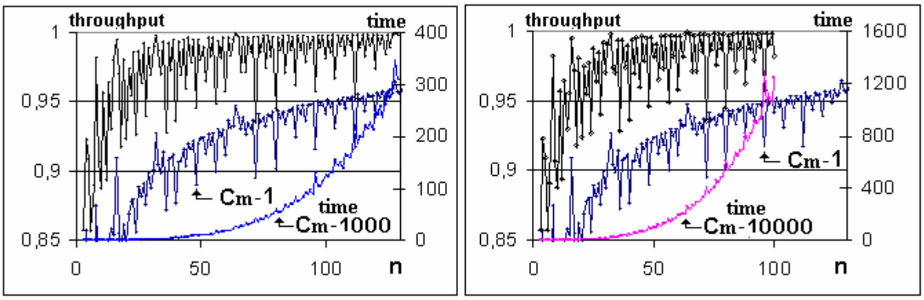


Fig.8 THR and time for modified Chang's load traffic Cm-1000 and Cm-10000.

It is seen that values of throughput for Cm-j are approaching the value of maximum of 100%, when j ≥ 1000 and $n \geq 64$. As a whole the results obtained are consistent with the expectation that the THR approaches the maximum value of 100%.

5. Conclusion

The main results of the paper include determining of an upper bound of the throughput for MiMa-algorithm under a modified family of patterns. The bound of the throughput of the algorithm approaches the maximal possible throughput value equal to 100%. This is achieved at the expense of an increased time of the algorithm execution. The further investigations should be orientated to optimize the time of execution through parallel computation of operations in MiMa-algorithm.

Acknowledgement

The work reported in the paper were supported by the project "Advanced Computing for Innovation" (AComIn), grant No. 316087, funded by the Capacity Programme (Research Potential of Convergence Regions) FP7.

References

1 Chao, H., B. Lui, "High performance switches and routers". John Wiley & Sons, 2007.

2 Hu, B., K. Yeung, C. He. On Iterative Scheduling for Input-queued Switches with a Speedup of 2-1/N. Proceedings of 15th IEEE Int. Conf. HPSR 2014, July 1-4, 2014, Vancouver, Canada, pp.26-31.

3 Chen, T., J. Mavor, Ph, Denyer, D. Renshaw. Traffic routing algorithm for serial superchip system customisation. IEE Proc.- E.part, vol. 137, no.1. pp.65-73, Jan 1990.

4 Anderson, T., S. Owicki, J. Saxe, and C.Thacker. High speed switch scheduling for local area networks. ACM Trans. Comput. Syst., vol. 11, no.4, P.319-352, Nov. 1993.

5 McKeown, N., "The iSLIP Scheduling Algorithm for Input-Queue Switches," IEEE/ACM Transactions on Networking, vol. 7, no. 2, pp. 188-200, April 1999.

6 Chang, H.J., G. Qu, S.Q. Zheng, "Performance of CTC(N) Switch under Various Traffic Models," In: Recent Advances in Computer Science and Information Engineering. Springer Berlin Heidelberg, pp. 785-793, 2012.

7 Tashev, T., V. Monov. Modeling of the Hotspot Load Traffic for Crossbar Switch Node by Means of Generalized Nets. Proceedings of 6th IEEE International Conference Intelligent Systems (IS)" 6-8 September 2012, Sofia, Bulgaria. pp.187-191.

8 Chang C-S., D-S. Lee, Y-S. Jou. Load Balanced Birkhoff-von Neumann Switches. Proceedings of IEEE Workshop on HPSR 2001, May 29-31, Dallas, USA, 2001. pp. 276-280.

9 Atanassov, K. Generalized Nets and System Theory. Acad. Press "Prof. M. Drinov", Sofia, Bulgaria, 1997.

10 Gochev, V., P. Gocheva, ".NET implementation of generalized nets. Object-oriented approach". Sofia: College of Telecommunications and Post Publishing House, 2012 (in Bulgarian).

11 Shahpazov, G., L. Doukovska. Generalized net model of internal financial structural unit's functionality with intuitionistic fuzzy estimations. Proc. of the 17th International Conference on Intuitionistic Fuzzy Sets, 2013, Sofia, Bulgaria. Notes on Intuitionistic Fuzzy Sets (NIFS), vol. 19, №3, pp. 111-117.

12 Tashev, T. An estimation of time required for modeling of an algorithm calculate a non-conflict schedule for crossbar switch node by means of grid-structure. Int. J. "Information Theories and Applications", vol. 19, no 2, pp.149-154, 2012.

13 Vabishchevich, P. VFort. http://www.nomoz.org/site/629615/vfort.html (last checked Apil 14, 2015).