HIERARCHICAL TWO LAYERS CONTROL COMMUTATOR FOR IMPLEMENTATION OF FULLY NON-CONFLICT SCHEDULE

Kiril Kolchakov, Vladimir Monov

***Abstract**: In this paper, an algorithm is developed for a full non-conflict schedule obtained through the activation of diagonals of sub- matrices in switching nodes of type Crossbar. The size of sub matrices in the Crossbar is optimized. Finite automats are designed for two layers control of sub-matrices a comparison with known algorithms with diagonal activation of connection matrix is done.*

***Keywords**: Network nodes, Crossbar switch, Conflict elimination, Packet messages, finite automat.*

***ACM Classification Keywords**: C.2.1 Network Architecture and Design, C.4 Performance of Systems Conference topic: Information Modeling, Distributed and Telecommunication Systems*

Introduction

The traffic via Crossbar switching nodes is casual and depends on the users. The formulation of the conflict issue during operation of the switching nodes is as follows: the dimensions of the switches in the switching nodes are N x N, where N sources of packet messages are connected to N receivers via the switch of the switching node. The switching node traffic is random by nature and depends on the users. Conflicts are available in the following two cases:

- When one source of message requests communication to two or more message receivers;
- When one message receiver receives communication requests from two or more message sources.

The evasion of conflicts is directly related to the switching node performance.

The status of the switch of the switching node is represented with the so called connection matrix. For N x N dimensional switch the dimension of the connection matrix T is N x N also, where every element T_{ij} = 1 if the connection request from i- source to j- receiver exists. In the opposite case T_{ij} = 0.

A conflict situation arises if any row of the connection matrix has more than a single 1, which corresponds to the case when one source requests a connection with more than one receiver. The presence of more than a single 1 in any column of the matrix T also indicates a conflict situation, it means that two or more sources have requested a connection with the same receiver [Kolchakov and Tashev, 2012], [Kolchakov K., 2012].

Description of algorithm ADAJS (Algorithm with diagonal activations of joint sub-switching matrices)

The connections matrix T with N x N size, where N is being the degree of two, is divided into sub matrices (S) with dimension n x n, (n also is a degree of two), i.e.:

$$T = [\, S_{ij} \,],\ i = 1\text{- } n,\ j = 1\text{- } n.$$

The sets of sub matrices located along the main diagonal are processed simultaneously in each of the diagonals. For submatritces in diagonals parallel to the main one, the principle of reconciliation is used. The idea of synthesis of the algorithm ADAJS (Algorithm with diagonal activations of joint sub-switching matrices) is based on the knowledge that the diagonal sub matrices with requests for service in the matrix T are non-conflict in the diagonal where they are located [Kolchakov and Monov, 2013].

There are diagonals with sub matrices of requests that are non-conflict to one another.

Figure 1 shows joint couple of non-conflict diagonals with sub matrices of requests for service and the main diagonal of sub matrices that cannot be jointed with anyone else.

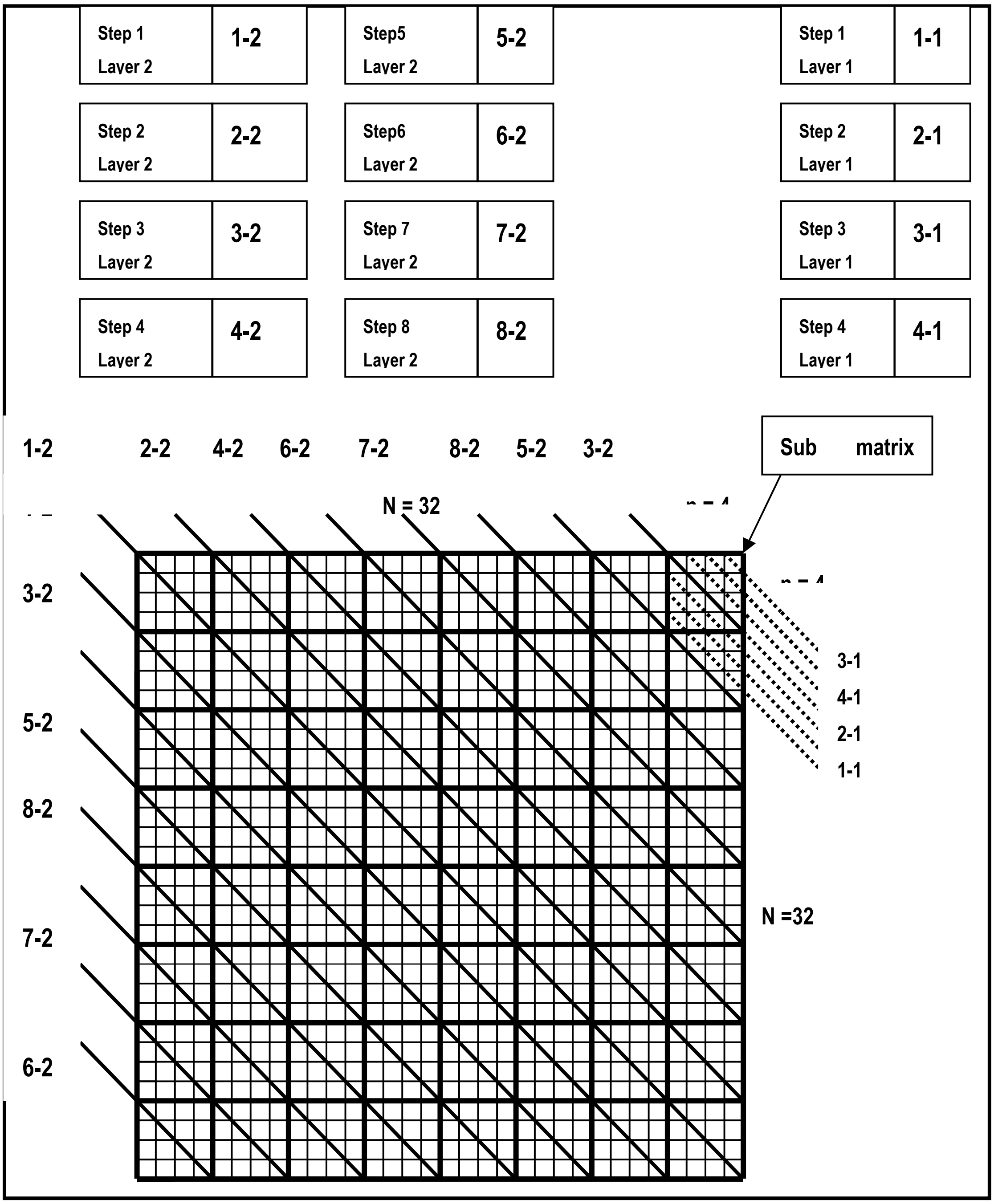

Figure 1. Diagonal activation of joint sub matrices (Layer 2+ Layer 1)

The whole process of the implementation of ADAJS algorithm for obtaining a non-conflict schedule is divided into steps. The first step refers to the main diagonal sub matrices processed simultaneously and without conflict. The next steps are related to the reconciliation of the diagonals parallel to the main diagonal by pairs (Figure 1). The analytical description of the steps shown on Figure 1 is as follows:

Step1 : $S_{11}, S_{22}, S_{33}, S_{44}, S_{55}, S_{66}, S_{77}, S_{88}$ **Step 5 :** $S_{17}, S_{28}, S_{31}, S_{42}, S_{53}, S_{64}, S_{75}, S_{86}$

Step2 : $S_{81}, S_{12}, S_{23}, S_{34}, S_{45}, S_{56}, S_{67}, S_{78}$ **Step 6 :** $S_{61}, S_{72}, S_{83}, S_{14}, S_{25}, S_{36}, S_{47}, S_{58}$

Step3 : $S_{18}, S_{21}, S_{32}, S_{43}, S_{54}, S_{65}, S_{76}, S_{87}$ **Step 7 :** $S_{51}, S_{62}, S_{73}, S_{84}, S_{15}, S_{26}, S_{37}, S_{48}$

Step4 : $S_{71}, S_{82}, S_{13}, S_{24}, S_{35}, S_{46}, S_{57}, S_{68}$ **Step 8 :** $S_{41}, S_{52}, S_{63}, S_{74}, S_{85}, S_{16}, S_{27}, S_{38}$

T = [S_{ij}] , i = 1 - 8, j = 1 - 8

The size (n) of the sub matrix determines the number of steps (I) as follows

$$I = N/n. \tag{1}$$

For N = const., I = f (n), where $1 < n \le N / 2$.

Sub-matrix optimal size

The goal is to determine the optimal size of the sub matrix related to performance of the software model SMADAJS. A clear minimum of the operating time (TW) is seen for n = 4 for N = 128, 256, 512, 1024 and 2048. It can be concluded that for size N up to 2048, the optimal sub matrix size is n opt = 4. In order to reduce CPU work controlling the requests execution in a switching node of type Crossbar, it is appropriate to synthesize a finite automat. It will take care to handle the individual sub matrices. Then CPU will be limited to initialize the finite automat during the work of the algorithm ADAJS for the respective sub matrix only.

In our case the optimal size of the sub matrix is n opt = 4. Therefore, the finite automat shell replaces CPU for a sub matrix 4 x 4. Figure 2 shows the four steps of the algorithm for sub matrix 4 x 4 corresponding to the four states of the finite automat - A1 to A4. A state A0 should be available for the initial reboot of the finite automat. In state A0 all requests in the sub matrix are not allowed.

Realization of ADAJS algorithm via hierarchical two layers control with finite automata for commutator of type Crossbar

The algorithm ADAJS contains naturally hierarchical two layers control of type "master – slave" itself. Layer 1 (slave) is the layer where the sub matrices are processed. Layer 2 (master) is the layer where the diagonals of sub matrices are processed.

In our case a Moore finite automat is suitable, because each state corresponds to an output signal that is not affected by the input word, as in the case of Mealy automat [M.Schaller and K. Svozil, 1996].

Figure 3 shows the state transition graph of the synthesized Moore automat for control of the sub matrix presented in Figure 2. Using Z_1 signal, the automat passes through the states from A_0 to A_4 and each state A_1 to A_4, corresponds to an output signal $St._1$ (Step1) to $St._4$ (Step4) which controls the connection sub matrix directly.

Using Z_2 signal, the automat goes to A0 state with an output signal R which is an indication for CPU switching node that the automat is free and could be started at any time.

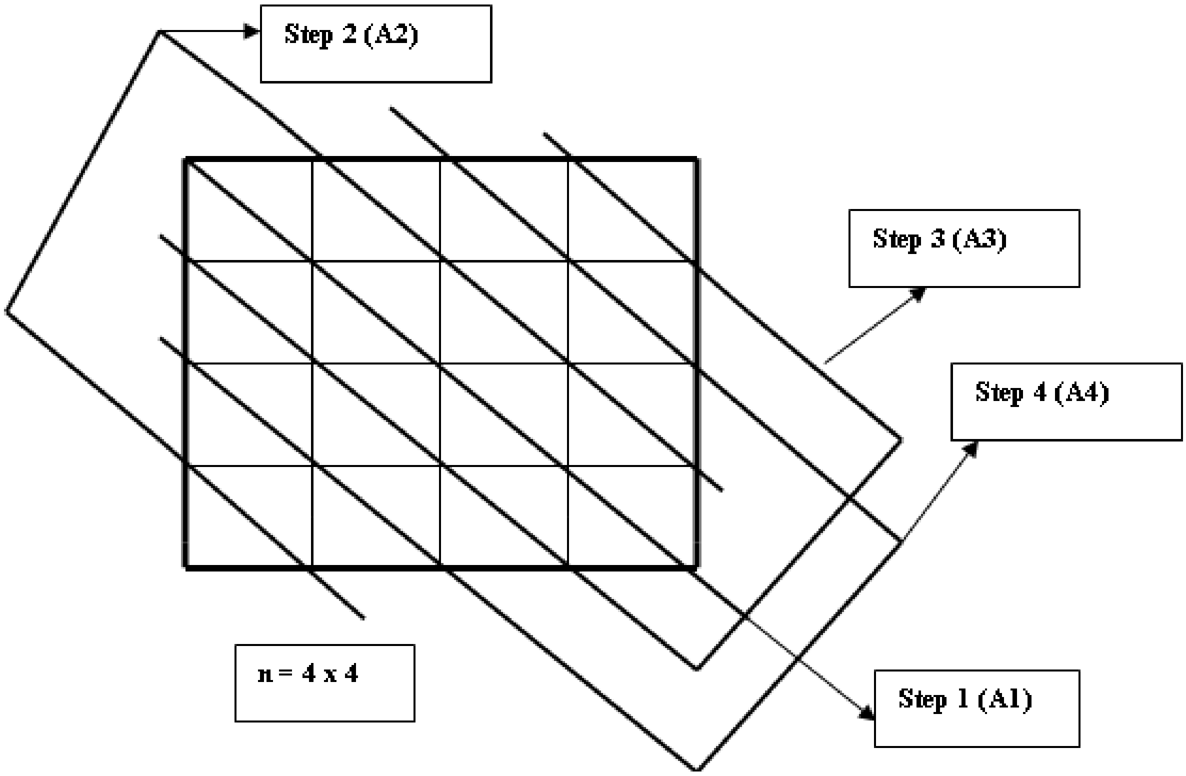

Figure 2. Sub matrix 4 x 4(optimal size) - (Layer 1)

Each sub matrix is controlled by Moore finite automat with five states (Figure 3). This is the first layer control (Layer 1).

Figure 4 shows a graph of transitions of Moore finite automat. This finite automat controls the joint diagonals of the sub matrices described in steps 1 through 8 on the second level (Layer 2) of Figure 1.

The difference between the graph of transitions of the automaton of Layer 1 (Figure 3) and that of Layer 2 (Figure 4) is the number of states.

We denote by τ the residence time of finite automat of Layer 1 in each of its states. For finite automat on the second level (Layer 2) this time is n.τ and since n opt = 4, it is obtained **4τ**.

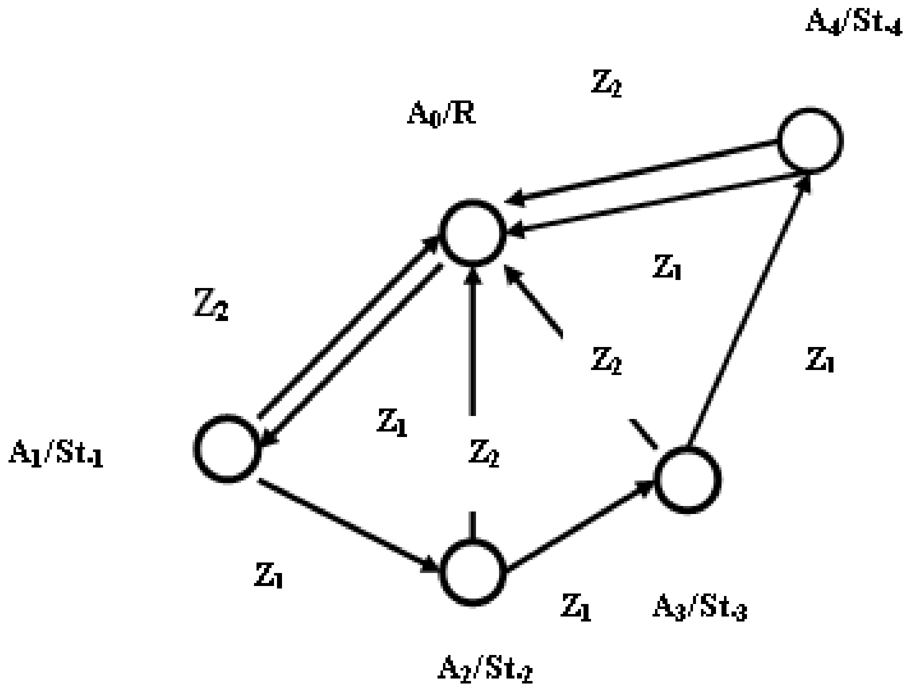

Figure 3. Graph of the transitions (Layer 1)

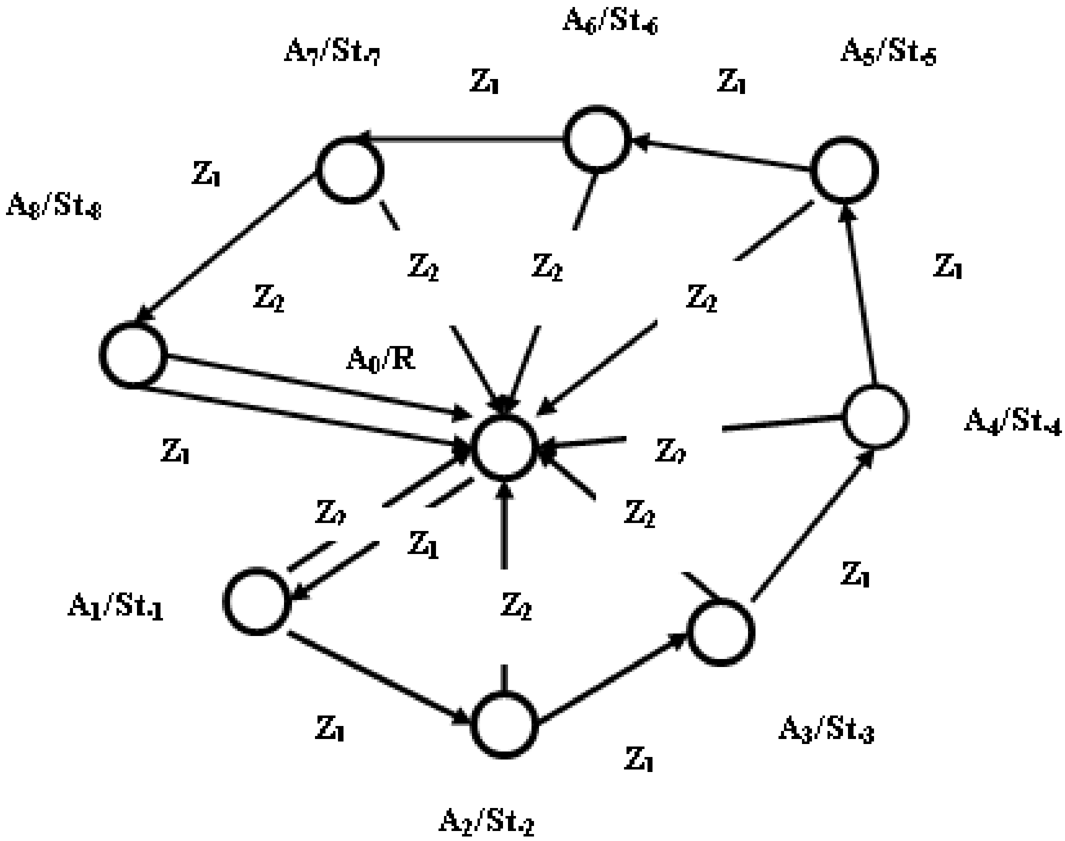

Figure 4. Graph of the transitions (Layer 2)

Comparison with other algorithms, with diagonal activation of the connection matrix

Table 1 represents a comparison between the results of algorithms with a diagonal matrix activation of connection matrix in terms of number of control levels, number of states of finite automata used in the implementation and operation time.

Table 1. Comparison between algorithms with a diagonal matrix activation of connection matrix

Algorithm	Layers	FA - states	operation time
ADA	1	2N	(2N-1).т
AJDA	1	N+1	N.т
ADAJS	2	5 (Layer 1), N/4+1 (Layer 2)	N.т

The advantage of the hierarchical two layers control for ADAJS is that the number of states of the finite automata of the second level is four times less than the states of AJDA finite automata.

In terms of running time ADAJS is almost twice faster than ADA and equal to AJDA.

Conclusion

The hiearchical two layers control commutator of type Crossbar by finite automata releases the communication processor from the requests processing tasks and ensures fully non-conflict requests execution. This control requires number of states of layer 2 control four times less than AJDA and eight times less than ADA which are with one layer control.

Bibliography

[D. Kim, K. Lee and H. Yoo, 2005] “A Reconfigurable Crossbar Switch with Adaptive Bandwidth Control for Networks-on-Chip”, IEEE International Symposium on Circuits and Systems, 2005.

[Kolchakov and Monov, 2013] Kolchakov K, Monov V. An Algorithm for Non – Conflict Schedule with Diagonal Activation of Joint Sub Matrices, 17-th International Conference on "DISTRIBUTED COMPUTER AND COMMUNICATION NETWORKS (DCCN-2013): "CONTROL, COMPUTATION, COMMUNICATIONS, Moscow, October 07-10, 2013, pp. 180-187, ISBN 978-5-94836-366-0.

[Kolchakov and Tashev, 2012] Kolchakov K., Tashev T. “An Algorithm of Non-conflict Schedule with Joint Diagonals Activation of Connectivity Matrix”, Proceedings of the International Conference on Computer Systems – CompSysTech’12, 22-23 June 2012, Ruse, Bulgaria, ACM PRESS, ICPS, VOL.630, pp.245-250, ISBN 978-1-4503-1193-9.

[Kolchakov K., 2011] Kolchakov K., “An Algorithm Synthesis of Non-Conflict Schedule by Diagonal Connectivity Matrix Activation” Proceedings of the International Conference AUTOMATICS AND INFORMATICS`11, John Atanasoff Society of Automatics and Informatics, Bulgaria, Sofia 03.10-07.10.2011., pp. B-247 – B251, Proceedings ISSN 1313-1850, CD ISSN 1313-1869.

[Kolchakov K., 2012] Kolchakov K. Examination on Algorithms for Non Conflict Schedule with Diagonal Activation in Case of a Large Size Switching Matrix. International Conference Automatics and Informatics `12 03-05.10.2012 Sofia, Bulgaria. Proceedings CD: ISSN 1313 – 1869. pp.341-344.

[M.Schaller and K. Svozil, 1996] "Automaton Logic", International Journal of Theoretical Physics, Vol.35, No. 5, pp. 911-940, 1996, Plenum Publishing Corporation-Springer.

[P. Wanjari and A. Chonhari, 2011] "Implementation of 4x4 crossbar switch for Network Processor", International Journal of Emerging Technology and Advanced Engineering, Website: www.ijetae.com (ISSN 2250 – 2459, Volume 1, Issue 2, December 2011).

[Tashev T., 2011] Modelling throughput crossbar switch node with nonuniform load traffic. Proceedings of the International Conference "Distributed Computer and communication networks DCCN 2011", October 26-28, 2011, Moscow, Russia. R&D Company "Information and Networking technologies", Moscow, Russia, pp.96-102. (in russian)

[Ташев Т., 2013] Моделирование пропускной способности MiMa-алгоритма для пакетного коммутатора при входящем трафике типа „горячей точки". Proceedings of the International Conference "Distributed Computer and communication networks: Control, Computation, Communications (DCCN-2013)", October 7-10, 2013, Moscow, Russia. JSC "TECHNOSPHERA", Moscow, Russia, pp.257-264.

Authors' Information

Kiril Kolchakov *- Department "Modelling and Optimization", Institute of Information and Communication Technologies – Bulgarian Academy of Sciences, "Acad. G. Bonchev" bl. 2, Sofia 1113, Bulgaria; e-mail: kkolchakov@iit.bas.bg*

Major Fields of Scientific Research: Distributed Information Systems Design, Methods and tools for net models researches.

Vladimir Monov *- Department "Modelling and Optimization", Institute of Information and Communication Technologies – Bulgarian Academy of Sciences, "Acad. G. Bonchev" bl. 2, Sofia 1113, Bulgaria; e-mail: vmonov@iit.bas.bg*

Major Fields of Scientific Research: Distributed Information Systems Design.