Buffering in Crosspoint-Queued Switch

Jelena Cvorovic, Igor Radusinovic, *Member, IEEE*, Milutin Radonjic, *Member, IEEE*

***Abstract* — In this paper we presented new performance analysis of the *crosspoint-queued* (CPQ) switch with N^2 crosspoint buffers size one under the Bernoulli i.i.d. incoming traffic. We modeled analysed switch with discrete Markov chain. Using that model for finite N, we made program in MATLAB for calculating switch performance - throughput, loss probability and average delay. These results are compared with simulation results for different work-conserving algorithms under the same traffic and results we found in literature.**

***Index Terms* — average delay, crosspoint-queued switch, loss probability, Markov chain, throughput.**

I. INTRODUCTION

In this paper, we analysed *crosspoint-queued* (CPQ) switch performance. The CPQ switching fabric is a buffered crossbar with crosspoint cells, as illustrated in Fig. 1. The incoming packets arrive directly to the crosspoint buffers, i.e. packets are not queued at the input and there is no any transfer of control information between inputs and crosspoint buffers. This means that there is no need for complex input scheduling as in crossbar switch with Virtual Output Queuing (VOQ) [1] or control communication between linecards and scheduling as in Combined Input and Crosspoint Queued (CICQ) switch [2]-[3], which represents one of the main advantages of this solution. Due to simplicity of this solution, buffers and switching fabric could be implemented on the single chip, what makes hardware implementation much easier.

In CPQ architecture illustrated in Fig. 1, the incoming packet originated from certain input and addressed to appropriate output is queued in crosspoint buffer. In every time slot, scheduler chooses one of the non-empty crosspoint buffers and forwards its head of line packet to the output. This selection should be based on different work-conserving algorithms [4] (OQ, LQF, RR, FRRM, RAND.). These algorithms are called work-conserving because each output always services a buffer whenever one of the buffers destined to it is non-empty. In LQF (Longest Queue First) algorithm based switch [5], each output schedules the longest queue in its column, resolving ties uniformly at random. The Round Robin algorithm [6] alternately serves queues without regard for traffic priority. FBRR (Frame Based Round Robin) [4] algorithm considers all crosspoint buffers in a round-robin order and when find non-empty buffer, services it until it is empty. FBRR is useful under some non-uniform arrival patterns when ERR (Exhaustive Round Robin) algorithm [7] could introduce starvation of low-level incoming traffic. In CQ switch with random algorithm (RAND), each output picks uniformly at random a nonempty buffer to serve.

Jelena Cvorovic, T-com, Podgorica, Montenegro (phone: 38267249949, e-mail: jelena.cvorovic@telekom.me)

Igor Radusinovic Faculty of Electronic, University of Montenegro, Podgorica, Montenegro (e-mail: igorr@ac.me)

Milutin Radonjic Faculty of Electronic, University of Montenegro, Podgorica, Montenegro (e-mail: micor@t-com.me)

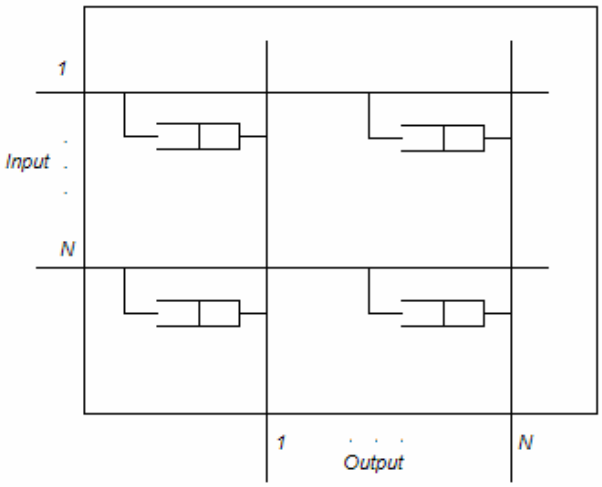

Fig. 1. Crosspoint-queued switch

Let N be number of switch inputs/outputs and $B=1$ size of crosspoint buffer (Fig. 1). We assumed that time is slotted into fixed-size *time-slots*, and that each of the time-slots is divided in two phases: departure and arrival. It is important to notice that in our case first comes departure phase, and then arrival phase. As a first step in analyzing behavior of CPQ switch with buffers larger than one cell, we presented new model for analyzing switch performance - throughput, loss probability and average delay.

Closed-form expressions for the throughput and average delay are already obtained in [8] using Z-transforms. Switch throughput in [8] is defined as the limiting ratio of the cumulative number of packets entering any crosspoint buffer by the cumulative number of arrived packets. For small crosspoint buffers of size one, under uniform Bernoulli i.i.d arrivals of load p, and any work-conserving scheduling algorithm, the switch throughput in [8] is:

$$Th=\frac{1}{p}\left[1-\frac{q}{1+\sum_{m=1}^{N-1}\binom{N-1}{m}q^{-m}\prod_{j=1}^{m}(q^{-j}-1)}\right] \qquad (1)$$

where $q = 1 - \frac{p}{N}$. The crosspoint throughput Th_{ij} is equal to the probability that a packet arriving to the crosspoint buffer is not dropped. Equivalently, if P_{ij} denotes the steady-state probability that crosspoint buffer keeps packet before the arrival phase, then crosspoint throughput is:

$$Th_{ij} = 1 - P_{ij} \tag{2}$$

An average delay is defined as ratio of probability P_{ij} and the probability that packet arrives and is absorbed in the crosspoint buffer [8]:

$$W = \frac{P_{ij}}{\frac{p}{N} Th_{ij}} = \frac{1 - Th_{ij}}{\frac{p}{N} Th_{ij}} = \frac{N}{p}\left(\frac{1}{Th} - 1\right) \tag{3}$$

Loss probability is the ratio of lost traffic relative to the input traffic.

Comparing our analytic and simulation results with results in [8], it appears that they are very close (Table 1, 2 and 3 in Section IV). We believe that the most significant advantages of our analysis lie in simplicity and extensibility for larger buffer.

The structure of this paper is as follows. In Section II, we introduce mathematical model of CPQ switch, based on column stochastic Markov matrix. Then, performance expressions for throughput, loss probability and average delay time are presented in Section III. Section IV provides simulation results for LQF, RR, ERR, FBRR and RAND algorithm as well as comparison of our analytical results, results of simulation and results in [8]. Finally, we conclude the paper in Section V.

II. MATHEMATICAL MODEL

A. State transition diagram

We modeled CPQ switch under the Bernoulli i.i.d (independent and identically distributed) arrival traffic. In case of Bernoulli i.i.d traffic, it is assumed that the arrival time of a new packet, after the completion of the previous packet, is independent and identically distributed; the traffic is independent across inputs. For that kind of traffic, we modeled *N* crosspoint buffers size one as one buffer size *N* and after that we modeled this buffer with discrete Markov chain $M^N/M/1/N$ with $N+1$ states. In this queue, at most *N* packets could arrive and at most one packet could leave during one time step. Therefore, the queue size can increase by more than one, but can only decrease by one in each time step. Probability of arrival is *p* and because of uniform kind of traffic, arrival probability on single input is p/N. Departure probability is 1 (one packet leaves queue with probability 1) if buffer is nonempty.

A Markov chain stays in a particular state for a certain amount of time, called the hold time. In a discrete-time Markov chain, the hold time assumes discrete values. As a result, changes in the states occur at discrete time values. State transition diagram for discrete $M^N/M/1/N$ Markov chain that we modeled is given in Fig. 2.

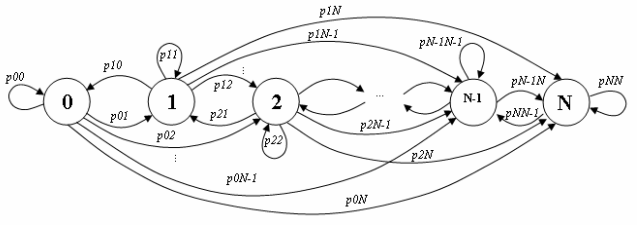

Fig. 2. State transition diagram

Probability that packet moves from state *0* to state *j* is p_{0j} (j=0, 1,…, N). In transition matrix, it is first column.

$$p_{oj} = \binom{N}{j}\left(\frac{p}{N}\right)^j \left(1 - \frac{p}{N}\right)^{N-j} \tag{4}$$

Transition probabilities from the state ‘1” are the same like those from state “0”, since we made assumption that one packet leaves state for sure, i.e. departure probability is 1.

$$p_{1j} = p_{0j} \tag{5}$$

Transition probabilities from state *i* (*i*=1,..., N) to state *j* (j=0, 1,…, N) for $i \le j+1$ are:

$$p_{ij} = \binom{N-i+1}{k}\left(\frac{p}{N}\right)^k \left(1 - \frac{p}{N}\right)^{N-i+1-k} \tag{6}$$

where k=0,..., N-i+1.

For the last state transition probabilities are:

$$p_{NN-1} = 1 - p_{NN} = 1 - \frac{p}{N} \tag{7}$$

B. State transition matrix

State transition matrix **P** is lower $(N+1)\times(N+1)$ *Hessenberg* matrix in which all the elements $p_{ij} = 0$ for $i > j+1$. For finite N transition matrix **P** is:

$$P = \begin{pmatrix} \left(1-\frac{p}{N}\right)^N & \left(1-\frac{p}{N}\right)^N & 0 & \cdots & 0 & 0 \\ \binom{N}{1}\frac{p}{N}\left(1-\frac{p}{N}\right)^{N-1} & \binom{N}{1}\frac{p}{N}\left(1-\frac{p}{N}\right)^{N-1} & \left(1-\frac{p}{N}\right)^{N-1} & \cdots & 0 & 0 \\ \binom{N}{2}\left(\frac{p}{N}\right)^2\left(1-\frac{p}{N}\right)^{N-2} & \binom{N}{2}\left(\frac{p}{N}\right)^2\left(1-\frac{p}{N}\right)^{N-2} & \binom{N-1}{1}\frac{p}{N}\left(1-\frac{p}{N}\right)^{N-2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \left(1-\frac{p}{4}\right)^2 & 0 \\ \binom{N}{N-1}\left(\frac{p}{N}\right)^{N-1}\left(1-\frac{p}{N}\right) & \binom{N}{N-1}\left(\frac{p}{N}\right)^{N-1}\left(1-\frac{p}{N}\right) & \binom{N-1}{N-2}\left(\frac{p}{N}\right)^{N-2}\left(1-\frac{p}{N}\right) & \cdots & 2\frac{p}{N}\left(1-\frac{p}{N}\right) & 1-\frac{p}{N} \\ \left(\frac{p}{N}\right)^N & \left(\frac{p}{N}\right)^N & \left(\frac{p}{N}\right)^{N-1} & \cdots & \left(\frac{p}{N}\right)^2 & \frac{p}{N} \end{pmatrix} \tag{8}$$

State transition matrix **P** has some peculiar properties:

1. The number of rows is equal to number of columns i.e. **P** is square matrix.
2. All the elements of **P** are real numbers i.e. **P** is real matrix.
3. For all values of *i* and *j*, $0 \le p_{ij} \le 1$ i.e. **P** is

nonnegative matrix.

4. Sum of each column is 1.
5. Magnitudes of all eigenvalues obey the condition $|\lambda_i| \leq 1$.
6. At least one of the eigenvalues of **P** equals 1.

From all these properties, we conclude that the transition matrix is column stochastic or *Markov* matrix [9].

III. PERFORMANCE

In this section, we analyse switch performance: throughput, loss probability and average delay.

The average input traffic $N_a(in)$ is given from the binomial distribution:

$$N_a(in) = Na = N\frac{p}{N} = p \tag{9}$$

Now we are interested in estimating the rate of packets leaving the queue. We call this rate the average output traffic $N_a(out)$ or throughput of queue.

$$N_a(out) = T_h = \sum_{i=1}^{N} c \cdot s_i = \sum_{i=1}^{N} s_i = 1 - s_0 \tag{10}$$

The unit of throughput in the above expression is packets/time step. The throughput could be expressed in unit packets/second:

$$Th' = \frac{Th}{T} \tag{11}$$

where T is duration of time step given in seconds.

Departure probability c is 1 (one packet lefts queue with probability 1). ***s*** is a steady-state distribution vector i.e. eigenvector of state transition matrix **P** that corresponds to unity eigenvalue.

We used the traffic conservation principle to calculate cell loss probability. Cell loss probability is given by:

$$p_b = N_a(lost)/N_a(in) = (N_a(in) - N_a(out))/N_a(in) = (p - T_h)/p \tag{12}$$

We used Little's theorem [10] to calculate an average delay, which is the average number of time slots that packet spends in the queue.

$$W = Q_a / T_h \tag{13}$$

The average queue size is given by the equation:

$$Q_a = \sum_{i=0}^{N} i \cdot s_i \tag{14}$$

Notice that each output provides services independently of other outputs. Therefore, without loss of generality we focused on one column and considered its performance.

Program in MATLAB, which we used to get performance, is based on equations (10), (12) and (13). We used function eig(P) to obtain matrices **X** and **D** [9].

$$[\mathbf{X}, \mathbf{D}] = eig(P)$$

Matrix **X** is matrix of eigenvectors:

$$\mathbf{X} = [\mathbf{x_1} \quad \mathbf{x_2} \quad \ldots \quad \mathbf{x_N}] \tag{15}$$

Eigenvector **x** satisfies the equation

$$\mathbf{Px} = \lambda \mathbf{x} \tag{16}$$

where λ is eigenvalue of **P**. Eigenvalues could be evaluated from

$$\det|\mathbf{P} - \lambda \mathbf{I}| = 0 \tag{17}$$

where **I** is unit $N \times N$ matrix.

D is diagonal matrix whose diagonal elements are the eigenvalues of **P**.

$$\mathbf{D} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_N \end{pmatrix} \tag{18}$$

State vector ***s*** defines as normalized eigenvector (column of matrix **X**) which corresponds to eigenvalue 1 in matrix **D** (if there is no value 1 in matrix **D** then we take closes value to that value, in order to get eigenvector in matrix **X**). State vector is used to calculate performance.

It is important to note that in CPQ switches the performance of each output can be analyzed independently, because the CPQ switch architecture isolates packets addressed to each output. As long as traffic from input *i* to output *j* is independent of other traffic processes, we will be able to study CPQ switches based on column performance.

In [8] performance is obtained using Z-transforms. Considering different defining of throughput and work environment in our paper and in [8], we made some adaptation to be able to compare these results. Table 1 confirms that throughputs are the same in both cases. Now we want to compare results for average delay. In [8] packets could arrive and leave in the same time step. In other words, it is possible for packet not to wait what is not possible in our switch performance-analyzing scenario. Therefore, due to comparison we increased average delay [8] by one. It is obvious from Table 3 that obtained results are identical to modified results from [8].

IV. SIMULATION RESULTS

Now we want to compare the results which we have got in MATLAB with simulation results and results from literature, for the 4×4 switch. The simulation is performed under uniform Bernoulli i.i.d arrival for different values of p - from 0.01 to 1. Time is divided into equal time slots that correspond to time required for transferring one cell. Each simulation is performed for a million time slots.

In Table 1, Table 2 and Table 3 are shown results for throughput, loss probability and average delay, respectively. We compared results for p=0.01, 0.1003, 0.3001, 0.4999, 0.6999, 0.91, 0.9502, 0.99 and 1. It is obvious from tables that with crosspoint buffer of one packet length and uniform traffic, CPQ switch for all simulated algorithms invoke almost identical performance, according to [8].

For example, we can consider arrival probability p=1. In this case, throughput is identical in our analytical model and in [8]. LQF algorithm gives quite lower results, while RR, ERR and FBRR give quite larger results then analytical. Random model provide the highest results for p=1, but that difference is still low. Also, we can see

TABLE 1: THROUGHPUT (packets/time step)

p	0.01	0.1003	0.3001	0.4999	0.6999	0.91	0.9502	0.99	1
Analytic	0.009999	0.100198	0.296937	0.483174	0.648636	0.788288	0.810325	0.830605	0.835461
LQF	0.01003	0.10018	0.29692	0.48316	0.64883	0.78813	0.8103	0.83046	0.835433
RR	0.01003	0.10018	0.29692	0.48318	0.64878	0.78816	0.81033	0.83056	0.835463
ERR	0.01003	0.10018	0.29692	0.48318	0.64878	0.78816	0.81033	0.83056	0.835463
FBRR	0.010025	0.100176	0.296919	0.483184	0.648781	0.788162	0.810328	0.830561	0.835463
RAND	0.010025	0.10018	0.296932	0.483137	0.648786	0.788192	0.810243	0.830528	0.835519
[8]	0.009999	0.100198	0.296937	0.483174	0.648636	0.788288	0.810325	0.830605	0.835461

TABLE 2: LOSS PROBABILITY

p	0.01	0.1003	0.3001	0.4999	0.6999	0.91	0.9502	0.99	1
Analytic	0.000009	0.001017	0.010541	0.033459	0.073245	0.13375	0.147206	0.161005	0.164539
LQF	0.00005	0.0010097	0.0105974	0.0335421	0.0730347	0.133902	0.1472163	0.1611688	0.1645673
RR	0	0.001027	0.0105949	0.0334981	0.0730968	0.13387	0.1471892	0.161063	0.164537
ERR	0	0.001027	0.0105949	0.0334981	0.0730968	0.13387	0.1471892	0.161063	0.164537
FBRR	0	0.001027	0,0105949	0,0334981	0,0730968	0,13387	0,1471892	0,161063	0,164537
RAND	0.00005	0.0009922	0.0105507	0.0335932	0.073089	0.133838	0.1472792	0.1610966	0.1644808
[8]	0.000009	0.001017	0.010541	0.033459	0.073245	0.13375	0.147206	0.161005	0.164539

TABLE 3: AVERAGE DELAY (time steps)

p	0.01	0.1003	0.3001	0.4999	0.6999	0.91	0.9502	0.99	1
Analytic	1.003778	1.040591	1.141996	1.276995	1.451688	1.678685	1.726654	1.77536	1.787774
LQF	1.004065	1.039989	1.142253	1.277582	1.451821	1.678727	1.726843	1.775956	1.787863
RR	1.004115	1.039987	1.142388	1.277687	1.451372	1.679431	1.727296	1.776494	1.787684
ERR	1.004115	1.039987	1.142388	1.277687	1.451372	1.679431	1.727296	1.776494	1.787684
FBRR	1,004115	1,039987	1,142388	1,277687	1,451372	1,679431	1,727296	1,776494	1,787684
RAND	1.004065	1.04004	1.142386	1.277534	1.451452	1.67972	1.72662	1.775974	1.787471
[8]	1.003778	1.040591	1.141996	1.276995	1.451688	1.678685	1.726654	1.77536	1.787774

that our analytic results for loss probability and average delay time are identical with those in [8].

Analyzing simulation results, we conclude that loss probability and average delay for CPQ switch with LQF algorithm is quite higher than for other algorithms.

V. CONCLUSION

In this paper we introduced new performance analyses of crosspoint-queued switch with crosspoint buffers of size one. Considering Bernoulli i.i.d. arrival traffic we modeled CPQ switch as discrete Markov chain, and presented results for throughput, loss probability and average delay. We showed that obtained results are tightly close to results presented in [8] and simulation results for different work-conserving scheduling algorithms. Main advantages of our approach in this paper are simplicity and extensibility. Based on this, our goal in near future is to model CPQ switch performance under Bernoulli i.i.d. arrival traffic for larger crosspoint buffer size.

REFERENCES

[1] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch (extended version)," IEEE Transactions on Communications, vol. 47, August 1999.

[2] M. Nabeshima, "Performance evaluation of a combined input-and crosspoint-queued switch," *IEICE Transactions on Communications*, vol. E83-B,, March 2000.

[3] R. Rojas-Cessa, E. Oki, H. Chao, "On the combined input-crosspoint buffered switch with round-robin arbitration", IEEE Transactions on Communications, vol. 53, No. 11, November 2005.

[4] D. Banovic, I, Radusinovic, "Scheduling algorithm for VOQ switches". Int. Journal of Electronics and Communications 2008;62:455-8.

[5] A. Mekkittikul, "Scheduling non-uniform traffic in high speed packet switches and routers", PhD. Thesis, Stanford University, 1998.

[6] E. Shin, V. Mooney, G. Relay, "Round-robin arbitrer design and generation" , Proceedings of the 15th international symposium on System Synthesis, Kyoto, 2002. p. 243-8.

[7] Y. Li, S. Panwar, H. Chao, "The dual round-robin matching switch with Exhaustive Service", Proceedings of IEEE HPSR, Kobe, 2002. p. 58-63.

[8] Y. Kanizo, D. Hay, I. Keslassy, "The Crosspoint-Queued Switch" Technical report TR08-04, Comnet, Technion, Israel

[9] F. Gebali, "Analysis of Computer and Communication networks", Springer Science+Business Media, LLC, New York, 2008

[10] J.D.C.Little, "A proof of the queuing formula $L = \lambda W$ ", Oper. Res. Vol. 9, No. 3, May-June 1961.

[11] P. Giacomazzi, A. Pattavina, "Performance Analysis of the ATM Shuffleout Switch Under Nonuniform Traffic Patterns", IEEE Transactions on Communications, vol. 44, No. 11, November 1996.

[12] Y. Ganjali, A. Keshavarzian, D. Shah, "Cell Switching Versus Packet switching in Input-Queued Switches", IEEE/ACM Transactions on Networking, vol. 13, No. 4, August 2005.