A Novel High Performance Scheduling Algorithm for Crosspoint Buffered Crossbar Switches

X.T. Wang, Y.W. Wang, S.C. Li, P. Li
University of Electronic Science and Technology of China
Chengdu, China

Abstract--**Crosspoint buffered crossbar switches have gained much attention due to simple distributed scheduling algorithms. However, almost all the algorithms proposed for buffered crossbar switches either have unsatisfactory scheduling performance under non-uniform traffic or poor service fairness between input traffic flows. In order to overcome the disadvantages, in this paper we propose a novel efficient scheduling algorithm named MCQF_RR, which takes advantage of the combined information about queue length and service waiting time of queues to perform scheduling, in order to ensure good service fairness. Simulation results show that the proposed MCQF_RR demonstrates good delay performance comparable to existing efficient algorithms under various admissible traffic patterns, while achieve better service fairness under extreme non-uniform traffic.**

Keywords-cross point buffered crossbar switches; scheduling; delay; fairness

I. INTRODUCTION

With the new network applications increasing continually, Internet traffic tends to grow rapidly. To keep pace with the growth demand of Internet traffic, it is necessary to explore high-performance switches. Crossbar-based architecture is the most important switch fabric, because of its internally non-blocking property. Based on buffering strategies in the switch fabric, crossbar-based switch can be classified into internally bufferless crossbar switch and crosspoint buffered crossbar switch. For internally bufferless crossbar switches, virtual output queued (VOQ) structure [1] has attracted much attention because of low-bandwidth requirements. The packets can only be buffered at the input ports and the switch fabric just operates at the line rate. VOQ switches need centralized schedulers to resolve input and output contention [2]. Many scheduling algorithms for VOQ switches have been proposed [3], [4]. Unfortunately, they either have a high scheduling complexity or can’t achieve good performance.

Crosspoint buffered crossbar switches, also called combined input-crosspoint queued (CICQ) switches, have been a solution to overcome the high scheduling complexity [5]. A small buffer is added at each crosspoint in the crossbar fabric. Due to the introduction of crosspoint buffers, the scheduling complexity is dramatically reduced compared with VOQ switches. So far, there have been many scheduling algorithms for CICQ switches and can be classified into three categories. One category is *Round Robin (RR)* based algorithm, such as RR_RR [6] and RR-LQD [9]. They use different pointer updating mechanisms to perform scheduling and achieve 100% throughput under uniform traffic. However, they performs instability under some non-uniform traffic. Another category is weight-based algorithm, such as LQF_RR [7] and OCF_OCF [5]. They take advantage of weight of input VOQs such as queue length or cell waiting time. It is shown that LQF_RR can provide 100% throughput for any admissible traffic [7]. However, since LQF_RR always favours the most occupied VOQ in input scheduling, some queues with low occupancy may appear poor service fairness and even queue starvation. OCF_OCF could achieve stability performance under various traffic as well. However, it needs a complex time stamping mechanism. The third category is *crosspoint buffer state* based algorithm, such as MCBF [8]. MCBF performs scheduling based on crosspoint buffer occupancies instead of information of VOQs. MCBF is unable to keep good performance for non-uniform traffic patterns.

In this paper, we propose a novel high performance scheduling algorithm for CICQ switches, named Most Critical Queue First-Round Robin (MCQF_RR). It takes advantage of the combined information about queue occupancy and service waiting time of input VOQs to make scheduling decisions. The time complexity of MCQF_RR is $O(\log N)$. Simulation results show that MCQF_RR can provide good fairness performance superior to existing efficient scheduling algorithms under extreme non-uniform traffic, while maintain excellent delay performance under uniform and non-uniform traffic patterns.

The remainder of this paper is organized as follows: In Section 2, we introduce our novel scheduling algorithm and make a detailed description. Section 3 presents a performance experimental study. In Section 4, we conclude the paper.

II. THE MOST CRITICAL QUEUE FIRST-ROUND ROBIN SCHEDULING ALGORITHM (MCQF_RR)

As the best performing LQF_RR always favours the VOQ with the highest occupancy, some queues may suffer poor service fairness [7]. In this section, we propose a Most Critical Queue First-Round Robin (MCQF_RR) scheduling algorithm to overcome the disadvantage of LQF_RR. MCQF_RR makes scheduling decisions taking advantage of combined weight information about queue occupancy and service waiting time of input VOQs.

A. *Notation*

The N × N CICQ switch illustrated in Fig. 1 is considered in this paper. Each input port maintains N separated VOQs to store cells destined for N output ports, where VOQij holds cells arrived to input i and destined for output j. There are N2 crosspoint buffers placed at the crosspoints inside the crossbar fabric, where CBij holds cells coming from input i and destined for output j. We give some definitions which will be used throughout the paper as follows:

Time slot: A time slot is the fixed time required to transmit a cell at the line rate.

Eligible VOQij (EVOQij): VOQij is eligible if it is nonempty and its corresponding crosspoint buffer CBij is not full.

Eligible crosspoint buffer (ECBij): CBij is eligible if it is nonempty.

Lij (n) denotes the queue length of VOQij at the beginning of time slot n.

Tij (n) denotes the service waiting time of VOQij which continuously loses service opportunities by time slot n since the last service.

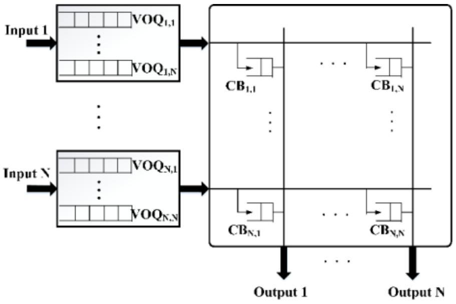


FIGURE I. N×N CICQ SWITCH.

B. *Algorithm Description*

MCQF_RR is based on Most Critical Queue First in input scheduling and Round Robin in output scheduling. The process of MCQF_RR is as follows:

Input Scheduling Phase:

For each input i (0≤j≤N-1), in each time slot n:

Step 1. Starting from the highest priority pointer's location, select the first $EVOQ_{ia}$ corresponding to $min_{\ j}\ L_{ij}\ (n)$ and $EVOQ_{ib}$ corresponding to $max_{\ j}\ L_{ij}\ (n)$.

Step 2. Compare the $T_{ia}\ (n)$ of $EVOQ_{ia}$ with $L_{ib}\ (n) + T_{ib}\ (n)$ of $EVOQ_{ib}$. If $T_{ia}\ (n) > L_{ib}\ (n) + T_{ib}\ (n)$, then serve $EVOQ_{ia}$ and send its HoL cell to crosspoint buffer CB_{ia}. Otherwise, serve $EVOQ_{ib}$ and send its HoL cell to crosspoint buffer CB_{ib}.

Step 3. At the end of the time slot n, update the $T_{ij}\ (n)$ of each VOQ_{ij} and the highest priority pointer's location. If a VOQ_{ij} has obtained the cell service, set $T_{ij}\ (n) = 0$ and move the highest priority pointer to the location $(j + 1)(mod\ N)$. Else, set $T_{ij}\ (n) = T_{ij}\ (n) + 1$.

During Output Scheduling Phase, for each output j, in each time slot n: Starting from the highest priority pointer's location, select the first ECB_{ij} and send its HoL cell to output j. Move the highest priority pointer to the location $(i + 1)(mod\ N)$.

MCQF_RR considers the service unfairness for a 2 × 2 CICQ switch. Traffic flows with $\lambda_{11} = 1$, $\lambda_{12} = 0$ arrive at the switch, where λ_{ij} is the arrival rate of VOQ_{ij}. Assuming that the queue length of each input VOQ is one cell at the beginning. In time slot 1, there is a cell arriving at VOQ_{ij}, but no cells arrive at VOQ_{12}. If Longest Queue First is used, it will serve VOQ_{11} with the longest length and VOQ_{12} will lose service. As there are continuous incoming cells in VOQ_{11}, VOQ_{12} will remain unserved. Therefore, LQF will lead to service unfairness among input VOQs. By contrast, MCQF overcomes the unfairness problem taking advantage of combined information of the VOQ. If the T_{12} (n) of VOQ_{12} with the lowest length increase to large enough being greater than $T_{11}\ (n) + L_{11}\ (n)$ of VOQ_{11} with the longest length which means that VOQ_{12} may appear service unfairness, then MCQF guarantees service to VOQ_{12}. Hence, no queues will be starved of service permanently.

C. *Properties of MCQF_RR*

In this section, we summarize the main characteristics of MCQF_RR: MCQF essentially favours the VOQ_{ij} with the maximum combined weight $L_{ij}\ (n) + T_{ij}\ (n)$, so that each VOQ can receive efficient service without queue starvation. In this way, MCQF_RR can achieve good performance and service fairness simultaneously. In term of time complexity, the selection of $EVOQ_{ia}$ and $EVOQ_{ib}$ in Step 1 both are $O(\log N)$. The comparison in Step 2 has a time complexity of $O(1)$. Therefore, the time complexity of MCQF_RR is $O(\log N)$.

III. PERFORMANCE STUDY

Simulations are carried to evaluate the performance of MCQF_RR and compare with the other algorithms, based on a 16 × 16 CICQ switch with the crosspoint buffer size of 1 cell. The principal item is the cell delay, which is measured as the time (in time slot) taken for a cell from an input to its output port. Average cell delay is the mean value of cells gathered over 1,000,000 time slots. Six scheduling algorithms are selected: RR_RR, RR-LQD, LQF_RR, OCF_OCF, MCBF and the proposed MCQF_RR.

A. *Delay Performance*

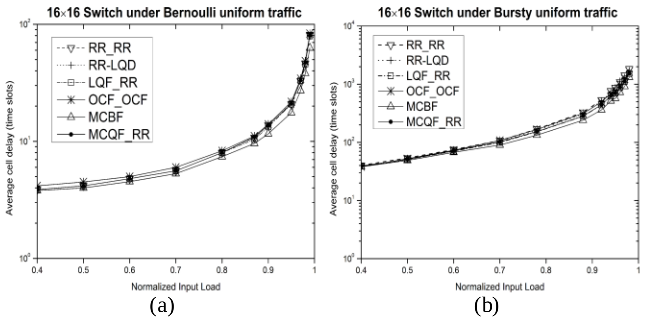
1). Uniform Traffic. Fig. 2(a) shows the average delays of different algorithms under Bernoulli uniform traffic. All the algorithms behave similarly with low average cell delay, which increases gradually as the input load ρ grows. Fig. 2(b) depicts the average delays under Bursty uniform traffic with burst lengths of 32 cells (l = 32). It is shown that, because of the burst property, the average delays achieved by various algorithms increase as the burst length increases and are almost identical.

2). Non-uniform Traffic. For diagonal traffic, the arrival rate of each VOQij is determined as:

$$\lambda_{ij} = \begin{cases} {}^{2\rho}\!/_{3}\,,\ if\ j = i \\ {}^{\rho}\!/_{3}\,,\ if\ j = (i+1)\,\mathrm{mod}\,N \\ 0,\ others \end{cases} \qquad (1)$$

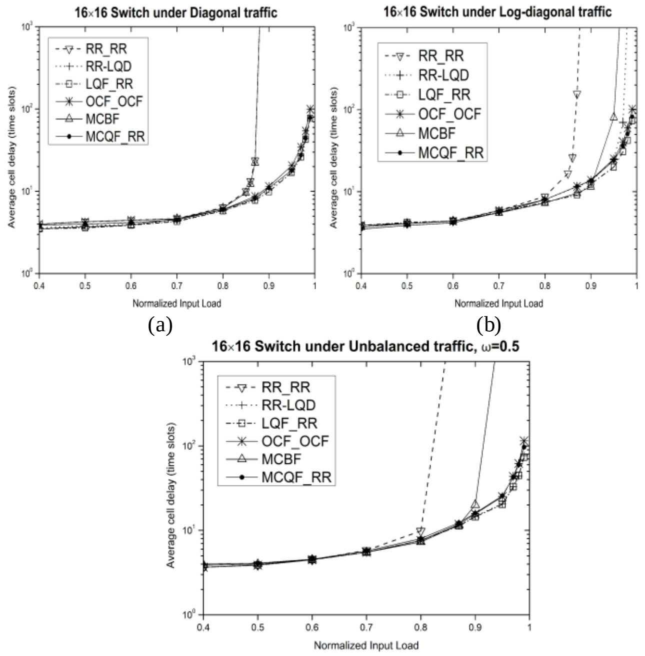
where ρ is the input load of each input and N is the number of ports.

In log-diagonal traffic, each input i has input load for all outputs, but sends twice as much traffic to output j than to *(j+1) mod N*, that is $\lambda_{ij} = 2\lambda_{i(j+1)mod\ N}$. For unbalanced traffic, there is an unbalanced probability ω as the fraction of the input load of each input port. The arrival rate of each VOQ_{ij} is defined as:



(a) (b)

FIGURE II. AVERAGE CELL DELAYS FOR UNIFORM TRAFFIC.



(a) (b)

(c)

FIGURE III. AVERAGE CELL DELAYS FOR NON-UNIFORM TRAFFIC.

$$\lambda_{ij} = \begin{cases} \rho(\omega + \frac{1-\omega}{N}), if\ i = j \\ \rho \cdot \frac{1-\omega}{N}, if\ i \neq j \end{cases} \tag{2}$$

Fig. 3(a) shows the average delay results of different algorithms for diagonal traffic. RR_RR and MCBF show instability for $\rho < 0.9$. The reason is that they don't consider the information about input VOQs in input scheduling. Among other algorithms, OCF_OCF performs the worst for $\rho > 0.95$, while the average delay of MCQF_RR are always close to the best performing LQF_RR. Fig. 3(b) plots the average delays under log-diagonal traffic. MCQF_RR delivers low average delay close to LQF_RR. OCF_OCF has higher delay than MCQF_RR especially at $\rho > 0.95$. RR_RR and MCBF show instability under ρ around 0.9, and RR-LQD saturates for $\rho > 0.95$. Fig. 3(c) shows the delay results under unbalanced traffic (ω=0.5). RR_RR and MCBF saturate at ρ around 0.9. Among the algorithms which maintain stability, OCF_OCF has the highest average delay at high load. MCQF_RR achieves low delay comparable to LQF_RR, ensures the queues with high load efficiently served.

B. *Fairness Performance*

In this section, the service fairness of MCQF_RR will be examined. According to the delay results, LQF_RR provides best delay performance for any traffic patterns, therefore we mainly compare the service fairness of MCQF_RR with LQF_RR. We consider the input scheduling process in service fairness comparison. The fairness index defined in [10] is used to to evaluate the fairness performance, let

$$FI(D_1, D_2, \cdots, D_N) = \frac{\left(\sum_{j=1}^{N} D_j\right)^2}{N\sum_{j=1}^{N} D_j{}^2} \tag{3}$$

where D_j is the average cell delay of VOQ_{ij}, $j \in [1, 2, \cdot\cdot\cdot, N]$, for input i. FI (D_1, D_2, $\cdot\cdot\cdot$, D_N) has a range of [0, 1]. The smaller is FI, the worse is the service fairness. Then, we define average fairness index (AFI), which is calculated as the fairness index averaged over all the considered loads of input traffic.

TABLE I. FI AND AFI RESULTS UNDER LOG-DIAGONAL TRAFFIC.

Algorithm	**Fairness Index (FI) for Input Load ρ**						**AFI**
	90%	**95%**	**97%**	**98%**	**99%**	**100%**	
LQF_RR	0.8708	0.7227	0.6621	0.6252	0.5692	0.3740	0.6373
RR-LQD	0.9328	0.7229	0.4053	0.1178	0.1135	0.1131	0.4009
MCQF_RR	0.9498	0.9404	0.9097	0.8381	0.6568	0.4433	0.7897

The fairness indexes of LQF_RR, RR-LQD and MCQF_RR under log-diagonal traffic for high input load ρ (90%–100%) are presented in Table 1. RR-LQD exhibits the worst, because of its unstable delay performance. Our proposed MCQF_RR achieves fairness indexes higher than LQF_RR for all loads. Furthermore, the AFI for MCQF_RR is higher than LQF_RR by almost 20%. Therefore, MCQF_RR outperforms LQF_RR in terms of fairness performance without sacrificing the delay performance, because of the scheduling scheme based on the combined information about queue length and service waiting time of input VOQs.

IV. CONCLUSION

This paper introduces a Most Critical Queue First-Round Robin (MCQF_RR) scheduling algorithm for crosspoint buffered crossbar switches, which is based on the combined information about queue length and service waiting time of input VOQs in input scheduling. The time complexity of MCQF_RR is $O(\log N)$. Simulation results show that the presented MCQF_RR provides significantly excellent delay performance comparable to high performance LQF_RR scheduling algorithm under any admissible input traffic patterns, and more importance is that MCQF_RR achieves good service fairness performance superior to LQF_RR under extreme non-uniform traffic.

REFERENCES

[1] Keshav, S. & Sharma, R., Issues and trends in router design, *IEEE Communications Magazine*, 36(9), pp. 144–151, 1998.

[2] McKeown, N., Scheduling algorithms for input-queued cell switches, Ph.D. thesis, University of California at Berkeley, 1995.

[3] McKeown, N., Mekkittikul, A. & Anantharam, V., Achieving 100% throughput in an input queued switch, *IEEE Transactions on*

Communications, 47(8), pp. 1260–1267, 1999.

[4] McKeown, N., The iSLIP Scheduling Algorithm for Input-Queued Switches, *IEEE/ACM Transactions on Networking*, 7(2), pp. 188–201, 1999.

[5] Nabeshima, M., Performance Evaluation of Combined Input- and Crosspoint-Queued Switch, *IEICE Transaction on Communications*, E83-B(3), pp. 737–741, 2000.

[6] Rojas-Cessa, R., Oki, E. & Jing, Z., CIXB-1: Combined Input One-Cell-Crosspoint Buffered Switch, *Proc. IEEE Workshop High Performance Switching and Routing (HPSR)*, pp. 324–329, 2001.

[7] Javadi, T., Magill, R. & Hrabik, T., A High-Throughput Scheduling Algorithm for a Buffered Crossbar Switch Fabric, *Proc. IEEE International Conference on Communications*, pp. 1581–1591, 2001.

[8] Mhamdi, L. & Hamdi, M., MCBF: A High-Performance Scheduling Algorithm for Buffered Crossbar Switches, *IEEE Communication Letters*, 7(9), pp. 451–453, 2003.

[9] Yun, Z., Peng, L. & Zhao, W., RR-LQD: A novel scheduling algorithm for CICQ switching fabrics, *Proc. 15th Asia-Pacific Conference on Communications (APCC)*, pp. 846–849, 2009.

[10] Jain, R., Durresiand A. & Babic, G., Throughput fairness index: An explanation, ATM Forum Document: ATM-Forum/99-0045, 1999.