Parallel matching algorithm for CIOQ switches with multiple traffic classes

T.-H. Lee and Y.-C. Kuo

Abstract: It has recently been shown that a combined input and output queued (CIOQ) switch with a speed-up factor of 2 can exactly emulate an output-queued (OQ) switch. The main benefit of using a CIOQ switch is to reduce memory bandwidth requirement while providing quality of service (QoS) guarantee. The key component for exact emulation is a matching algorithm for bipartite graphs. For example, a CIOQ switch with a speed-up factor of 2, which adopts the least cushion first/most urgent first (LCF/MUF) matching algorithm, can exactly emulate an OQ switch with any arbitrary service discipline. However, the complexities of cushion calculation and cell reordering required at the output ports make the algorithm very difficult to be realised in a high-speed switch. The authors propose an approximate LCF/MUF algorithm and evaluate its performance for the weighted round-robin service discipline. For ease of implementation, the proposed algorithm calculates approximate cushions and does not perform reordering at the output ports. The trade-off is that it loses the property of exact emulation. It was found, via computer simulations, that the performance of a CIOQ switch with the proposed single-iteration matching algorithm is close to that of an OQ switch under uniform, nonuniform, and correlated input traffic models for offered load up to 0.9. In addition, the cell departure order can be maintained under the single-iteration algorithm.

1 Introduction

In the past decade, many service disciplines had been proposed to provide quality of service (QoS) guarantees in an integrated services network [1–4]. Most of these algorithms were designed to be used at the output ports of an output-queued (OQ) switch. The main problem of OQ switches is that the switching fabric of an $N \times N$ switch must run N times as fast as its line rate in the worst case, where N denotes the number of input/output ports. OQ switches have a serious scaling problem because the advancement in memory bandwidth is much slower than the advancement in transmission speed. A parallel packet switch (PPS) architecture which is constructed with multiple identical lower speed OQ switches operating independently and in parallel was proposed to make a large-capacity switch with extremely high line-rates feasible [5–7]. Clearly, the lower speed OQ switch is still a factor which can set a limit on system capacity.

To alleviate memory bandwidth requirement, input queueing is widely considered in building a large capacity switch. However, input-queued (IQ) switches suffer from head-of-line (HOL) blocking which limits the maximum throughput to about 0.586 under the uniform traffic assumption [8]. It can be improved to approach 100% throughput if the cells are delivered from input ports to output ports based on maximum matching [9]. Unfortunately, the high computational complexity of currently known algorithms prohibits maximum matching from being used in a high-speed switch. To reduce the complexity, several maximal matching algorithms (e.g. PIM [10], LPF [11], *i*SLIP [12] and FIRM [13]) have been proposed. However, none of these algorithms can provide a QoS guarantee. Another approach to improve the performance of an IQ switch is to speed-up the switching fabric. Because of speed-up, an output port may receive cells faster than it can transmit them. As a result, buffering at output ports is necessary and the switch becomes a combined input- and output-queued (CIOQ) switch.

To obtain good performance in a CIOQ switch, one has to schedule the usage of the switching fabric wisely. Several algorithms had been investigated to achieve this goal. For example, an algorithm named the least output occupancy first algorithm (LOOFA) [14] was proposed for a CIOQ switch with a speed-up factor of 2 to achieve 100% throughput. The algorithm basically performs maximal matching and thus can be realised in a high-speed switch. However, achieving 100% throughput is not sufficient for QoS guarantee. It has been proved [15, 16], that a CIOQ switch with a speed-up factor of 2 can exactly emulate an OQ switch which employs a wide variety of scheduling algorithm if stable matching is adopted. Unfortunately, the complexity of stable matching is $O(N^2)$, and thus is impractical for a large system. In [17], the authors proposed a parallel maximal matching algorithm (of complexity $O(N)$), called the least cushion first/most urgent first (LCF/MUF) algorithm, and showed that a CIOQ switch with a speed-up factor of 2 can exactly emulate an OQ switch which adopts any service discipline at the output ports. But the complexity of cushion calculation and cell reordering required at the output ports make the algorithm very difficult to realised. In this paper, we propose an approximate LCF/MUF algorithm and evaluate its performance for the weighted round-robin (WRR) service discipline. The proposal simplifies cushion calculation and

IEE Proceedings online no. 20030600

doi:10.1049/ip-com:20030600

Paper first received 11th December 2002 and in revised form 14th April 2003

The authors are with the Institute of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan 300, Republic of China

does not perform cell reordering at output ports. The trade-off is that it loses the property of exact emulation. However, we found via computer simulations that the performance of a CIOQ switch using the proposed single-iteration approximate LCF/MUF algorithm is close to that of an OQ switch under uniform, nonuniform, and correlated input traffic models. In addition, the cell departure order is maintained for the WRR service discipline.

2 LCF/MUF matching algorithm

In this Section, we briefly review the LCF/MUF algorithm proposed in [17]. Before describing the algorithm, we need the following definitions. Let $x_{i,j}$ denote a cell buffered at input port i destined to output port j.

Definition 1: The cushion of cell $x_{i,j}$ buffered at input port i, denoted by $C(x_{i,j})$, equals the number of cells currently residing in output port j which will depart the emulated OQ switch earlier than cell $x_{i,j}$.

Definition 2: The cushion between input port i and output port j, denoted by $C(i,j)$, is the minimum of $C(x_{i,j})$ of all cells buffered at input port i destined to same output port j. If there is no cell destined to output port j, then $C(i,j)$ is set to ∞.

Definition 3: The scheduling matrix of a $N \times N$ switch is a $N \times N$ square matrix whose (i,j)th entry equals $C(i,j)$.

The matching procedures made by the LCF/MUF algorithm are described below:

Step 1: Select the (i,j)th entry of the scheduling matrix which satisfies $C(i,j) = \min_{k,l}\{C(k,l)\}$ (least cushion first). If the selected entry is ∞, then stop. If there is more than one entry with the least cushion residing in different columns, then choose the most urgent cell $x_{i,j}$ among those input ports which correspond to the selected entries (most urgent first). (The most urgent cell $x_{i,j}$ means the cell which will depart output port j in the emulated OQ switch earlier than any other cell currently residing in some input port. It is clear that urgency of a cell depends on the service discipline adopted at output ports.)

Step 2: Eliminate the ith row and the jth column (i.e. match output port j to input port i) of the scheduling matrix. If the reduced matrix becomes null, then stop. Otherwise, use the reduced matrix and go to step 1.

It was proved in [17] that a CIOQ switch with a speed-up factor of 2 that adopts the LCF/MUF matching algorithm can exactly emulate an OQ switch with any arbitrary service discipline. For convenience, the emulated OQ switch is referred to as the shadow OQ switch. As was pointed out before, it is very difficult to obtain the cushions of cells at all input ports for a complicated service discipline such as WFQ [18], PGPS [3] or WF^{2}Q [4]. As such, for high-speed switches, it is better to select a service scheduling algorithm which can provide a QoS guarantee and is simple for cushion calculation. The WRR scheme satisfies these requirements and thus is studied in this paper. We assume that the output port j ($1 \le j \le N$) maintains K queues, denoted by $OQ_{j,k}$ for $1 \le k \le K$, with integral weights w_1, w_2,..., and w_K. The WRR service algorithm adopted at the output ports for the shadow OQ switch is described below.

Step 1: Let $k = 1$.

Step 2: If $w_k > 0$, then $w_k = w_k - 1$. The HOL cell of output queue $OQ_{j,k}$ is served if queue $OQ_{j,k}$ is non-empty. If $w_k = 0$ for all k, $1 \le k \le K$, then reset all the weights and go to step 1.

Step 3: $k = k + 1$ modulo K. Go to step 2.

Figure 1 shows an example of the WRR service discipline with $K = 3$, $w_1 = 4$, $w_2 = 3$, and $w_3 = 1$. Figures 1*a* and 1*b* show one cycle of the WRR scheme. In Fig. 1*a*, every queue is non-empty when visited. In Fig. 1*b* queue 2 is empty in its second and third visits and thus the cycle is shorter than a regular one.

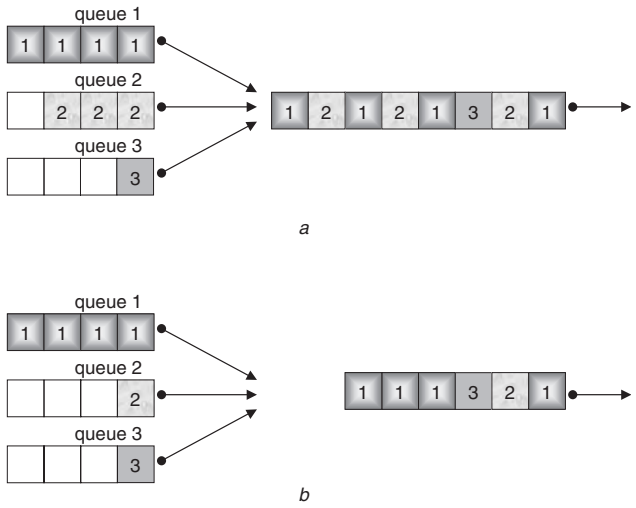

Fig. 1 *Examples of the WRR service algorithm with $K = 3$, $w_1 = 4$, $w_2 = 3$, and $w_3 = 1$*
a Regular cycle
b Shortened cycle

3 Approximate LCF/MUF algorithm

The conceptual model of the considered CIOQ switch is illustrated in Fig. 2. Every input port maintains per output port, per priority queues. In other words, input port i ($1 \le i \le N$) maintains a separate set of FIFO queues for output port j ($1 \le j \le N$). As a consequence, there are N sets of queues at each input port and each set consists of K priority FIFO queues. The set of queues maintained at input port i for output port j are named as $VOQ_{i,j,k}$ for $1 \le k \le K$. A new arrival cell at input port is placed at the tail of the appropriate queue depending on its destination and priority. At output port j, there are K queues, denoted by $OQ_{j,k}$, which are served based on the WRR scheduling algorithm.

3.1 Single-iteration matching

The proposed approximate LCF/MUF matching algorithm in the CIOQ switch is similar to the PIM algorithm [10] and has three phases in each iteration. In phase 1, input ports send requests to output ports. These requests contain the arrival times of the HOL cells. In phase 2, every output port computes the cushions for all cells contained in the requests, selects the least cushion cell, and then sends a grant message back to the corresponding input port. Finally, in phase 3, every input port which receives grant messages picks one, based on the cushion, and sends an accept message to the selected output port. To simplify the calculation of cushions in phase 2, cells belonging to the same queue at an output port are not reordered, i.e. the queue is the same as a FIFO queue and cells are served in the order in which they arrive at the output port. It should be noted that the calculated cushion is just an approximation because, for exact cushion

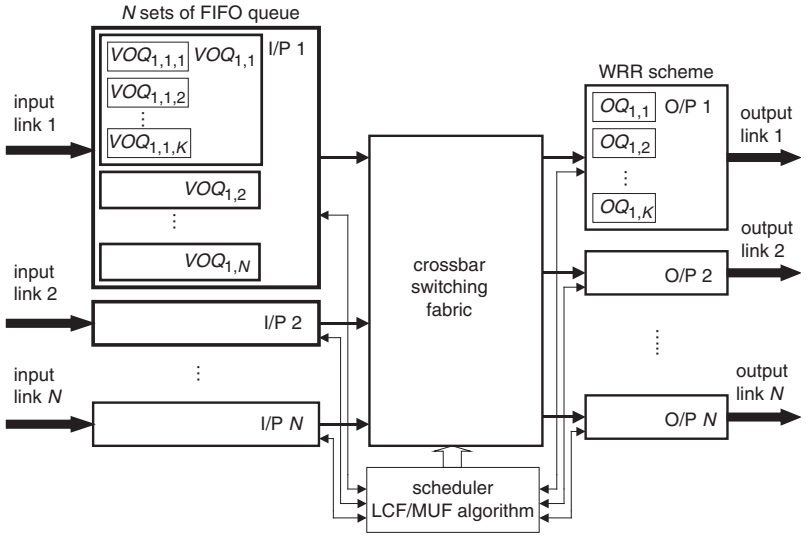

Fig. 2 *CIOQ switch using the proposed approximate LCF/MUF matching algorithm and serving cells at output ports with the WRR scheme*

calculation, one has to know the departure times of the cells for the shadow OQ switch. Details of the matching algorithm are described below.

Phase 1: Input ports send requests to output ports. In phase 1, every input port i sends a request to each output port j. The request contains the arrival times of the HOL cells of all the K queues $VOQ_{i,j,k}$. The arrival time is set to infinity if a queue is empty.

Phase 2: Output ports send grants to input ports. In phase 2, every output port calculates the approximate cushions for the cells of each received request which was sent in phase 1. The cushion of a cell is the number of time slots it has to wait before its service at the output port, assuming that it is delivered to the output port in the current time slot and that there is no other arrival in the future. After the cushions of all cells are calculated, the output port selects the request with the least cushion and sends a grant message to the input port which sent the request. If there are ties, then the cell with the earliest arrival time wins the contest. The grant message contains the calculated cushion of the request and the identification of the selected queue.

Phase 3: Input ports send accepts to output ports. If an input port receives exactly one grant message from some output port, then it sends an accept message to the output port in phase 3. If an input port receives multiple grants, then it selects the grant with the least cushion. If there are ties, then the cell with the earliest arrival time is selected.

Figure 3 shows an example of cushion calculation when the WRR service scheme is adopted at output ports. In this example, there are three queues at an output port with weights 4, 3 and 1. Note that the cushion of a cell depends on the state (i.e. the position in a service cycle) of the output port. In Fig. 3*a*, the cushion of a request belonging to the second priority queue, denoted as $C(w_2)$, is equal to 4 as the current service token is at the first position of the service cycle. In Fig. 3*b*, $C(w_2) = 2$ as the current service token is at the fifth position.

Based on the outcome of the above matching procedure, cells are delivered from input ports to output ports. Remember that the switching fabric is speeded up by a factor of 2 and thus every input port can send at most 2 cells to output ports in a slot.

For ease of implementation, our scheme does not allow cell reordering at output ports. In fact, for the single-iteration scheme, it is not necessary to reorder cells at output ports. The reason is explained as follows. Consider two cells x and y of the same priority at different input ports. Assume that both x and y are HOL cells and destined for the same output port. If cell x arrived earlier than cell y, then the output port will select cell x in phase 2. Therefore, cell out-of-order will not happen at any output port.

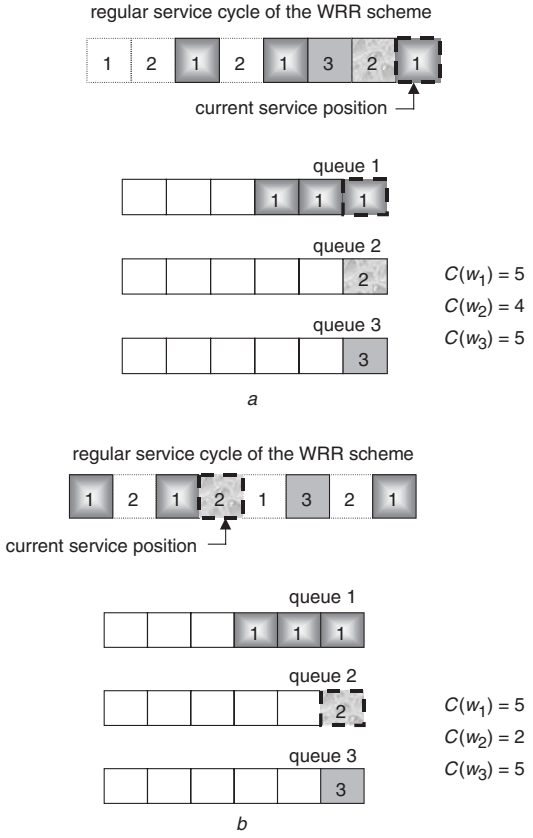

Fig. 3 *Examples of cushion calculation for the WRR scheme*
a Current token at first position in service cycle
b Current token at fifth position in service cycle

3.2 Multiple-iteration matching

To improve the throughput performance of the switch fabric, the above matching procedure can be repeated for

multiple iterations. After each iteration, an input port stops sending requests once it has sent an accept message to a certain output port. Similarly, an output port stops calculating the cushions and sending a grant message once it has received an accept message. The subsequent iterations will try to match the rest of the input and output ports that remain unmatched in previous iterations. Obviously, the throughput increases as the number of iteration increases. However, multiple iterations may result in out-of-order delivery of cells and hurt the goal of emulating an OQ switch if cell reordering is not allowed at output ports. Our simulation results show that multiple-iteration matching does not yield a better performance than single-iteration matching.

4 Input traffic models

To quantitatively evaluate the performance of our considered CIOQ switch, we describe practical input traffic models that will be sent simultaneously to both the shadow OQ switch and the CIOQ switch in each simulation experiment. We introduce three input traffic models: uniform, nonuniform and correlated models.

4.1 Uniform and nonuniform input traffic

The uniform input traffic model is used to characterise the interactive behaviour of data arrivals in computer networks [8]; there is no time correlation between arrivals. Cells arrival on the N input links are governed by i.i.d. Bernoulli processes.

For nonuniform traffic, we use the same model as in [19]. We define nonuniform traffic by using an unbalance weight probability T_w. Let us consider the input port i and the output port j. Given ρ_i, the offered input load for input port i, the traffic load from input port i to output port j, denoted by $\rho_{i,j}$, is given by

$$\rho_{i,j} = \begin{cases} \rho_i\left(T_w + \frac{1-T_w}{N}\right) & \text{if } i = j \\ \rho_i \frac{1-T_w}{N} & \text{otherwise} \end{cases} \quad (1)$$

When $T_w = 0$, the offered traffic is uniform. On the other hand, when $T_w = 1$, the traffic is completely unbalanced in the sense that all the traffic of input port i is destined for output port i.

4.2 Correlated input traffic

The correlated traffic model we are studying is characterised by a two-state Markov process alternating between active and idle states with probabilities p and q, respectively [20, 21] (see Fig. 4). This model is often used to describe IP packets fragmented into ATM cells. Based on this model, the traffic source will generate a cell every slot when it is in the active state. Note that there is at least one cell in a burst. We define the random variable B as the number of time slots that the active period (burst) lasts. The probability which the active period lasts for a duration of m time slots (i.e. consists of m cells in a burst) is

$$\Pr[B = m] = p(1-p)^{m-1}, \quad m \geq 1 \quad (2)$$

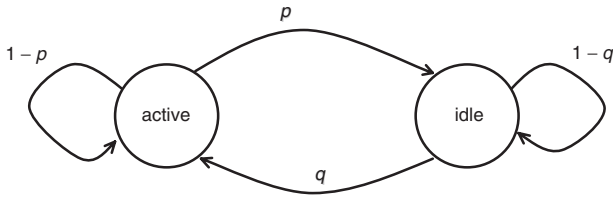

Fig. 4 *Two-state Markov chain process*

The mean burst length is given by

$$E[B] = \sum_{m=1}^{\infty} m \cdot \Pr[B = m] = 1/p \quad (3)$$

Similarly, we define the random variable I as the number of time slots for which the idle period lasts. The probability that an idle period lasts for n time slots is

$$\Pr[I = n] = q(1-q)^n, \quad n \geq 0 \quad (4)$$

and the mean idle period duration is given by

$$E[I] = \sum_{n=1}^{\infty} n \cdot \Pr[I = n] = (1-q)/q \quad (5)$$

Given p and q, the offered traffic load ρ can be found by

$$\rho = \frac{E[B]}{E[B] + E[I]} = \frac{q}{p + q - pq} \quad (6)$$

We assume there is no correlation between different bursts and that the destination of each burst is uniformly distributed among the output ports.

5 Numerical results

To evaluate the performance of the CIOQ switch using the proposed matching algorithm, we measure the latency of the cells which are simultaneously fed to both the shadow OQ switch and the CIOQ switch under uniform, nonuniform and correlated traffic models. We define $d(x) = |d_{OQ}(x) - d_{CIOQ}(x)|$ as the deviation index of cell x. Here $d_{OQ}(x)$ and $d_{CIOQ}(x)$ denote, respectively, the departure times of cell x for the shadow OQ switch and the CIOQ switch. Given a fixed d, we measure the percentage of cells that has a deviation index smaller than or equal to d. This percentage is denoted by P_d. For example, $P_{d=0}$ equals 100% means that the CIOQ switch exactly emulates the shadow OQ switch.

Our simulation experiments are divided into five parts. In the first three parts, we measure the values of P_d under uniform, nonuniform and correlated traffic models, respectively. We found that the performance of a CIOQ switch using the proposed matching algorithm is close to that of an OQ switch. In the fourth part, we study the effect of multiple-iteration matching. Results show that single-iteration matching is good enough and thus cell reordering is not needed. In the last part, we change the speed-up factor and show that a CIOQ switch with a speed-up factor of 2 yields much better performance than one without speed-up. A speed-up factor larger than 2 is not necessary. Our simulation results also show the performance of the CIOQ switch, with a speed-up of 2, using the FIRM algorithm for comparison. It will be seen that the LCF/MUF algorithm performs better under these three traffic models. We performed simulations for the CIOQ switch using the *i*SLIP algorithm. The performance of *i*SLIP is worse than that of FIRM and is not shown in this paper.

Simulations are performed for $N = 8$, 16, and 32 for 3 priority levels with weights $w_1 = 4$, $w_2 = 3$, and $w_3 = 1$, under uniform, nonuniform and correlated traffic models. For correlated traffic, the mean burst length $\ell = 8$, 16, 32 and 64 are considered. In each experiment, the statistics of about one million cells over all queues at input ports are collected.

5.1 Performance under uniform traffic

Simulation results for uniform input traffic are shown in Figs. 5 and 6. For switch size $N = 16$ and the switch adopts

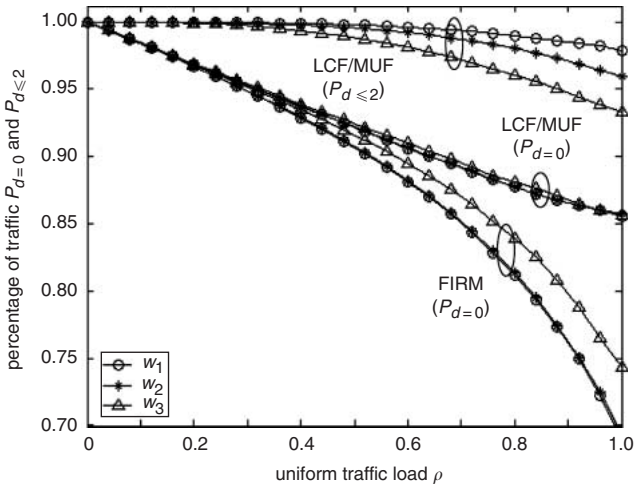

Fig. 5 *Performance of 16 × 16 CIOQ switch under uniform traffic*
Curves show the values of $P_{d=0}$ and $P_{d\leq 2}$ for all three priority traffics ($w_1 = 4$, $w_2 = 3$, and $w_3 = 1$)

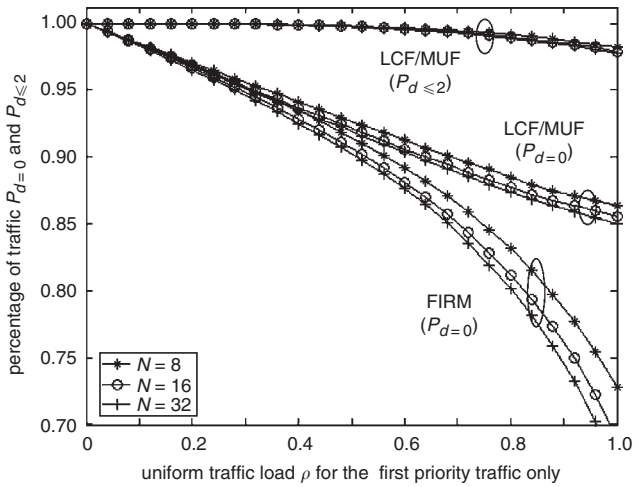

Fig. 6 *Performance of the first priority traffic as a function of offered load for various switch sizes*

the proposed LCF/MUF algorithm, Fig. 5 shows that the values of $P_{d=0}$ are larger than 0.866 for all three priority traffics under an offered load up to 0.9 and the values of $P_{d\leq 2}$ are close to 100%. The curves also show that the performances of all three priority traffics are similar. In Fig. 6, the curves show the performances of the CIOQ switch for various switch sizes. It can be seen that the performance degrades as the number of ports increases. But it degrades slowly when N is larger than 16. Based on these results, we conclude that the performance of a CIOQ switch with the proposed matching algorithm is close to that of an OQ switch under uniform traffic.

Figures 5 and 6 also show that the performances of the FIRM algorithm are worse than those of the LCF/MUF algorithm. The difference is significant under heavy load conditions. To avoid drawing too many lines, which would make these Figures less comprehensible, the curves of $P_{d\leq 2}$ for the FIRM algorithm are not shown. Similarly, the curves for the second and third priority traffic cells are not shown in Fig. 6. Our results show that $P_{d\leq 2}$ for the FIRM algorithm is smaller than that for the LCF/MUF algorithm. The performances of the second and third priority traffic cells are similar to those of the first priority traffic cells in Fig. 6. It should be noted that the cushion calculation helps the LCF/MUF algorithm to achieve a better performance than the FIRM algorithm. Moreover, the FIRM algorithm may result in an out-of-sequence delivery that also hurts its performance if reordering is not allowed at the output ports.

5.2 Performance under non-uniform traffic

For nonuniform input traffic, simulation results are shown in Figs. 7 and 8 for switch size $N = 16$ which adopts the proposed LCF/MUF algorithm and the FIRM algorithm, respectively. When $T_w = 0$, the offered traffic is uniform and thus the values of $P_{d=0}$ are the same as those shown in Fig. 5. The value of $P_{d=0}$ increases and is equal to 100% as T_w increases to one. The reason is that, when $T_w = 1$, the output port j is always matched to the input port j for all j.

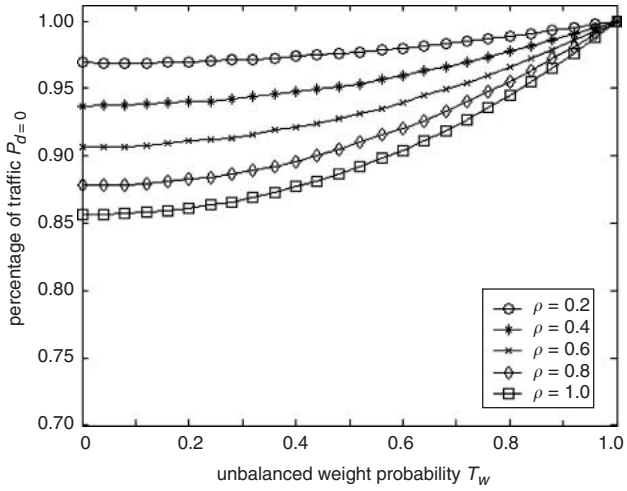

Fig. 7 *Performance of 16 × 16 CIOQ switch adopting the proposed LCF/MUF algorithm*
Curves show values of $P_{d=0}$ of first priority traffic for different T_w value nonuniform traffic for various traffic loads

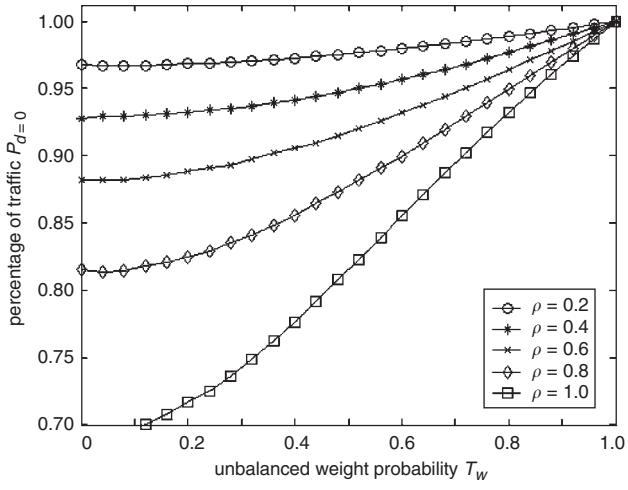

Fig. 8 *Performance of 16 × 16 CIOQ switch adopting the FIRM algorithm*
Curves show values of $P_{d=0}$ of first priority traffic for different T_w values nonuniform traffic for various traffic loads

5.3 Performance under correlated traffic

For the correlated traffic model, results depend on the mean burst length ℓ. In Fig. 9, we show the results for $\ell = 16$ cells and $N = 16$. When the switch adopts the proposed algorithm, the values of $P_{d=0}$ are larger than 0.875 for all three priority traffics under an offered load up to 0.9 and

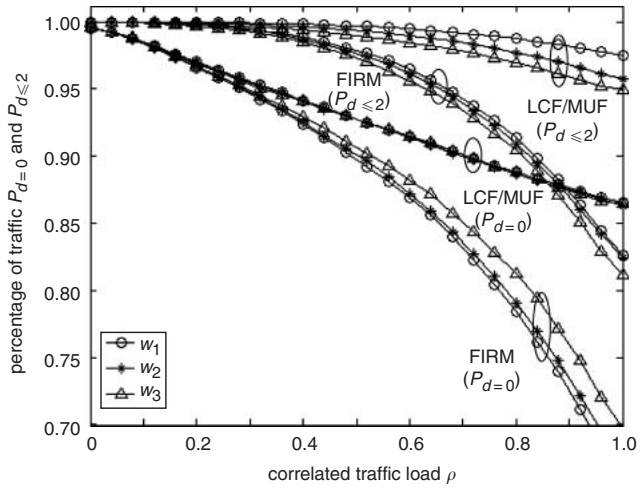

Fig. 9 *Performance of a 16 × 16 CIOQ switch with $\ell = 16$*
Curves are shown for priority traffic ($w_1 = 4$, $w_2 = 3$, and $w_3 = 1$)

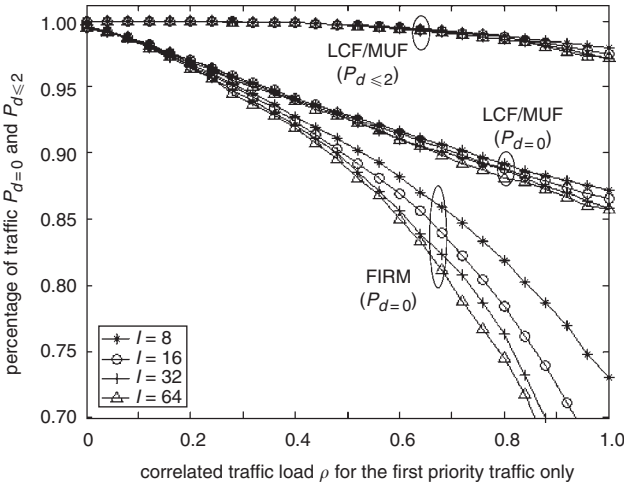

Fig. 11 *Performance of 16 × 16 CIOQ switch under correlated traffic*
Curves show the performances of the first priority traffic as a function of offered load for mean burst lengths $\ell = 8$, 16, 32 and 64 cells

the values of $P_{d\leq 2}$ are close to 100%. The performances of all the three priority traffics are similar under both uniform and correlated traffic models. In Fig. 10, the curves show the performance of the CIOQ switch for various switch sizes. It can be seen that the performance degrades as the number of ports increases. Again, it degrades slowly when N is larger than 16. In Fig. 11 we performed similar simulations for others values of ℓ. It can be seen that the performance degrades as ℓ increases. But the degradation becomes slow once ℓ is larger than 16. Again, to avoid drawing too many lines, the curves of $P_{d\leq 2}$ for the FIRM algorithm are not shown. The curves for the second and third priority traffic cells are not shown in Figs. 10 and 11. Based on these results, we conclude that the performance of a CIOQ switch with the proposed matching algorithm is close to that of an OQ switch under correlated traffic.

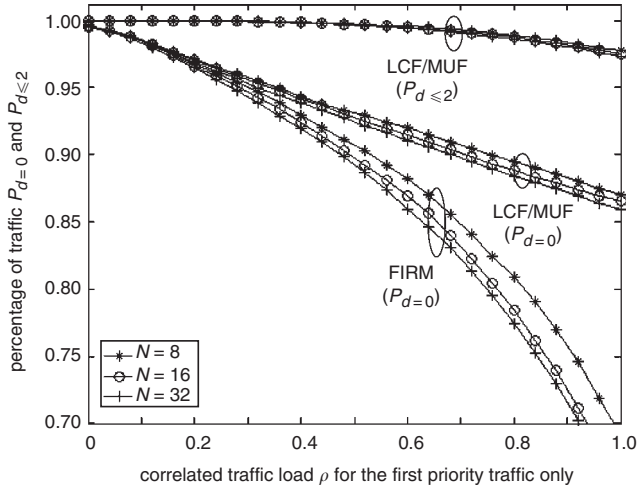

Fig. 10 *Performance of the first priority traffic for various switch sizes $\ell = 16$ cells*

5.4 Effects of multiple-iteration matching

We now study the effects of multiple-iteration matching. In our experiments, the algorithm was performed with three and five iterations. The results are shown in Fig. 12. As mentioned before, multiple-iteration matching does not improve the performance in terms of emulation of a shadow

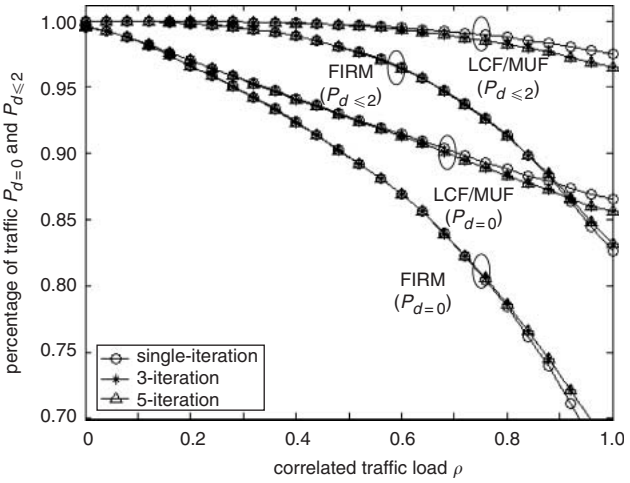

Fig. 12 *Performance of a 16 × 16 CIOQ switch under correlated traffic with mean burst length $\ell = 16$*
Curves show performance of first priority traffic under single-iteration, 3-iteration, and 5-iteration matching

OQ switch. Therefore, we conclude that single-iteration matching is adequate for the proposed approximate LCF/MUF algorithm.

5.5 Effects of speed-up factors

We now study the effects of the speed-up factors. In Fig. 13, we show the results for $\ell = 16$ cells and $N = 16$ for speed-up factors $S = 1$, 2, and 3. It can be seen that, for $S = 1$ which means no speed-up, the system performance degrades seriously as the traffic load increases. The reason is that the approximate LCF/MUF is not a maximum matching algorithm. The performance can be improved if the matching is performed for multiple iterations. However, the improvement is not obvious. As illustrated in Fig. 13, the results for $S = 3$ are almost the same as those for $S = 2$. We also performed simulations for $S = 4$ and 5 and the results are similar. So, a speed-up factor of 2 is important to achieve satisfactory performance. A speed-up factor greater than 2 is not necessary.

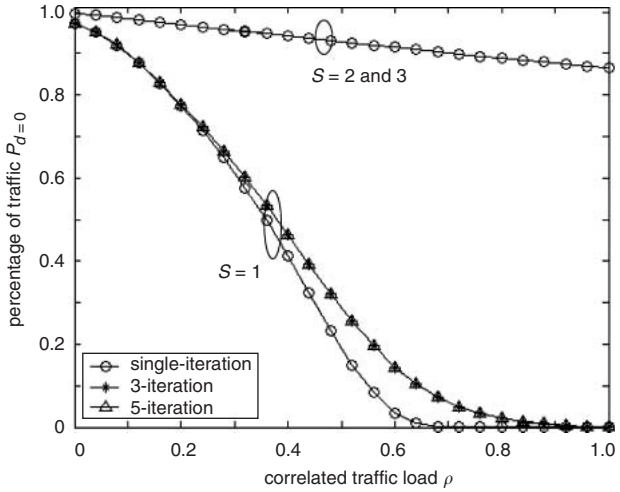

Fig. 13 *Performance of 16 × 16 CIOQ switch under correlated traffic with mean burst length $\ell = 16$*
Curves show performance first priority traffic under single-iteration, 3-iteration, and 5-iteration matching for various speed-up factors ($S = 1, 2, 3$)

6 Conclusions

We have proposed an approximate LCF/MUF matching algorithm. Our proposal makes the LCF/MUF algorithm much more feasible for implementation because it needs only single-iteration matching and the calculation of cushions of cells is largely simplified. Numerical results show that the performance of the proposed approximate matching is very close to that of the LCF/MUF algorithm under uniform, nonuniform, and correlated traffic models for offered load up to 0.9. Our simulation results show that the performance of the proposed approximate LCF/MUF algorithm is significantly better than that of a simple matching algorithm such as FIRM or *i*SLIP. The price paid for the LCF/MUF algorithm is higher hardware cost because it requires cushion calculations.

Since the memory bandwidth sets a limit on switch capacity, the CIOQ switch is likely to be more suitable than the OQ switch in building a large-capacity switch. Our proposal reduces the implementation complexity of a CIOQ switch and thus should be useful for building large-capacity switches with guaranteed QoS.

7 Acknowledgment

This work was supported by the Ministry of Education, Taiwan, Republic of China, under contract 90X011C.

8 References

1 Zhang, H.: 'Service disciplines for guaranteed performance service in packet-switching networks', *Proc. IEEE*, 1995, **83**, (10), pp. 1374–1399
2 Ferrari, D., and Verma, D.C.: 'A scheme for real-time channel establishment in wide-area networks', *IEEE J. Sel. Areas Commun.*, 1990, **8**, (3), pp. 368–379
3 Parekh, A.K., and Gallager, R.G.: 'A generalized process sharing approach to flow control in integrated services networks: the single node case', *IEEE ACM Trans. Netw.*, 1993, **1**, (3), pp. 344–357
4 Bennett, J.C.R., and Zhang, H.: 'WF^{2}Q: worst-case fair weighted fair queueing'. Proc. IEEE INFOCOM'96, San Francisco, CA, USA, March 1996, pp. 120–128
5 Iyer, S., Awadallah, A., and McKeown, N.: 'Analysis of a packet switch with memories running slower than the line-rate'. Proc. IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000, pp. 529–537
6 Iyer, S., and McKeown, N.: 'Making parallel switches practical'. Proc. IEEE INFOCOM 2001, Anchorage, AK, USA, April 2001, pp. 1680–1687
7 Khotimsky, D., and Krishnan, S.: 'Stability analysis of a parallel packet switch with bufferless input demultiplexors'. Proc. IEEE ICC 2001, Helsinki, Finland, June 2001, pp. 100–111
8 Karol, M., Hluchyj, M., and Morgan, S.: 'Input verse output queueing on a space division switch', *IEEE Trans. Commun.*, 1987, **35**, pp. 1347–1763
9 McKeown, N., Anantharam, V., and Walrand, J.: 'Achieving 100% throughput in an input-queued switch'. Proc. IEEE INFOCOM'96, San Francisco, CA, USA, March 1996, pp. 296–302
10 Anderson, T.E., Owicki, S.S., Saxe, J.B., and Thacker, C.P.: 'High speed switch scheduling for local area networks', *ACM Trans. Comput. Syst.*, 1993, **11**, (4), pp. 319–352
11 Mekkittikul, A., and McKeown, N.: 'A practical scheduling algorithm to achieve 100% throughput in input-queued switches'. Proc. IEEE INFOCOM'98, San Francisco, CA, USA, 1998, pp. 792–799
12 McKeown, N.: 'The *i*SLIP scheduling algorithms for input-queued switches', *IEEE/ACM Trans. Netw.*, 1999, **7**, (2), pp. 188–201
13 Serpanos, D.N., and Antoniadis, P.I.: 'FIRM: a class of distributed scheduling algorithms for high-speed ATM switches with multiple input queues'. Proc. IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000, pp. 548–555
14 Krishan, P., Patel, N.S., Charny, A., and Simcoe, R.: 'On the speed-up required for work-conserving crossbar switches'. Proc. IWQoS, Napa, CA, USA, 1998, pp. 225–234
15 Stoica, I., and Zhang, H.: 'Exact emulation of an output queueing switch by a combined input output queueing switch'. Proc. IWQoS, Napa, CA, USA, 1998, pp. 218–224
16 Chuang, S.T., Goel, A., McKeown, N., and Prabhakar, B.: 'Matching output qeueueing with a combined input/output-queued switch', *IEEE J. Sel. Areas Commun.*, 1999, **17**, (6), pp. 1030–1039
17 Lee, T.-H., Kuo, Y.W., and Huang, J.-C.: 'Quality of service guarantee in a combined input output queued switch', *IEICE Trans. Commun.*, 2000, **E83-B**, (2), pp. 190–195
18 Demers, A., Keshav, S., and Shenker, S.: 'Analysis and simulation of a fair queueing algorithm', *Internetw. Res. Exper.*, 1990, **1**, pp. 3–26
19 Rojas-Cessa, R., Oki, E., and Chao, H.J.: 'CIXOB-*k*: Combined input-crosspoint-output buffered packet switch'. Proc. IEEE Globecom, 2001, pp. 2654–2660
20 Li, S.Q.: 'Performance of a non-blocking space-division packet switch with correlated input traffic'. Proc. IEEE Globecom'89, Vol. 3, pp. 1754–1763
21 Liew, S.C.: 'Performance of various input-buffered and output-buffered ATM switch design principles under bursty traffic: simulation study', *IEEE Trans. Commun.*, 1994, **42**, pp. 1371–1379