EARLIEST DUE DATE FIRST MATCHING FOR INPUT-QUEUED CELL SWITCHES

Shizhao Li[1] *Jinhui Li*[1] *Nirwan Ansari*[2] *

[1]Elect. & Comp. Engrng. Dept., New Jersey Inst. of Technology, University Heights, Newark, NJ 07102, U.S.A.
[2]Information Engrng. Dept., Chinese Univ. of Hong Kong, Sha Tin, Hong Kong (on leave from NJIT)

ABSTRACT

The input-queued switching architecture is becoming an attractive alternative for high speed switches owing to its scalability. Tremendous efforts have been made to overcome the throughput problem caused by the contentions occurred at input and output sides of a switch. Existing input queueing algorithms mostly aim at improving throughput without considering QoS features. In this paper, a new algorithm, referred to as earliest due date first matching (EDDFM), is introduced to improve upon existing algorithms in term of probability of cell delay overdue. It is shown both analytically and by simulations that EDDFM is stable and non-starving. Simulations also demonstrate that it has lower cell delay overdue probability than previously proposed algorithms.

1. INTRODUCTION

The input-queued switching architecture has been adopted for high speed switch implementation owing to its scalability. A major problem with this architecture is the head-of-line blocking (HOL) [1], which limits the throughput of an input-queued switch to 58.6% under Bernoulli traffic when a single FIFO queue is used in each input.

Previous research has shown that the throughput of an input-queued switch can be improved by using well designed buffering schemes and scheduling algorithms. The HOL blocking can be completely eliminated by adopting virtual output queueing, in which multiple virtual output queues (VOQs) directed to different outputs are maintained at each input. However, the contention among inputs still limits the throughput, i.e., if more than one input sends cell to the same output, only one out of the competing cells can get through the switch fabric. Moreover, when more than one cell can be accessed by the scheduler in one input, selecting different cells for transmission could lead to different throughput, owing to the interdependence of the inputs.

Maximizing the throughput is similar to the matching problem in a bipartite graph [2]. An iterative algorithm called *i*SLIP, which is a maximum size matching based scheduler, can achieve 100% throughput for independent and symmetric traffic [3]. Round robin scheduler, which has low implementation complexity, is adopted in *i*SLIP to resolve the contention among cells stored in the same input. However, the priority of a round robin scheduler is not a function of the queue length. Thus, *i*SLIP performs poorly for non-symmetric traffic, in which the average queue length of the FIFO queues could differ strikingly under loaded traffic. Maximum weight matching can achieve high throughput under both symmetric and non-symmetric traffic in which each session is assigned a weight and a match with the maximum aggregate weight is obtained. Longest queue first (LQF) [4], oldest cell first (OCF) [5], and longest normalized queue first (LNQF) [6] are among the maximum weight matching approach. The weights in LQF and LNQF are set according to the queue length and normalized queue length of each VOQ, respectively, while the weight in OCF is set to the delay time of head cell of each VOQ. None of them considers the actual delay requirement of each session. As a result, these algorithms may generate adverse effect on the probability of cell delay overdue. Noting that delay due date should be incorporated in the scheduler design, we thus propose an algorithm appropriately called earliest delay due date first matching (EDDFM).

The rest of the paper is organized as follows. In Section 2, we describe our switch and traffic models. Section 3 describes the proposed algorithm and presents the proof that EDDFM is stable for all admissible independent traffic. The performance of the proposed algorithm is presented in Section 4. Concluding remarks are given in Section 5.

2. SWITCH AND TRAFFIC MODELS

Consider an $N \times N$ input-queued cell switch consisting of N inputs, N outputs and a non-blocking switch fabric. To eliminate the HOL blocking, virtual output queueing is adopted, as shown in Fig. 1.

Let $Q_{i,j}$ denote the VOQ directed to output j at input i, and $A_{i,j}$ denote the arrival process to $Q_{i,j}$. To provide QoS features, switch resources such as the bandwidth and storage should be allocated on a per-session basis. There could be more than one session arrived at a certain input directed to the same output. Thus, multiple sessions could share the same VOQ, each of which is maintained as a FIFO queue. Let $I_{i,j,k}$ be the kth session in $Q_{i,j}$ with arrival rate $\lambda_{i,j,k}$. Therefore, the arrival rate of $A_{i,j}$ can be expressed as $\lambda_{i,j} = \sum_k \lambda_{i,j,k}$. An arrival process A_i, which is the aggregate arrival process to input i, is said to be uniform if $\lambda_{i,m} = \lambda_{i,n}$, $\forall\ m \neq n$, $1 \leq m, n \leq N$. Otherwise, the process is said to be non-uniform. The traffic pattern is

*This work has been supported in part by Lucent Technologies.

admissible if and only if $\sum_{j=1}^{N} \lambda_{i,j} \leq 1$ and $\sum_{i=1}^{N} \lambda_{i,j} \leq 1$.

The traffic in a real network is highly correlated from cell to cell, and cells tend to arrive at the switch in "bursts." One way of modeling a bursty source is by using an ON-OFF model in the discrete-time domain. This model is equivalent to a two-state Markov Modulated Deterministic Process (MMDP)[7]. The two states, OFF state and ON state, are shown in the Fig. 2. In the OFF state, the source does not send any cells. In the ON state, the source sends data cells at the peak cell rate (P). The source can independently shift from one state to another as shown in Fig. 2. In a discrete-time domain, state changes may occur only at the end of a time-slot. At each time slot, the source in the OFF state changes to the ON state with a probability α. Similarly, the source in the ON state changes to the OFF state with a probability β. It must be remembered that there is no correlation between the two probabilities. The probabilities of the source being in the OFF state and ON state are given by $P_{off} = \frac{\beta}{(\alpha+\beta)}$ and $P_{on} = \frac{\alpha}{(\alpha+\beta)}$, respectively.

The bursty source is characterized by the peak cell rate (P), the average cell rate (A), and the average number of cells per burst (B). The burstiness of the traffic is defined as the ratio of the peak cell rate and average cell rate. Given these parameters, the state transition probabilities can be computed as $\alpha = \frac{A}{B(P-A)}$ and $\beta = \frac{1}{B}$.

3. THE EDDFM ALGORITHM

The basic objective of scheduling an input-queued switch is to find a contention free matching based on the connection requests. At the beginning of every time slot, each input sends requests to the scheduler. The scheduler selects a matching between the inputs and outputs with the constraints of unique pairing, i.e., at most one input can be matched to each output, and vice versa. At the end of the time slot, a cell is transmitted per matched input-output pair.

Finding a contention free matching between inputs and outputs is equivalent to solving a bipartite graph matching problem, as shown in Fig. 3(a). Each vertex on the left side represents an input, and that on the right side represents an output. An edge connects input vertex i and output vertex j if there are cells stored in $Q_{i,j}$. Associated with each edge is a weight, which is defined differently by different algorithms. For example, setting weight as the queue length of the VOQs leads to LQF, and setting weights as the delay time of the head cell in the VOQs leads to OCF. A maximum weight matching algorithm computes a match which can maximize the aggregate weight. Note that maximum size matching in which the number of connections between the input and output is maximized is a special case of maximum weight matching with the weights of the non-empty VOQs set to 1 and those of empty VOQs set to 0, respectively. A maximum weight match and a maximum size match for the same request graph can be different as shown in Fig. 3(b) and 3(c), respectively. It was shown that maximum size matching is not stable for non-symmetric traffic [4]. Therefore, maximum weight matching is adopted in EDDFM.

When a cell belonging to session $I_{i,j,k}$ in $Q_{i,j}$ arrives at time slot n, the EDDFM algorithm sets the initial weight of the cell to $P_{i,j,k}(n) = P_m - DB_{i,j,k}$, where $DB_{i,j,k}$ is the delay bound of session $I_{i,j,k}$ and P_m is an integer that is greater than $max_{\forall i,j,k}(DB_{i,j,k})$. Then the cell is inserted in the queue at the position closest possible to the HOL such that all the cells beyond this cell have smaller weights. If a cell in the queue is served in a time slot, it will be deleted from the queue; Otherwise, its weight will increase by one. The weight of $VOQ\ Q_{i,j}$ at time slot n, $w_{i,j}(n)$, is set to the weight of the HOL cell of this queue.

Let $\underline{W}_i(n) = (w_{i,1}(n), w_{i,2}(n), \ldots, w_{i,N}(n))^T$ be the weight vector of input i and $S = [S_{i,j}(n)]$ be the service matrix which indicates the match between input and outputs. $S_{i,j}(n)$ is set to 1 if input i is scheduled to transmit a cell to output j. Otherwise, $S_{i,j}(n)$ is set to 0. Let $\underline{S}_i(n) = (S_{i,1}(n), S_{i,2}(n), \ldots, S_{i,N}(n))^T$ be the service vector associated with input i. The EDDFM scheduler, as shown in Fig. 4, performs the following for each time slot n:

1. Each input i set the weight of every $VOQ\ Q_{i,j}$ to the weight of the HOL cell, and sends the weight vector $\underline{W}_i(n)$ to the scheduler;
2. The scheduler searches for a match that achieves the maximum aggregate weight under the constraint of unique pairing, i.e.,

$$\arg\max_{S}[\sum_{i,j} S_{i,j}(n) w_{i,j}(n)]$$

   such that $\sum_i S_{i,j}(n) = \sum_j S_{i,j}(n) = 1$, sends the service vector $\underline{S}_i(n)$ to the corresponding input, and uses the service matrix $[S_{i,j}(n)]$ to configure the fabric;
3. Each input selects the HOL cell from the matched VOQ indicated by $\underline{S}_i(n)$ for transmission.

Lemma 1 *Using EDDFM algorithm, the weights of the VOQs are stable for all admissible independent traffic, i.e.,* $E[W_{i,j}(n)] < \infty, \forall i, j, n$.

Proof: Considering $w_{i,j}(n+1)$, the weight of $Q_{i,j}$ at time slot $n+1$, we get:

$$w_{i,j}(n+1) = 0, \tag{1}$$

if $Q_{i,j}$ is empty at time slot $n+1$.

$$w_{i,j}(n+1) = P_{i,j,k}(n+1), \tag{2}$$

if a cell arrives at $Q_{i,j}$ and its weight is larger than the weight of the HOL cell.

$$w_{i,j}(n+1) = w_{i,j}(n) + 1, \tag{3}$$

if the queue is not served in this time slot, and no cell with a larger weight arrives.

$$w_{i,j}(n+1) = w_{i,j}(n) - \tau_{i,j}(n) + 1, \tag{4}$$

otherwise. $\tau_{i,j}(n)$ is the difference of the weights between HOL cell and the cell behind it at time slot n, and $\tau_{i,j}(n) \geq 0$.

Let the approximate next-state weight of $Q_{i,j}$

$$\tilde{w}_{i,j}(n+1) = w_{i,j}(n) + 1 - S_{i,j}(n)\tau_{i,j}(n), \quad (5)$$

we can obtain:

$$w_{i,j}(n+1) \leq max[\tilde{w}_{i,j}(n+1), P_m] \quad (6)$$

Define $\underline{W}(n)$=$[w_{1,1}(n), ..., w_{1,N}(n), w_{2,1}(n), ..., w_{N,N}(n)]^T$ and $\underline{\tilde{W}}(n+1)$=$[\tilde{w}_{1,1}(n+1), ..., \tilde{w}_{1,N}(n+1), \tilde{w}_{2,1}(n+1), ..., \tilde{w}_{N,N}(n+1)]^T$, then we can obtain the following inequality:

$$\underline{W}^T(n+1)Q\underline{W}(n+1) \leq \underline{\tilde{W}}^T(n+1)Q\underline{\tilde{W}}(n+1) + P_m^2 N. \quad (7)$$

Similar to the proof of the stability of OCF in [5], we can prove that the weights of the VOQs are stable under the EDDFM algorithm.

Let $L_{i,j}(n)$ denotes the queue occupancy of $Q_{i,j}$ at time slot n. A switch is stable if $E[L_{i,j}(n)] < \infty, \forall i, j, n$.

Theorem 1 *A switch using EDDFM algorithm is stable for all admissible independent traffic.*

Proof: Under EDDFM algorithm, the queue occupancy is always less than or equal to the weight of HOL cell, so the queue occupancies are stable.

Theorem 2 *Under EDDFM algorithm, no session will be starved.*

Proof: A cell's weight will keep increasing until it is served. So EDDFM is a starvation-free algorithm.

4. PERFORMANCE

A 4×4 input-queued switch was considered for simulations in which the bursty traffic was generated based on the on-off traffic model. The average burst length was chosen to be 20 cells and the burstiness was 2. The traffic was non-symmetric, i.e., the arrival rates of the VOQs in the same input were different, and were 0.5, 1, 2 and 5Mbps. Two sessions in each VOQ, a fast session with a rate four times that of a slow session, were generated. A traffic load of 0.9 was assumed, and each simulation lasted through 100 seconds.

Three levels of delay bound, which are short, medium and long, were assumed in the simulation. The delay bounds are assigned according to the following rules: sessions with rates over 10% of the link capacity are treated as fast sessions and are assigned short delay, sessions with rates between 1% and 10% of the link capacity are treated as medium sessions and are assigned short and medium delay randomly, sessions with rates less than 1% of the link capacity are treated as slow sessions and are assigned short, medium and long delay randomly. The medium delay and long delay are set to five times and ten times of short delay, respectively. The configuration of delay bounds of each session remain the same for different algorithm for comparison purpose. The probabilities of cell overdue for different algorithms are shown in Fig. 5. The values of the delay bound in the figure are associated with short delay.

The fairness of a scheduler is defined as [8]:

$$F_S = \max_{\forall i,j,\ j \neq i} \left| \frac{W_i(t_1, t_2)}{\lambda_i} - \frac{W_j(t_1, t_2)}{\lambda_j} \right|,$$

where $W_i(t_1, t_2)$ is the number of cells delivered for session i during the time interval $[t_1, t_2]$, and λ_i is the rate of session i. The fairness is the maximum difference of the normalized service time, which is the service a session received normalized by its rate, among all sessions. It provides a metric on how fair a server is. The smaller the amount of fairness, the fairer the server is.

Table 1 summarizes the performance comparison among EDDMF, LNQF, LQF and OCF. The following conclusions can be derived from Table 1 and Fig. 5.

1. The expected queue length of an input buffer in a switch using EDDFM is in the same order as the other algorithms, and thus the switch is stable.
2. EDDFM provides comparable delay for each session, implying that EDDFM is a non-starvation algorithm.
3. The fairness of EDDFM is in the same order as LNQF and OCF, and is smaller than that of LQF.
4. EDDFM has the lowest probability of cell overdue.
5. The average time for EDDFM to complete transmitting a burst are much smaller than LNQF and LQF, implying that EDDFM does not perform well in reducing burstiness.

5. CONCLUSIONS

We have proposed a new algorithm, EDDFM, to improve upon existing algorithms in terms of probability of cell delay overdue and fairness. EDDFM is proved analytically to be stable and starvation-free under all admissible independent traffic patterns. Simulations also show that EDDFM has lower probability of cell overdue than LNQF, LQF, and OCF.

REFERENCES

[1] M. J. Karol, M. G. Hluchyj and S. P. Morgan, "Input versus output queueing on a space-division packet switch," *IEEE Trans. Commun.*, vol. COM-35, pp. 1347-1356, Dec. 1987.

[2] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms,* New York: McGraw Hill, 1989.

[3] N. McKeown, P. Varaiya and J. Walrand, "Scheduling cells in an input-queued switch," *IEE Electronics Letters*, pp. 2174-2175, Dec. 9th, 1993.

[4] N. McKeown, V. Anantharam and J. Walrand, "Achieving 100% throughput in an input-queued switch," *Proceedings of INFOCOM'96*, pp. 296-302, 1996.

[5] A. Mekkittikul and N. McKeown, "A starvation-free algorithm for achieving 100% throughput in an input-queued switch," *Proceedings of ICCCN'96*, pp. 226-231, 1996.

[6] S. Li, and N. Ansari, "Scheduling input-queued ATM switches with QoS features" *Proceedings of ICCCN'98*, pp. 107-112, Lafayette, LA, Oct. 1998.

[7] R. Bolla, F. Davoli, and M. Marchese, "Evaluation of a cell loss rate computation method in ATM multiplexers with multiple bursty sources and different traffic classes," *Proceedings of GLOBECOM'96*, pp. 437-441, Nov. 1996.

[8] S. Golestani, "A self-clocked fair queueing scheme for broadband applications," *Proceedings of INFOCOM'94*, pp. 636-646, 1994.

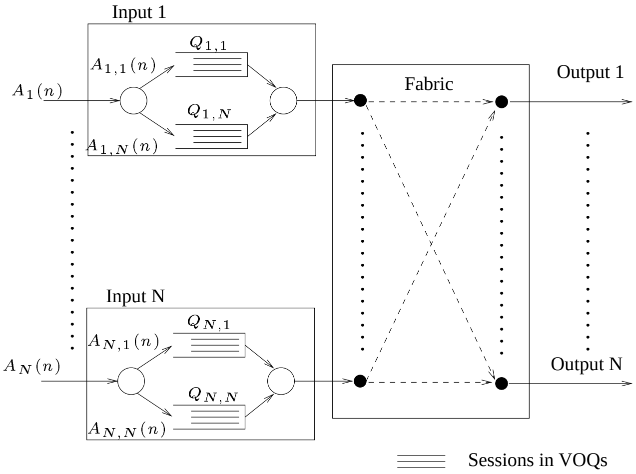

Figure 1. Input-queue switch model

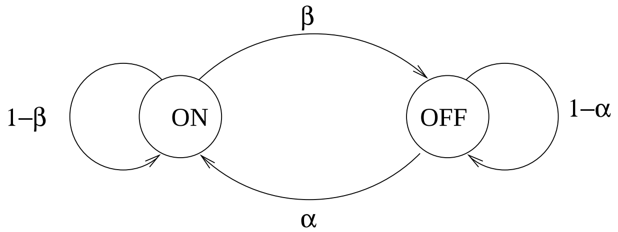

Figure 2. Simple ON-OFF traffic model

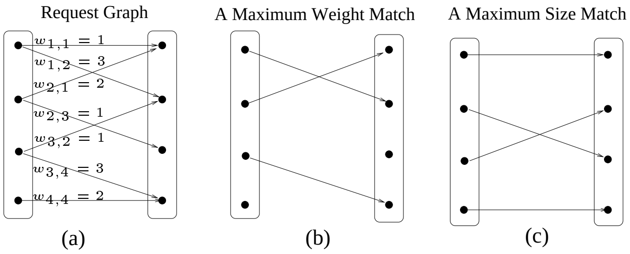

Figure 3. A Bipartite graph matching example:(a) the request graph, (b) a maximum weight match, and (c) a maximum size match.

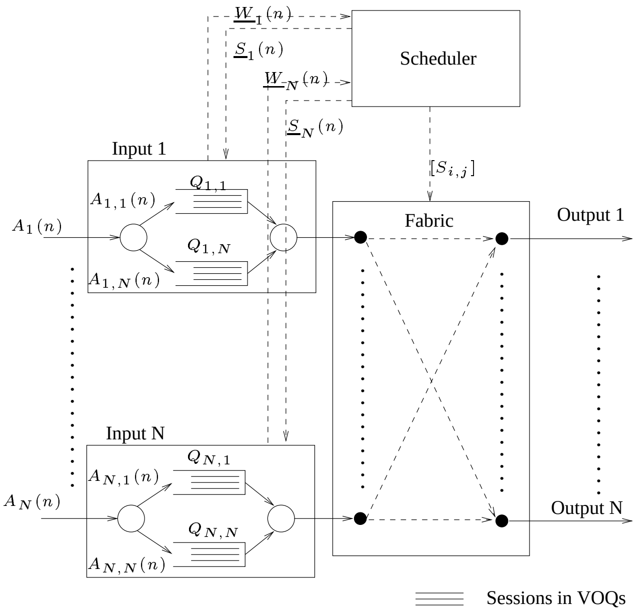

Figure 4. LNQF scheduler

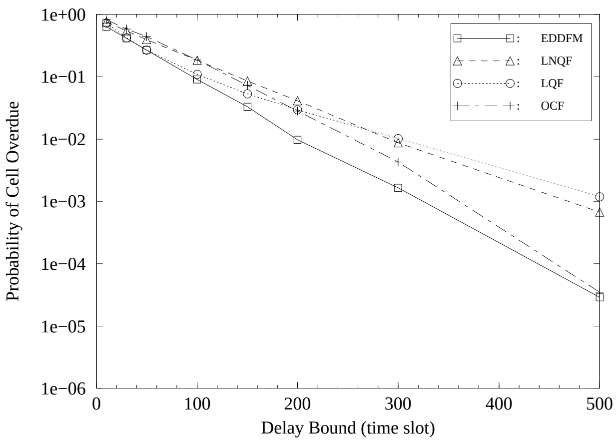

Figure 5. Comparison of probability of cell overdue

Table 1. Statistics of the simulation results

schedulers	EDDFM	LNQF	LQF	OCF
average delay(time slot) of VOQ1 session 1 λ=0.1Mbps	89.0	22.5	535.6	62.7
average delay(time slot) of VOQ1 session 2 λ=0.4Mbps	75.4	40.8	296.6	63.8
average delay(time slot) of VOQ2 session 1 λ=0.2Mbps	62.4	38.7	193.7	66.8
average delay(time slot) of VOQ2 session 2 λ=0.8Mbps	44.7	59.7	85.1	67.4
average delay(time slot) of VOQ3 session 1 λ=0.4Mbps	73.5	49.4	114.7	64.8
average delay(time slot) of VOQ3 session 2 λ=1.6Mbps	51.9	64.4	44.6	68.2
average delay(time slot) of VOQ4 session 1 λ=1Mbps	47.7	52.1	76.0	57.9
average delay(time slot) of VOQ4 session 2 λ=4Mbps	47.4	62.4	24.7	59.5
average queue length (cell)	50.3	61.3	62.2	64.5
fairness	7.5	10.4	20.0	8.1
average transmission time(time slot) per burst:	97.0	125.9	130.7	97.6

EARLIEST DUE DATE FIRST MATCHING FOR INPUT-QUEUED CELL SWITCHES

Shizhao Li[1] , *Jinhui Li*[1] *and Nirwan Ansari*[2] [1]
[1]Elect. & Comp. Engrng. Dept., New Jersey Inst. of Technology, University Heights, Newark, NJ 07102, U.S.A.
[2]Information Engrng. Dept., Chinese Univ. of Hong Kong, Sha Tin, Hong Kong (on leave from NJIT)

The input-queued switching architecture is becoming an attractive alternative for high speed switches owing to its scalability. Tremendous efforts have been made to overcome the throughput problem caused by the contentions occurred at input and output sides of a switch. Existing input queueing algorithms mostly aim at improving throughput without considering QoS features. In this paper, a new algorithm, referred to as earliest due date first matching (EDDFM), is introduced to improve upon existing algorithms in term of probability of cell delay overdue. It is shown both analytically and by simulations that EDDFM is stable and non-starving. Simulations also demonstrate that it has lower cell delay overdue probability than previously proposed algorithms.