

입출력 버퍼형 ATM 스위치의 단순 셀 스케줄링 알고리즘

한만수*, 한인택*, 이범철*

*한국전자통신연구원 교환전송기술연구소

e-mail : hanms@etri.re.kr

Simple Cell Scheduling Algorithm for Input and Output Buffered ATM Switch

Man-Soo Han*, In-Tak Han*, Beom-Cheol Lee*

*Switching & Transmission Technology Lab., ETRI

요 약

입출력버퍼형 스위치를 위한 간단한 셀 스케줄링 알고리즘을 제시한다. 스위치는 고속동작 및 성능향상을 위해 이중 스위칭 플랜을 갖고 있다. 제안한 알고리즘은 각각의 스위칭 플랜에서 독립적으로 수행되며 전송요청 (request), 전송허가(grant), 전송확정 (accept)의 3 단계 동작으로 이루어져 있다. 또한 각 3 단계동작을 한 셀시간에 한 번씩만 수행하여 단위 셀시간이 작은 고속 스위칭에 적합하다. 모의실험 결과 제안한 알고리즘의 성능이 Bernoulli 트래픽 입력에 대해 출력버퍼형 스위치의 성능과 거의 동일하였다.

1. Introduction

An input buffered switch suffers from a low performance of 58.6% due to head-of-line (HOL) blocking whereas its hardware complexity is simpler than that of an output buffered switch. Many studies have attempted to improve the performance of the input buffered switch including an input buffered architecture with iterative matching scheme [1-3] or an input and output buffered architecture with speedup [4][5].

Iterative matching algorithms, iSLIP [1], FIRM [2] and PIM [3] are introduced to enhance the switch performance. The more the algorithms are iterated, the better the performance is. The iterative scheme, however, prevents each algorithm operating at high-speed since an iteration requires an additional processing time. In [4], to emulate an output buffered switch, the critical cells first algorithm with speedup of two is introduced. Also, in [5], the lowest occupancy output first algorithm with speedup of two is introduced. Even though both methods emulate an output buffered switch, they are too complex to implement using the current technology.

In this paper, we suggest a simple matching algorithm (SMA) for the input and output buffered switch with dual switching planes. Applying the architecture of multiple switching planes [6], the dual switching planes are used to

achieve speedup of two (see Fig. 1). SMA consists of three steps: request, grant and accept steps. The three steps of SMA operate in parallel on each switching plane. The operation of each step is based on the operation of iSLIP. But instead of using the pointer update scheme of iSLIP, SMA uses a simple round-robin scheme at the pointer update operation.

During a cell time, SMA operates just once whereas the algorithms of [4] and [5] operate twice and the algorithms of [1], [2] and [3] have to operate more than twice for a better performance. SMA and the dual switching planes make the performance almost identical to the performance of output buffered switch. Therefore, SMA fits for high-speed and high-performance switches. We first describe the switch architecture and SMA. Next, under Bernoulli traffic, we show the performances of SMA is almost identical to that of output buffered switch by simulations.

2. Simple Matching Algorithm

The input and output buffered switch is comprised of three main blocks: N input buffer modules, N output buffer modules, and two $N \times N$ space-division switching planes where N is the number of input and output ports (see Fig. 1). The bandwidth of a switching plane is same to the bandwidth of input and output ports. i -th input buffer

module contains N FIFO queues, Q_{ij} , $j = 1, \dots, N$, where $i = 1, \dots, N$. In i -th input buffer module, a cell arrived from an input port routes to Q_{ij} if it is destined for j -th output buffer module. Using HOL status of each Q_{ij} , SMA selects FIFO queues to serve and determines which switching plane each selected queue uses. Since dual switching planes are used, each input buffer module can transmit two cells during a cell time. Also, each output buffer module can receive two cells during a cell time.

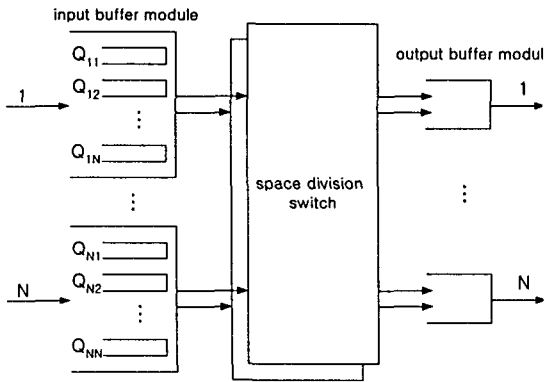


Fig. 1. Input and output buffered switch with dual switching planes

We now describe the operation of SMA. SMA uses accept round-robin pointer (ARP) and grant round-robin pointer (GRP). For i -th input buffer module, two ARPs A_{i1} and A_{i2} are assigned to 1st and 2nd switching planes, respectively, where $i = 1, \dots, N$. Also, for j -th output buffer module, two GRPs G_{j1} and G_{j2} are assigned to 1st and 2nd switching planes, respectively, where $j = 1, \dots, N$. The aim of SMA is to maximally match an A_{ik} to a G_{jk} when nonempty Q_{ij} 's are given where $k = 1, 2$. If ARP A_{ik} is matched with GRP G_{jk} , then Q_{ij} is said to be accepted for k -th switching plane and the HOL cell of Q_{ij} will use k -th switching plane in the transmission.

A pointer value g_{jk} represents the highest priority element of g_{jk} where $1 \leq g_{jk} \leq N$. When $g_{jk} = 1$, G_{jk} checks if it received a request from a queue Q_{ij} , $i \in [1, N]$, in order of $Q_{1j}, Q_{(1+1)j}, \dots, Q_{Nj}, Q_{1j}, \dots, Q_{(1-1)j}$. G_{jk} grants the first request at the check process. Similarly, a pointer value a_{ik} represents the highest priority element of A_{ik} where $1 \leq a_{ik} \leq N$. When $a_{ik} = 1$, A_{ik} checks if it received a grant from a G_{jk} , $j \in [1, N]$, in order of $G_{jk}, G_{(j+1)k}, \dots, G_{Nk}, G_{1k}, \dots, G_{(j-1)k}$. A_{ik} accepts the first grant at the check process.

The initial value of a_{i1} is

$$a_{i1} = N - (i - 1), \quad (1)$$

while the initial value of g_{j1} is

$$g_{j1} = N - (j - 1). \quad (2)$$

On the other hand, the initial value of a_{i2} is given by (modulo N)

$$a_{i2} = \frac{N}{2} - (i - 1). \quad (3)$$

Also, the initial value of g_{j2} is given by (modulo N)

$$g_{j2} = \frac{N}{2} - (j - 1). \quad (4)$$

At the end of each cell time, every pointer value a_{ik} and g_{jk} is increased (modulo N) by one independently of the match results of the current cell time. In iSLIP, the pointer update is dependent on the match result [1]. Therefore, the pointer update scheme of SMA is simpler than that of iSLIP.

At the beginning of each cell time, the match process begins. All ARPs and GRPs are initially unmatched. The three steps operate in parallel on each switching plane and are as follows:

- **Step 1. Request.** If Q_{ij} has an HOL cell, Q_{ij} sends a request to GRP G_{jk} , $j \in [1, N]$.
- **Step 2. Grant.** If G_{jk} receives any requests, it grants the nearest request from g_{jk} -th element. G_{jk} notifies each A_{ik} , $i \in [1, N]$, whether or not its request was granted. At the end of each cell time, every g_{jk} is increased (modulo N) by one.
- **Step 3. Accept.** If ARP A_{ik} receives any grants, it accepts the nearest grant from a_{ik} -th element. At the end of each cell time, every a_{ik} is increased (modulo N) by one.

From the pointer update mechanism and the initial values of the pointers, if $a_{i1} = j$ then we have $g_{j1} = i$; if $a_{i2} = j$ then we have $g_{j2} = i$. That means if $g_{jk} = i$ and G_{jk} grants the i -th element, then the grant is always accepted since there exists a unique A_{ik} such that $a_{ik} = j$. In addition, $|a_{i1} - a_{i2}| = |g_{j1} - g_{j2}| = \frac{N}{2}$ and every a_{ik} and g_{jk} is increased (modulo N) by one at the end of each cell time. Therefore, every HOL cell can be served within $\frac{N}{2}$ cell times, i.e., the service guarantee [2] is $\frac{N}{2}$ cell times.

If $|a_{i1} - a_{i2}|$ and $|g_{j1} - g_{j2}|$ are greater or less than $\frac{N}{2}$, every HOL cell can not be served within $\frac{N}{2}$ cell times.

For instance, assume that the initial values of a_{i1} and g_{j1} are given by Eqs (1) and (2), respectively. Also, suppose that the initial values of a_{i2} and g_{j2} are $a_{i2} = \frac{3N}{4} - (i-1)$ and $g_{j2} = \frac{3N}{4} - (j-1)$, respectively. Then, the service guarantee becomes $\frac{3N}{4}$ cell times. The initial value assignment of Eqs (1), (2), (3) and (4) yields the tightest service guarantee. The service guarantee of iSLIP is $N^2 + (N-1)^2$ cell times while that of FIRM is N^2 cell times [2]. Therefore, SMA is more adequate than iSLIP and FIRM for a real time traffic since a tight service guarantee is desired for the transmission of real time traffic.

A Q_{ij} can be accepted for both switching planes since SMA operates independently on each switching plane. In that case, if the Q_{ij} has more than two cells, it transmits the first two cells using both switching planes; if the Q_{ij} has only one cell, it transmits the cell using one of two switching planes.

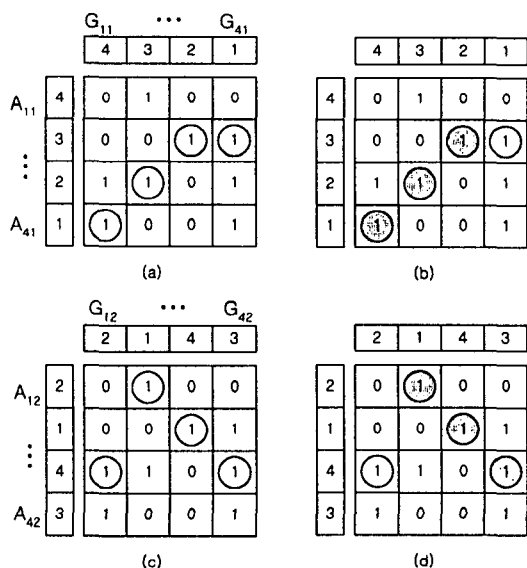


Fig. 2. Example of SMA

Fig. 2 shows an example of SMA with $N = 4$. Figs. 2 (a) and 2 (c) illustrate the grant operation for switching planes 1 and 2, respectively. Figs. 2 (b) and 2 (d) depict the accept operation for switching planes 1 and 2, respectively. 4×4 box array represents the HOL status of Q_{ij} . If Q_{ij} has an HOL cell, then the inside number of the box is 1; if Q_{ij} is empty, then the inside number of the box is 0.

In the request operation, Q_{ij} sends a request to G_{jk} , $j \in [1,4]$, if Q_{ij} has an HOL cell. For example, since Q_{31} and Q_{41} have an HOL cell, they send a request to G_{11} and

G_{12} . But, Q_{11} and Q_{21} do not send a request to any GRP since they are empty.

In the grant operation, each G_{jk} grants the nearest request from its pointer value (represented by an integer in a box) in the j -th column. The circle represents its corresponding element is granted. For example, G_{12} grants the request of Q_{31} since Q_{31} is the first nonempty queue from the highest priority element (i.e., Q_{21}) in the first column.

In the accept operation, each A_{ik} accepts the nearest grant from its pointer value (represented by an integer in a box) in i -th row. The filled circle means its corresponding element is accepted. For example, A_{22} accepts the grant of G_{32} since G_{32} is the first grant from the highest priority element (i.e., G_{12}) in the second row. Note that Q_{23} is accepted for both switching planes. If Q_{23} has more than two cells, it transmits the first two cells using both switching planes; if Q_{23} has only one cell, it transmits the cell using one switching plane. At the end of cell time, a_{ik} and g_{jk} are increased (modulo 4) by one. That is, $\{G_{11}, \dots, G_{41}\}$ and $\{A_{11}, \dots, A_{41}\}$ become $\{1, 4, 3, 2\}$ from $\{4, 3, 2, 1\}$; $\{G_{12}, \dots, G_{42}\}$ and $\{A_{12}, \dots, A_{42}\}$ become $\{3, 2, 1, 4\}$ from $\{2, 1, 4, 3\}$.

3. Simulation Results

The traffic model of simulations is independent Bernoulli arrivals with destinations uniformly distributed over all outputs. For a 64×64 switch, we compare the performance of the output buffered switch (OBS) with that of SMA. The simulation time of each method is 100,000 cell time slots.

Unlike iSLIP, FIRM and PIM, we do not need to iterate an algorithm during a cell time to improve the performance. The three steps of SMA operate just once during a cell time. It is clear that SMA fits for high-speed switching compared with iSLIP, FIRM and PIM. As it can be observed from Figs. 3 and 4, the performance of SMA, the mean delay and cell delay variation, is almost identical to that of output buffered switch.

4. Conclusions

We present a simple cell scheduling algorithm, SMA, for an input and output buffered switch with dual switching planes. SMA independently operates on each switching plane and consists of request, grant and accept steps. SMA is more adequate for high-speed switching compared with other iterative algorithms since the three steps of SMA operate just once during a cell time. Moreover, SMA provides the tightest service guarantee compared with iSLIP and FIRM. Under Bernoulli traffic, SMA and the architecture of dual switching planes make the switch performance almost identical to that of output buffered switch.

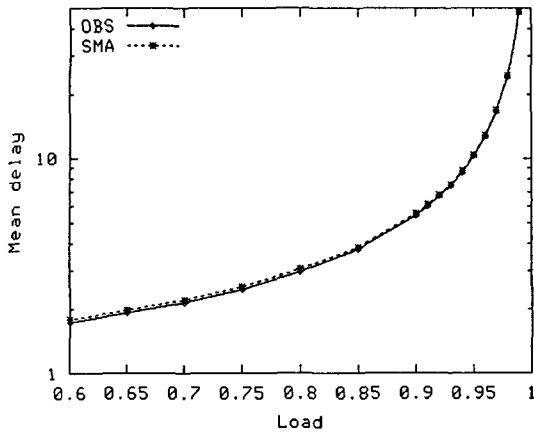


Fig. 3. Mean delay (cell) for 64×64 switch

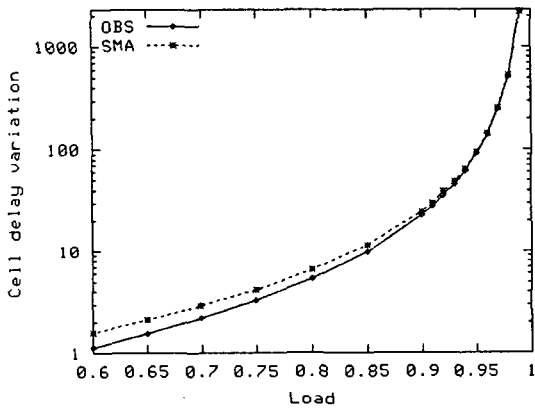


Fig. 4. Cell delay variation (cell^2) for 64×64 switch

“On the speedup required for work-conserving crossbar switches,” *IEEE J. Select. Areas Communi.*, Vol. 17, No. 6, pp. 1057-1066, June 1999.

- [6] M. S. Han, J. H. Lee and B. C. Lee, “Fast scheduling algorithm for input and output buffered ATM switch with multiple switching planes,” *Electronics Letters*, Vol. 35, No. 23, pp. 1999-2000, Nov. 1999.

References

- [1] N. McKeown, “The iSLIP scheduling algorithm for input-queued switches,” *IEEE/ACM Trans. Networking*, Vol. 7, No. 2, pp. 188-201, April 1999.
- [2] D. N. Serpanos and P. I. Antoniadis, “FIRM: High-speed Distributed Scheduling for ATM Switches with Multiple Input Queues,” *Electronics Letters*, Vol. 35, No. 22, pp. 1915-1916, Oct. 1999.
- [3] T. E. Anderson, S. S. Owicki, J. B. Saxe and C. P. Thacker, “High-speed switch scheduling for Local-Area Network,” *ACM Trans. Computer Systems*, Vol. 11, No. 4, pp. 319-352, Nov. 1993.
- [4] S. T. Chuang, A. Goel, N. McKeown and B. Prabhaker, “Matching output queueing with a combined input/output-queued switch,” *IEEE J. Select. Areas Communi.*, Vol. 17, No. 6, pp. 1030-1039, June 1999.
- [5] P. Krishna, N. S. Patel, A. Charny and R. J. Simcoe,