

# Design of an Input-queued ATM Switch supporting multicast and Research on its Scheduling Policy

ZHAI Mingyu, ZHAO Qi, LUO Junzhou, GU Guanqun  
{myzhai,qzhao}@seu.edu.cn

(Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China)  
(The Key Laboratory of Computer Network and Information Integration, Ministry of Education, China, 210096)

**Abstract** Currently the research on input-queued ATM switches is one of the most active research fields. Many achievements have been made in the research on scheduling algorithms for unicast input-queued ATM switches and also applied in commerce. But the goal of the research on scheduling algorithms for multicast input-queued ATM switches only focuses on providing high throughput and inadvertently ignoring its undesired effects on QoS of the multicast traffic. In this paper we present a design scheme of input-queued ATM switches supporting multicast and corresponding scheduling algorithm, referred to as multicast longest normalized queue first (MLNQF). The algorithm MLNQF has the characteristics of improving throughput, satisfying QoS requirements and providing service fairly.

## 1. INTRODUCTION

Currently the research on input-queued ATM switches is one of the most active research fields. The advantages of input-queued technique are that it owns the simple Switch Fabric and low requirement for inner switch speed, but the disadvantages are the difficulty of schedule because a cell has to not only compete for the output ports with other cells, but also compete with the cells at the same input port. The input-queued technique usually uses a FIFO queue, in this case, the research of Karol[4] concluded the throughput of input-queued ATM switches only arrives at 58.6% at the condition of unicast and average flow due to the existence of HOL(Head of Line) congestion. HOL congestion is that when a cell at the head of FIFO queue failed in competing for the output, even if the output port correspondent with the successive cell is idle, this successive cell still cannot output due to the congestion of the cell at the head of queue. The effect of HOL congestion is more serious to the input-queued ATM switches supporting multicast, since the cell at the head could wish to output more than one output ports, which makes the throughput smaller. Therefore, the input-queued ATM switches are not thought to be in practice. Currently, the research on input-queued ATM switches supporting unicast only has made some progress, and it says that if some appropriate input-queued techniques (such as virtual

input-queued technique[5][6]) are used, the throughput of input-queued ATM switches could arrive at the 100%. So the research of the input-queued ATM switches automatically enter a new stage.

## 2. RELATED WORK

Until now, the research of the input-queued ATM switches supporting the multicast is still scarce[3]. Recently many practical multicast scheduling policies based on the FIFO input-queue such as randomly selecting policy[7], loop priority preserved policy [8] and concentrated scheduling policy[2] have been proposed. But in all these schemes the problem of low throughput still exists. H.Duan proposed the 3DQ scheme of high-performance input-queued ATM switches supporting multicast[1]. In 3DQ, every cell is queued according to its VC number, the output port it requested and its priority. Corresponding with the technique of 3DQ input queue, this design scheme schedules the cells of each input port by using the MUCS (Matrix Unit Cell Schedule). Through analysis, we find MUCS has the following main problems: (1) because the designer of 3DQ not only regards a multicast VC as more than one virtual unicast VCs, but also lines up them in the output-port queue by using many VC tags. Under this direction, MUCS could only schedule every input port to output one unicast cell or one copy of multicast cell in a time slot. Obviously in some flow condition (such as random heterogeneous service), this affects the throughput of switches seriously. (2) MUCS determines the schedule order according to the weight of the cell, but not the priority of the cells, which could be the unfairness to some VCs.

## 3. OUR DESIGN SCHEME.

### 3.1 The model of switches framework using MVOQ (Multicast Virtual Output Queue)

The research of scheduling algorithms[5][6] for unicast input queue indicates that the virtual output queue technique

---

This work is supported by Chinese NSF (item number: 98046) and "973" project (item number: G1998030405). Zhai Mingyu, Ph.D candidate, department of computer science and engineering, Southeast Univ. His research interest includes high-speed network, and network security. Email: myzhai@seu.edu.cn. Zhao Qi, research assistant, department of computer science and engineering, Southeast Univ. Her research interest includes high-speed network, and wireless network. Luo Junzhou, professor, department of computer science and engineering, Southeast Univ. His research interest includes computer network, and Petri Net. Gu Guanqun, professor, Academician of Chinese Engineering Academy, department of computer science and engineering, Southeast Univ. His research interest includes computer network, and distributed computing.

could eliminate completely the HOL congestion brought by FIFO, so acquire 100% throughput. Thus, we wish to introduce the MVOQ technique in our design scheme. Using this technique, when a unicast cell arrived at the input port, it is lined up in the corresponding MVOQ according to the output port it requested and waits for scheduling output. For the multicast cell, because it need be output more than one output destination ports; if the policy directly copies these multicast cells and loads them to their corresponding MVOQs, it must lead to the very large waste of input-queued cache. Based on the scheme of centered-queue ATM switches, we propose the following design scheme for multicast input-queued ATM switches.

In our scheme, the multicast input-queued ATM switches are mainly made of three parts: input-queued manager, scheduling service and switch fabric (See Fig. 1). In this

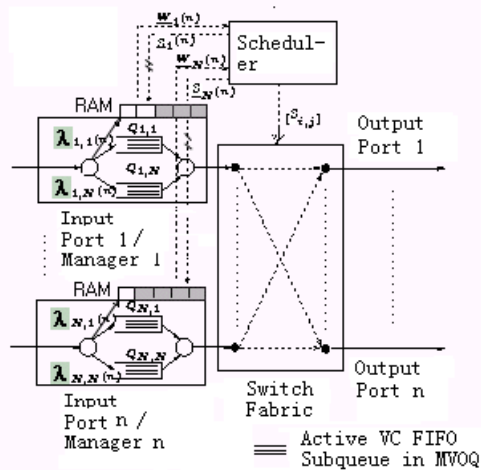


Fig1. ATM Switch Model Using MVOQ

section, we emphasize to discuss the input-queued manager, which completes the following two tasks:

1. Manage the input port buffer. The input-queued manager divides the buffer at input port into two sections: the cells loading area and the queue area. The cells loading area loads the arriving cells using RAM.
2. Using the MVOQ technique to manage cells. When a cell (unicast cell or multicast cell) arrives at the input port, the input-queued manager firstly finds an idle unit to load this cell at the cells loading areas; and uses the cell address to represent this cell, which is called the cell representative. The input-queued manager maintains a counter called the cell counter for each cell. When a cell arrive, if it is a unicast cell, set the value of its cell counter 1; if it is a multicast cell, set the value of its cell to be the number of the output destination ports. The input-queued manager maintains a queue for every output port, and this queue is made up by the FIFO sub-queue of the VC (Active VC) requesting to output to that output port. FIFO sub-queue can insure the cell scheduling order. The element of this FIFO sub-queue is not the cell essence, but the cell representative. To the unicast cell, the input-queued manager loads the cell representative to the corresponding output port sub-queue according to the output destination port this cell requests. To the multicast cell, the input-queued

manager generates some copies of the cell representative into the corresponding output port queues according to the output destination ports this cell requests. When the cell is scheduled to output one time, the cell counter decreases 1; when the value of the cell counter is 0, the input-queued manager is responsible to reclaim the buffer space it occupied. We call this technique of the input-queued manager as the MVOQ technique, which can be implemented by using the linked list.

The MVOQ technique not only makes use of the limited buffer space of the input port efficiently, but also eliminates the HOL congestion. Moreover, the MVOQ technique adopts the two-level queue, which is based on output port and VC respectively, so the implementation is relatively easy. Other than 3DQ design [1], we no longer queue up every VC in the virtual output port queue according to its priority, but let the scheduling algorithm dynamically determines how to set the priority of each VC through computation. Thus, our MVOQ design will have more wide compatibility.

### 3.2 Scheduling algorithms for multicast input queue with QoS features

Since our scheme adopts the MVOQ technique, it makes the design of scheduling algorithms very easy. Because the research on scheduling algorithms for unicast virtual input queue is very mature, we could get the scheduling algorithms for multicast input queue easily by improving these algorithms. In this section, we firstly introduce the scheduling algorithms for unicast input queue S.Li proposed LNQF (Longest Normalized Queue First)<sup>[6]</sup>, then give the improved algorithm in the environment for multicast. S.Li has proved that the LNQF algorithm has the stability of scheduling to all admissible traffic pattern flow modes, and concluded that LNQF can provide delay guarantee, throughput improvement, fair service and decrease the burst of flow.

The following are the assumptions and symbols that LNQF algorithm uses.

1. The input queues of  $N \times N$  ATM switches adopt the virtual output-queued technique, and adopt the FIFO sub-queue in every active VC. The switch fabric is non-congestion;
2. In a time slot, each input port only could schedule at most one unicast cell to output, and every output port could receive at most one input cell;
3.  $Q_{ij}$  indicates the virtual output queue corresponded to output port  $j$  in input port  $i$ ;  $I_{ij,k}$  indicates the  $k$ th active VC in  $Q_{ij}$ , and arrives at rate  $\lambda_{ij,k}(n)$  in the  $n$  time slot;  $\lambda_{ij}(n)$  indicates the sum rate of  $Q_{ij}$
4. In the  $n$ th time slot,  $l_{ij,k}(n)$  indicates the length of FIFO sub-queue, and  $l_{ij}(n)$  indicates the queue length of  $Q_{ij}$
5. In the  $n$ th time slot, the weight  $w_{ij}(n)$  of  $Q_{ij}$  is defined

$$w_{ij}(n) = l_{ij}(n) / \lambda_{ij}(n) \quad (3)$$

the weights of all virtual output queues at input port  $i$  are expressed by  $W_i(n)$ ,  $W_i(n) = (w_{i,1}(n), \dots, w_{i,N}(n))^T$ ;

6. The service vector  $S_i(n)$  at the input port  $i$  is defined as  $S_i(n) = (s_{i,1}(n), \dots, s_{i,N}(n))$ . If the cell in input port is scheduled

to output port  $j$  set  $s_{i,j}(n)$  1; otherwise, set  $s_{i,j}(n)$  0.  $S = (s_{i,j}(n))$  indicates service matrix.

Under the above conditions, the LNQF algorithm carries out the following operations in every time slot:

1. Calculate the normalized queue length of every VOQ and produce weight vector  $W_i(n)$ , then send it to the scheduler;
2. The scheduler obtains a mapping between input port and output port, satisfying  $\arg \max [\sum_{ij} s_{i,j}(n) * w_{i,j}(n)]$  and  $\sum_i s_{i,j}(n) = \sum_j s_{i,j}(n) = 1$ . Then scheduler sends the service vector  $S_i(n)$  to the corresponding input port  $i$ , and configures the switch fabric using service matrix  $S$ ;
3. If some  $s_{i,j}(n) = 1$ , the input port  $i$  calculates the normalized queue length of all active VCs of  $Q_{i,j}$ , then select the cell of active VC with the longest normalized queue length to output.

In order to apply the LNQF algorithm to the multicast environment, we make the following improvements, and the improved algorithm is called multicast longest normalized queue first algorithm (MLNQF).

Assuming:

1. The input queues of  $N * N$  ATM switches adopt the multicast virtual output queue (MVOQ) technique, and the switch fabric is non-congestion;
2. Each input port only could schedule at most one unicast cell or multiple copies of the same multicast cell to output, and every output port could receive at most one input cell or cell copy;
- 3-6 are the same as the LNQF algorithms, except that  $Q_{i,j}$  indicates the MVOQ corresponded to output  $j$  at input port  $i$ , other symbols are also corresponded to MVOQ.
7. The cell representative matrix  $C(n)$ , whose row vector indicates the cell representative vector  $C_i(n)$  at the input port  $i$  in  $n$  time slot.  $C_i(n)$  is defined as:  $C_i(n) = (C_{i,1}(n), \dots, C_{i,N}(n))^T$ .  $C_{i,j}(n)$  indicates the queue-head cell representative of the active VC with the longest normalized queue length in all the active VCs of MVOQ  $_{i,j}$  at  $n$ th time slot.  $C_{i,j}(n) = 0$  indicates that MVOQ  $_{i,j}$  doesn't exist.

Under the above conditions, the MLNQF algorithm initially sets the input port from 1 to  $N$  with priority from 1 to  $N$  respectively (1 indicates the highest priority). Carry out the following operations in each time slot:

1. At the beginning of each time slot, change the priority alternately among all input ports, 1 turns to be  $N$ , and 2 turns to be 1, etc (except for the first time slot)

2. In a time slot, every input port  $i$  calculates the normalized queue length of its MVOQ, and produces the weight vector  $W_i(n)$  and get the cell representative vector  $C_i(n)$  of every input port  $i$  at the same time, then Send the  $W_i(n)$  and  $C_i(n)$  together to the scheduler ( in Figure 1 we omit  $C_i(n)$ )

3. The scheduler obtains a mapping between input port and output port, satisfying:

$$\arg \max [\sum_{ij} s_{i,j}(n) * w_{i,j}(n)] \text{ and } \sum_i s_{i,j}(n) = \sum_j s_{i,j}(n) = 1$$

If there isn't still mapping output port existing after obtaining the mapping in 3, the algorithm begins to implement the loop priority preserved policy. The scheduler produces  $N$  bits token, and the bit position corresponding to the non-mapping output port is 1. Then the scheduler begins to match these output ports with bit position of 1 from the input port with the current priority of 1, satisfying: if some bit of service vector in the current input port is 1, assuming its coordination  $(i,j)$  in service matrix  $S$ , then the corresponding element in the cell representative matrix is  $C_{i,j}(n)$ . Get the line number  $x$  of all the elements whose value is  $C_{i,j}(n)$  in the  $i$ th row of  $C$ , then match it with the token. If  $\text{token}(x) = 1$  is satisfied, set  $s_{i,x}(n) = 1$  and  $\text{token}(x) = 0$ .

Carry out this operation until all input ports are traveled or all output ports are matched. At last, the scheduler sends the service vector  $S_i(n)$  to the corresponding input port  $i$ , and configures the switch fabric using the service matrix  $S$ .

4. If some  $s_{i,j}(n) = 1$ , the input port  $i$  obtains the actual cell corresponding to  $(i,j)$  according to the cell representative matrix, and schedules it to output.

Analysis of MLNQF algorithms complexity: Calculation of the largest weight matching in Operation 3 is equal to the solution of network flow problem. Currently the most

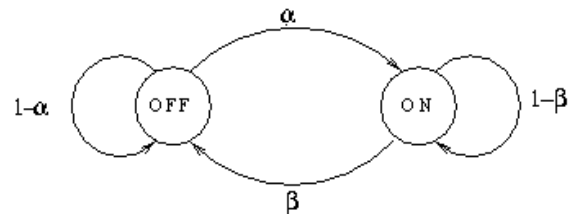


Fig. 2. ON-OFF Traffic Model

efficient algorithm to resolve this problem has the time complexity  $O(N^3 \log N)^{[5,6,7]}$ . Operation 4 Carries out the loop priority preserved policy, whose time complexity is  $O(N * \min(m, N-m))$ , and  $m$  is the number of matched input or output ports in Operation 3. Therefore the MLNQF algorithms complexity is still  $O(N^3 \log N)$  with the same as LNQF.

### 3.3 Simulation analysis of algorithms

We can discover that MLNQF not only keeps all the advantages of LNQF algorithm, but also improves the throughput and has QoS feature according to the characteristic of multicast cell and using loop priority preserved policy. To support this conclusion, we made the simulation analysis of MLNQF algorithm in  $4 * 4$  input-queued ATM switch supporting multicast.

Input Port	VC Sequence Number	Peak Rate P (Mbps)	Output Port			
			1	2	3	4
1	1	2	1			
	2	2	1	1	1	
	3	2		1	1	
	4	2		1		
2	1	1.5	1			
	2	1.5	1	1		1
	3	1.5			1	1
	4	1.5			1	
3	1	1	1	1		
	2	1				1
	3	1	1	1		1
	4	1			1	
4	1	0.5		1		1
	2	0.5	1	1		
	3	0.5			1	
	4	0.5	1			

Table 1 Parameters for simulations

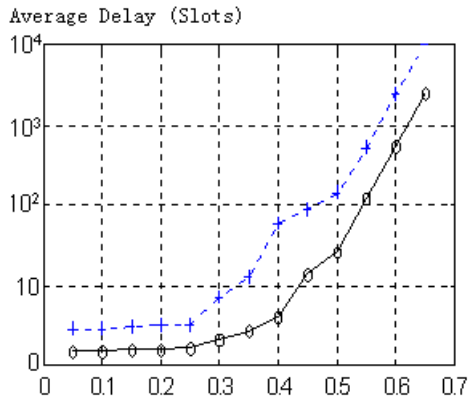


Fig. 3. Average Delay v.s.  $P_{on}$  for a 4x4 ATM Switch

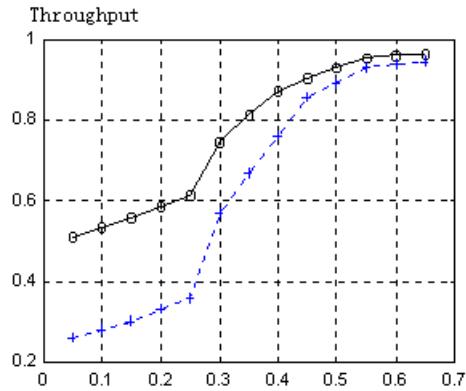


Fig. 4. Throughput v.s.  $P_{on}$  for a 4x4 ATM Switch

### 3.3.1 Source model

The actual network flow usually has the burst feature. In the discrete time zone, we generally use ON-OFF model to describe the data source with burst feature (see Fig. 2). In OFF state, the source doesn't send any cell; in ON state, source sends the cell at the rate of the cell with climax rate (P). When a time slot completes, the source can shift between OFF and ON states independently. The shifting probability from OFF state to ON state is A, and the shifting probability from ON state to OFF state is B. Thus, we usually use three parameters to describe ON-OFF source: the cell with climax rate (P), the average length of OFF state  $1/A$  and the average length of OFF state  $1/B$ . Therefore, the probability of ON-OFF source in ON state (transmit probability)  $P_{on}$  can be expressed by:  $P_{on} = A / (A+B)$ .

In our simulation, every input port has four active VCs; every VC source corresponds to ON-OFF model and multicast to the different output ports with homogeneous distribution (see Table 1). For example, the climax rate VC3 in input port is 2Mbps, and multicasts to output port 2 and 3.

### 3.3.2 Results of Simulation.

Fig. 3 and Fig. 4 provide the comparison of the delay and (calculated according to the utilization efficiency of output link 1) throughput quality between LNMF algorithm and MLNMF algorithm under different transmit rate  $P_{on}$ . We can discover that MLNMF algorithm makes a great improvement in both delay and throughput compared with LNMF algorithm. This is due to the adoption of MVOQ technique and consideration of the multicast feature in the cell scheduling (carrying out the 4th operation in MLNMF algorithm). Also, due to the use of MVOQ technique, the throughput using MLNMF algorithm could arrived over 93% when the transmit rate  $P_{on}$  is high. Therefore, we believe that the MLNMF can achieve the two aims, which are presented to the scheduling algorithms for multicast input queue: improving the throughput and owning some QoS features.

## REFERENCE

- [1] Duan,H. A high-performance OC-12/OC-48 queue design prototype for input-buffered ATM switch. In : Proceeding of IEEE INFOCOM'97, Kobe, Japan,1997,20-28
- [2] McKeown,N. Multicast scheduling for input-queued switches. IEEE JSAC, 1997,15(5):855-866
- [3] Andrews,M. Integrated scheduling of unicast and multicast traffic in an input-queued switch. In : Proceeding of IEEE INFOCOM'99, New York, USA,1999, 1144 -1151
- [4] Karol,M. Input versus output queueing on a space division switch. IEEE Trans. Communications,1988, 35(12):1347-1356
- [5] Mckeown, N. Achieving 100% throughput in an input-queued switches, In : Proceeding of IEEE INFOCOM'96, San Francisco, CA, USA, 1996,296-302
- [6] Li,S. Scheduling input-queued ATM switches with QoS features. In : Proceeding of IEEE ICCCN'98, Lafayette, Louisiana, USA, 1998, 107 -112
- [7] Hayes,J. Performance analysis of a multicast switch. IEEE Trans. Comm., April 1991,39(4):581-587
- [8] Chen, X. Performance comparison of two input access methods for a multicast switch. IEEE Trans. communications, May 1994 , 42(5):2174-2177