JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 17, 229-243 (2001)

Short Paper

Design and Analysis of an ATM Switch With Priority Discarding Scheme*

KUOCHEN WANG AND HSIN-JUNG WANG+
Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan 300, R.O.C.
+*Second Operation and Maintenance*
Pacific Cellular Corporation
Chungli, Taiwan 320, R.O.C.

In this paper, we propose an $N \times N$ high speed and non-blocking asynchronous transfer mode (ATM) switch with input and output buffers. In this switch, each buffer adopts a priority discarding scheme, which discards incoming cells of low-priority traffic when its queue length is greater than a predefined threshold value. Our switch also supports broadcast/multicast functions without increasing the cost and imposing a significant performance penalty. We use the discrete-time Markov chain model to analyze cell delay and cell loss probability for each traffic class. An example 4×4 ATM switch has been described with VHDL. We have verified the functionality of the switch via VHDL simulation, and have synthesized the switch to evaluate its area and timing. Experimental results and synthesis results show that our proposed ATM switch can meet a requirement for high speed and support QOS.

Keywords: ATM switch, multiple-bus, VHDL, QOS, high speed, priority discarding scheme

1. INTRODUCTION

The asynchronous transfer mode (ATM) promises to be the ultimate in on-premise internetworking technology. Its high-bandwidth can transfer graphics, audio, video, and text from application to application at much higher speed than now available [1]. Over the past few years, various ATM switch structures have been proposed [2-4]. There are three main switch structures: time division (shared medium) [5], multistage interconnection networks (MIN) [6, 7], and single stage [8]. Each structure has its own advantages and shortcomings, so we must perform a trade-off analysis in choosing the best switch structure. To support multicast operations in the Knockout switch, K. Y. Eng *et al.* [9] used a centralized multicast module to replicate cells and broadcast them over a

Received August 29, 1998; revised September 30 & December 2, 1999; accepted January 18, 2000.
Communicated by Nen-Fu Huang.
* This research was supported in part by the National Science Council, R.O.C., under Grant NSC85-2213-E-009-120.

fully-connected network. This results in a high cost due to the centralized processing and the limited number of multicast cells. Cell sequence may not be preserved if the traffic is mixed with multicast and single cast cells. In order to have better performance for ATM switches, we have to resolve the blocking conditions and reduce the probability of cell loss. A feasible, straightforward solution is to use buffering strategies at input ports [10], output ports [8, 11, 12], within the switch fabric [13, 14], or a mixture of the three [15, 16].

Cells may be held in buffers before they are processed by the switch. If the traffic is heavy, the buffers may overflow resulting in cell loss. In this situation, some cells may be selected to be discarded, and make the other cells meet the QOS requirement using some kind of priority control. Several priority control schemes have been proposed. In the *push-out* scheme [17], cells go into a single buffer. If the buffer is full and a high priority cell arrives, one low priority cell is found in the buffer and is "pushed out" (discarded). This is a complex process that may take some time. In another scheme called *buffer separation* [17], a separate buffer is used for each priority traffic class. Then low priority cells can easily be found and discarded. However, cells will be guaranteed to be delivered in sequence only if they all have the same "priority" on a connection path. This is not always possible to assure in an ATM network connection. In the *partial buffer sharing* scheme [17], low priority cells will be accepted if the queue length is less than a predefined threshold value. This is easy and efficient to implement [17]. Kroner *et al.* [18] have analyzed and compared the performances of these schemes. They have concluded that the partial buffer sharing scheme is the best alternative. We adopt this for our switch since it provides good performance and its buffer management is simple.

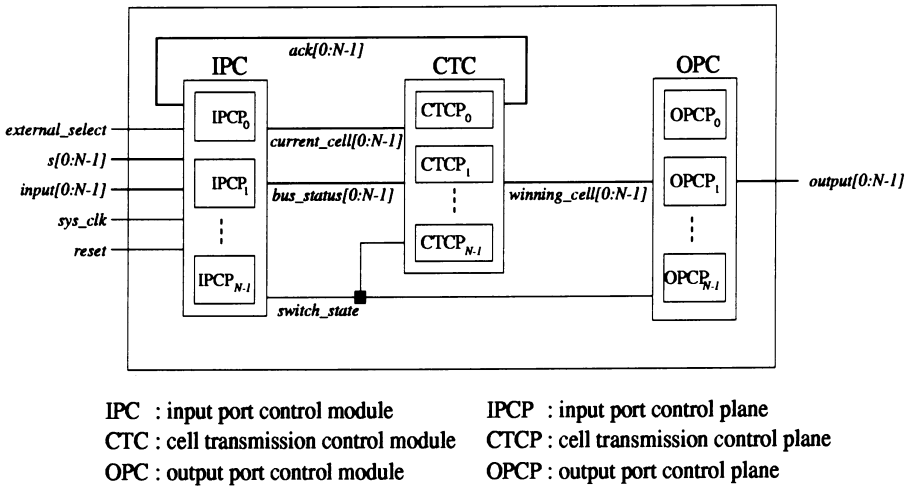
In this paper, we focus on high speed, high throughput, and a minimum cell loss rate to design a new generation of ATM switches. The switch is capable of broadcast/multicast as well as single cast. To meet these characteristics, our ATM switch architecture is based on a multiple-bus structure. All input ports can transfer cells to output ports in parallel. The bus structure routes cells from input ports to output ports directly. It just needs one stage to decide whether to transmit a cell or not, so it has a brief switching time. We use an input and output buffering strategy to reduce the cell loss probability. In each buffer, we adopt a priority discarding scheme according to the CLP bit in each cell header to meet the QOS for each traffic class. When the buffer length is greater than a predefine threshold value, the switch begins discarding arrival cells of low-priority traffic. We have described the proposed switch with VHDL [19-21] and its functionality has been verified using a VHDL simulator. The switch has also been synthesized to evaluate its area and timing.

The organization of this paper is as follows. The proposed switch architecture and the VHDL synthesis results are described in Section 2. The performance analysis of our proposed switch and some experimental results are shown in Section 3. Finally, some concluding remarks are addressed in Section 4.

2. PROPOSED SWITCH ARCHITECTURE

Fig. 1 illustrates the architecture of our proposed $N \times N$ ATM switch. Three are major components: *Input Port Control Module* (*IPC*), *Cell Transmission Control Module*

(*CTC*), and *Output Port Control Module* (*OPC*). The signals of the proposed switch are defined as follows:



IPC : input port control module
CTC : cell transmission control module
OPC : output port control module
IPCP : input port control plane
CTCP : cell transmission control plane
OPCP : output port control plane

Fig. 1. Architecture of the proposed $N \times N$ ATM switch.

- *external_select*: a 1-bit signal, which goes high to indicate that there is input traffic to be processed.
- *s*[0 : $N-1$]: a status bus for input ports 0 to $N-1$ to indicate whether or not an input has a cell to be transferred or not.
- *input*[0 : $N-1$]: input ports 0 to $N-1$.
- *sys_clk*: a global 1-bit clock signal for this switch.
- *reset*: a global 1-bit control signal that goes high for switch initialization.
- *current_cell*[0 : $N-1$]: indicates the first cell in each input buffer with respect to input ports 0 to $N-1$.
- *bus_status*[0 : $N-1$]: indicates validity of current cells, which are in the bus between IPC and CTC.
- *switch_state*: a three-bit value to represent the present state of the switch.
- *ack*[0 : $N-1$] : a set of acknowledgment signals with respect to input ports 0 to $N-1$, denoting which cells have won or lost the competition in this switch cycle.
- *winning_cell*[0 : $N-1$]: indicates the cells which have won the competition in CTC with respect to cell transmission control planes 0 to $N-1$.
- *output*[0 : $N-1$] : output ports 0 to $N-1$.

Each the signals: *input*, *current_cell*, *winning_cell* and *output* has N buses, and each bus has $424 + N$ bits. The function and detailed design of each module are described in the following subsections.

2.1 Input Port Control Module

The main tasks of the IPC module are buffering, cell header processing and VCI

translation, and single casting/multicasting. We construct the IPC module by using N input port control planes ($IPCP_i$), as shown in Fig. 1, where N is the number of input ports and i goes from 0 to $N-1$.

The configuration of an $IPCP_i$ is shown in Fig. 2. Each $IPCP_i$ contains a translation table to perform VCI translation and an input buffer to resolve the cell loss problem. At each input port, before storing incoming cells in the input buffer, VCI translation has to be performed first. After looking up the translation table, the cell's old VCI is translated into a new VCI and appended with an internal header that contains the local address.

The internal cell format is shown in Fig. 3, where $A_0 - A_{N-1}$ is the local address and N is the number of output ports. If any bit position in $A_0 - A_{N-1}$ is "1", a cell should be sent to the corresponding output port. For example, if the local address ($A_0A_1A_2A_3$) of a cell is "0101" in a 4×4 switch, the cell has to be sent to both outputs 1 and 3. By using this representation, the local address can easily specify the situations of multicast/broadcast, as well as single cast.

Note that there are two multicast addressing schemes [22]: *explicit addressing and cell address filters*. Neither scheme is scalable. We adopted the latter scheme so as to simplify the switch design. However, we can predefine a maximum switch size, e.g., 128, to resolve this scalability problem. In fact, the number of ports in commercially available switches is almost never greater than 128. For example, the IBM's 8285 N-ways ATM Workgroup Switch is a 12-port single-box ATM switch which can be expanded to 48 ports [23]. We focus our switch design on $N \leq 128$ ports (bits).

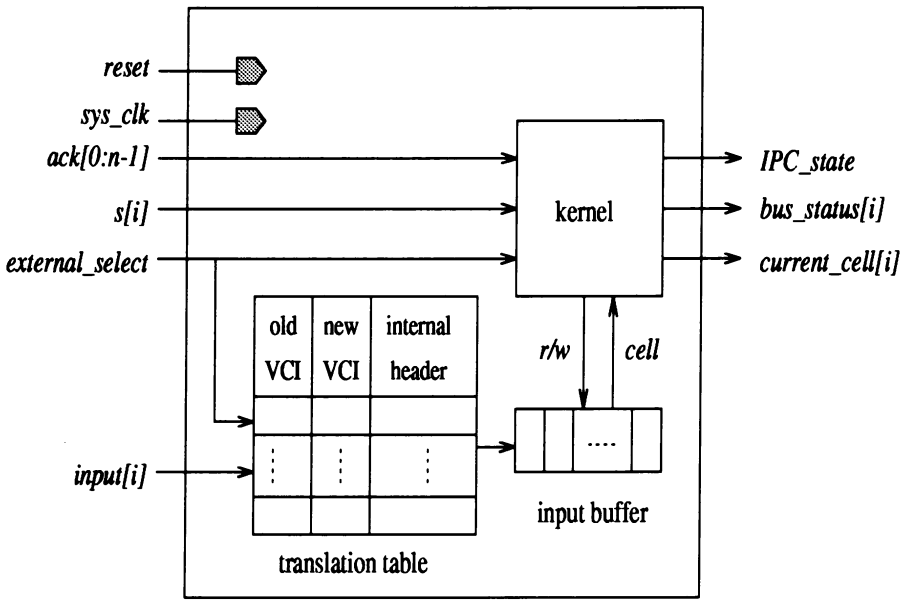


Fig. 2. Configuration of an $IPCP_i$.

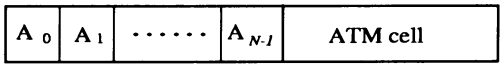


Fig. 3. Internal cell format.

After VCI translation, the cells will be stored in their respective input buffers. The size of each input buffer depends on the characteristic of its input traffic. We model each input buffer as a pseudo ring for storing cells, and adopt a priority discarding buffering strategy. There are two registers, *head_addr* and *tail_addr*, which indicate the front and the rear addresses of an input buffer. We also use a register to record the available space of an input buffer. When the number of cells in the input buffer reaches a threshold value, the buffer starts to reject low priority incoming cells. Using this priority discarding buffer strategy, we can reduce the cell loss rate of high priority cells. The head of line (HOL) cells are sent to CTC to compete with other cells which are destined for the same output port. From the acknowledgment signals (*ack*[0.. $N - 1$]), IPC knows which cells have won the competition. If a cell did not win the competition, IPC has to hold it in its input buffer and retransmit it in the next time slot.

2.2 Cell Transmission Control Module

The main tasks of CTC are arbitrating output contention and generating acknowledgment signals. As shown in Fig. 1, CTC is constructed with N cell transmission control planes ($CTCP_i$), where N is the number of output ports and $i = 0, 1,\ldots, N - 1$. Fig. 4 shows the configuration of $CTCP_i$. Each $CTCP_i$ contains three parts: *N cell filters*, a *competition network* and an *acknowledgment generator* (*ack_gen*). The N cell filters at $CTCP_i$ filter the cells off the broadcasting buses *current_cell*[0: $N - 1$]) from each IPCP if their ith local address bits are marked as '1.' To reduce the complexity of OPC, we have to limit simultaneously the number of cells which are destined for the same output. We use a competition network, which is a concentrator, to select a fixed number L of cells from N incoming cells to achieve an N to L concentration ($L < N$). The competition result will be transferred to the input port control planes via acknowledgment signals generated by the acknowledgment generator. The cells that won the competition will be allowed to transfer while the others will remain in IPCPs and wait for the next switch cycle. Note that when two cells which are of the same priority arrive at a 2×2 switching element of the competition network, the cell in a port with a lower index will win the competition. A barrel shifter can be added to the input ports of the competition network to resolve this unfairness.

2.3 Output Port Control Module

The last part of the proposed switch is the OPC module, which is constructed with N output port control planes ($OPCP_i$), where N is the number of output ports and $i = 0, 1,\ldots, N - 1$. OPC handles the winning cells coming from the competition network of CTC. The configuration of an $OPCP_i$ is shown in Fig.5. An $OPCP_i$ can be seen as a shared buffer which is composed of a shifter and L first-in first-out buffers. A shifter has L inputs and L outputs, and circularly shifts cells from inputs to outputs such that the L buffers are filled

in a cyclic fashion [24]. Note that an HOL cell is removed from one of the L buffers in a cyclic fashion and sent to the outgoing bus in each switch cycle. In this way, it allows complete sharing of the L FIFO buffers and provides the equivalence of a single queue, operating under the FIFO queuing discipline.

Since both the input and output buffers operate in FIFO mode, and there is only one path between each source-destination pair, the cell out-of-sequence problem will not occur.

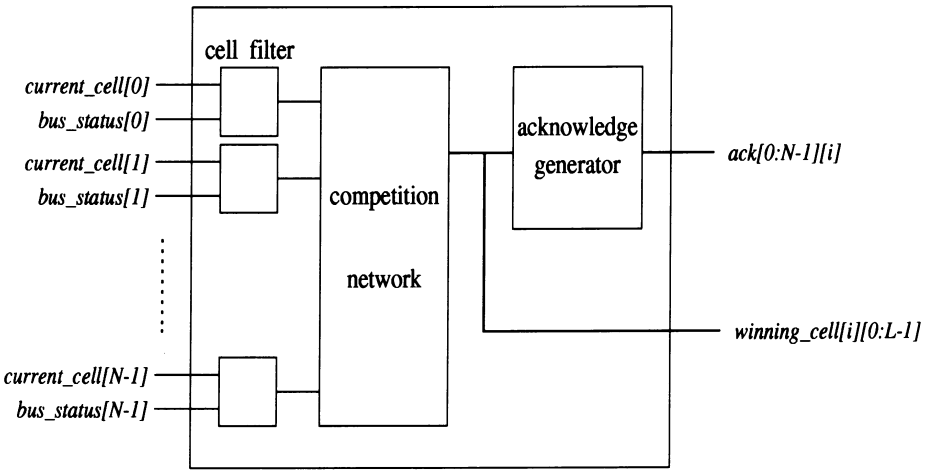


Fig. 4. Configuration of a $CTCP_i$.

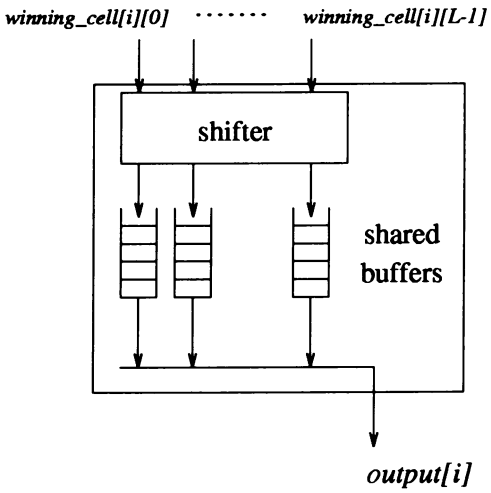


Fig. 5. Configuration of an $OPCP_i$.

2.4 The Switch as a Finite State Machine

We use a finite state machine to describe the operations of the proposed ATM switch. As shown in Fig. 6, the switch has two states. If the reset signal goes high, the switch will be in the *Reset* state. In this state, all registers will be set to an initial value and the switch will stay in this state until the reset signal goes low and the switch external control signal (*external_select*) activates (goes high). When *external_select* = 1 there are arrival cells in the input ports and the switch will be in the switching state (**Switching**) to start a switch cycle. Since a switch cycle contains four substates, and each substate takes one clock cycle, therefore a switch cycle takes four clock cycles. Each substate is described as follows:

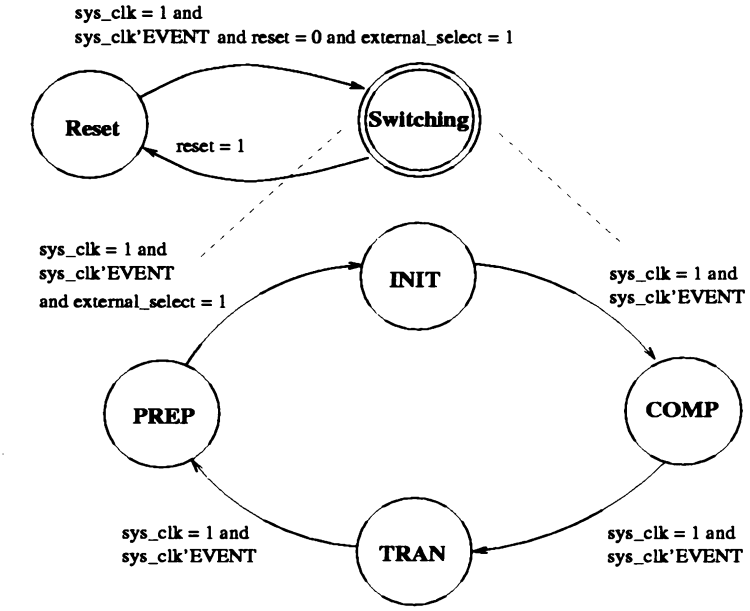


Fig. 6. State diagram of the proposed switch.

- **INIT** (initial) state: If there are arrival cells, they will be put into their input buffers after VCI translation and internal header padding. Then, IPC extracts the HOL cell from each input buffer in this switch cycle, or continues the last incomplete cell switching.

- **COMP** (competition) state: In order to reduce the complexity of OPC, we must limit the number of cells destined for the same output port in the same switch cycle. When too many input ports try to transfer cells to the same output port at the same time, output contention occurs. In this situation, all the input cells with the same destination address are arbitrated. In the proposed switch, CTC will handle this situation. Hence, at the beginning of the COMP state, IPC has to copy each cell and send it to CTC for

competition via the broadcasting buses (*current_cell*[0: $N-1$]). Then, it waits for the acknowledgment signals which contain the competition results from CTC. The cells which have won the competition will be sent to OPC via the buses *winning_cell*[0.. $N-1$].

- **TRAN** (transmission) state: In this state, OPC retrieves the winning cells form CTC and store them in output buffers. At the same time, IPC will get the acknowledgment signals from CTC and decide which cells need to be transferred in the next switch cycle. Since an input port may not win authorization to send the current cell to all destinations for broadcast/multicast traffic, we should check whether or not the current cell has been sent to all destinations. If an input port did not win a transmission authorization to all destinations, we have to record those destinations to which the current cell should be sent in the next switch cycle. We call this condition an incomplete transmission. In the proposed switch, we use a register (*address_reg*) to store the outstanding destination addresses, and a flag (*incomplete_flag*) to denote whether an input port completes transferring a cell. At the beginning of this state, we need to check the destination addresses of the current cell with the competition results from CTC. By using an *XOR* operation between the destination of the current cell and the competition result from CTC, we can find the outstanding destination addresses, which are stored in a register (*address_reg*) due to incomplete transmission.

- **PREP** (preparation) state: In this state, if *incomplete_flag* is enabled, IPC will replace the local address with the value of *address_reg* for each incomplete transmission cell, and then wait for the next switch cycle. In the output part, OPC extracts HOL cells from the shared buffers and sends them to the outgoing bus (*output*[0.. $N-1$]).

2.5 VHDL Synthesis Results

We use a VHDL synthesis tool and the *CCL08μm* library as the technology library to generate a gate level representation of an example 4×4 ATM switch. The *CCL08μm* library is a 0.8 micron CMOS library provided by the CCL of Industrial Technology Research Institute. The area of the example switch is 693,792 area units. Each state takes one clock cycle, and the clock cycle time has to be longer than the maximum delay of each state. In this example, the maximum delay occurs in CTC, and is 15.1 ns. Under this delay, the switch can support 6,950 Mb/s for each I/O link.

3. PERFORMANCE ANALYSIS

In this section, we evaluate the performance of the proposed switch in terms of cell loss probability for each traffic class, cell delay, and throughput.

3.1 Performance Analysis

Our switch is an $N \times N$ non-blocking switch fabric with finite input buffers and finite shared output buffers. There are back-pressure signals (*ack_gen*) between IPC and

CTC. Since the number of outputs in the competition network is L, there are at most L arrival cells at a given output port in the same time slot. Each cell arrives at each input port independently; we use λ to denote the cell arrival rate per time slot. Assume that a high-priority cell arrives with probability r and a low-priority cell with probability $(1 - r)$. Thus, the arrival rates of high- and low-priority cells are

$$\lambda_H = r\lambda \text{ and } \lambda_L = (1 - r)\lambda \tag{1}$$

respectively. We also assume that the arrival rate at each input port is identical, and destinations of input cells are uniformly distributed among all outputs. Each input buffer has a finite size S_I and a threshold value T_I. When an input cell arrives, it is accepted if the input queue length is less than the threshold T_I. However, a low-priority cell is discarded when the input queue length is greater than or equal to T_I. When an input buffer is full, the corresponding input arrival cells will be discarded. This priority discarding scheme is also applied to each output buffer with buffer size S_O and threshold value T_O. The performance expressions of our switch are summarized as follows:

$$TH = 1 - p_{0,O}P_L(0, 0) \tag{2}$$

$$DL = \frac{\sum_{i=0}^{S_I} ip_{i,I}}{\lambda^{HOL}(1-\beta)} + \frac{\sum_{i=0}^{S_O} ip_{i,O}}{1 - p_{0,O}P_L(0,0)} \tag{3}$$

$$\text{CL} = 1 - (1 - \alpha)(1 - \delta) = \alpha + \delta - \alpha\delta \tag{4}$$

where TH, DL, and CL denote throughput, cell delay, and cell loss probability of the entire switch, respectively. λ^{HOL} denotes the total arrival rate from the HOL of each input. $P_L(K_H, K_L)$ is the probability of K_H high- and K_L low-priority cells arriving at the competition network and destined for a particular output port. β is the cell rejection probability of an input port. $p_{i,I} = \Pr\{i \text{ cells in an input buffer}\}$, $p_{i,O} = \Pr\{i \text{ cells in an output buffer}\}$ represent the steady-state probability of an input buffer and an output buffer, respectively. α, δ denote the total cell loss probability of the input buffer and the output buffer, respectively.

3.2 Experimental Results

In this subsection, we present some experimental results of our switch and evaluate the effectiveness of the priority discarding scheme. In the following analysis, we let $N = 64$, $L = 6$, $S_I = 8$ and $T_I = 6$, and assume the system cell loss probability equals to the output cell loss probability. The cell loss probabilities of high- and low-priority cells versus total offered load are shown in Fig. 7. If the cell loss probability of the high-priority traffic is equal to 10^{-9}, the total offered load without the priority discarding scheme is about 0.66. However, if the priority discarding scheme is used, the total offered load can be increased to 0.73 and 0.8 for $r = 0.7$ and 0.5, respectively. Note that when the total offered load is 0.4, the cell loss probability of high-priority cells is very low (10^{-21} for $r = 0.5$). Fig. 8 shows the cell loss probabilities as a function of λ with various values of T_O. Note that the difference between the cell loss probabilities of these two priority classes

two priority classes decreases as T_O increases. We can use this result to determine the threshold value of T_O when the cell loss probabilities of the two priority classes are known. Fig. 9 shows the cell loss probabilities versus the output buffer size S_O for various values of r and $T_O = \frac{3}{4} S_O$ and $\lambda = 0.8$. With the same high-priority cell loss probability, a switch with the priority discarding scheme requires smaller buffer size than a switch without the scheme. Table 1 shows the minimum required output buffer size for different high-priority cell loss probabilities with $\lambda = 0.8$. To achieve a cell loss probability of 10^{-9} for high-priority traffic, the required output buffer size is 43 for the case without priority control and 24 for the case with priority control and $r = 0.5$. In this case, the required buffer size is reduced by 44%. Thus, there is great savings in buffer size when the priority discarding scheme is used. It can save more with either a lower value of r or a higher cell loss probability.

Table 1. Output buffer requirements for different high-priority cell loss probabilities.

Cell loss probability	$r = 0.3$	$r = 0.5$	$r = 0.8$	no control
1.0×10^{-6}	12	16	22	28
1.0×10^{-8}	16	22	31	38
1.0×10^{-9}	18	24	34	43
1.0×10^{-10}	22	28	39	49

Table 2. Maximum throughput versus *L*.

L	$r = 0.3$	$r = 0.5$	$r = 0.8$	no control
1	0. 635013	0. 635013	0.635013	0.635013
2	0. 898991	0. 899011	0.899101	0.899216
3	0. 964014	0. 964294	0.965723	0.969455
4	0. 972539	0. 972860	0.974479	0.979044
5	0. 973718	0.974043	0.975685	0.980365
6	0. 973871	0.974196	0.975841	0.980541
7	0. 973888	0.974213	0.975859	0.980561
8	0.973889	0.974215	0.97586	0.980563

Table 2 shows the maximum throughput of the proposed switch with respect to L. Note that the maximum throughput grows quite rapidly as L increases, and the maximum throughput of 97% can be achieved with $L = 4$. And, there is little gain in the maximum throughput when $L \geq 5$. In the latter case, with no cell loss constraints between high- and low-priority cells, the maximum throughput increases only slightly as r increases. But, if we set cell loss constraints for different priority traffic, we can get different results as shown in Table 3. Without the priority discarding scheme, the most stringent cell loss constraint must be satisfied by both classes of traffic. Thus, the maximum throughput of a switch with the priority discarding scheme is higher than that without the scheme. It also indicates that the maximum throughput increases as r decreases. From these results, we conclude that the priority discarding scheme is especially effective when the ratio of low-priority cells increases. Note in Table 2 that the maximum throughput of the input-buffer switch (the $L = 1$ case) is about 64%. This indicates that our switch can resolve the HOL problem and improve the throughput of the switch.

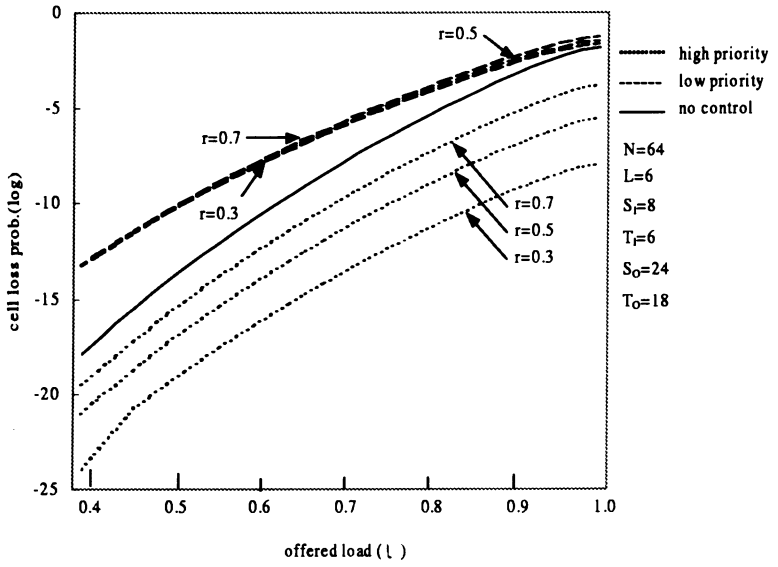


Fig. 7. Cell loss probabilities versus total offered load λ with various values of r.

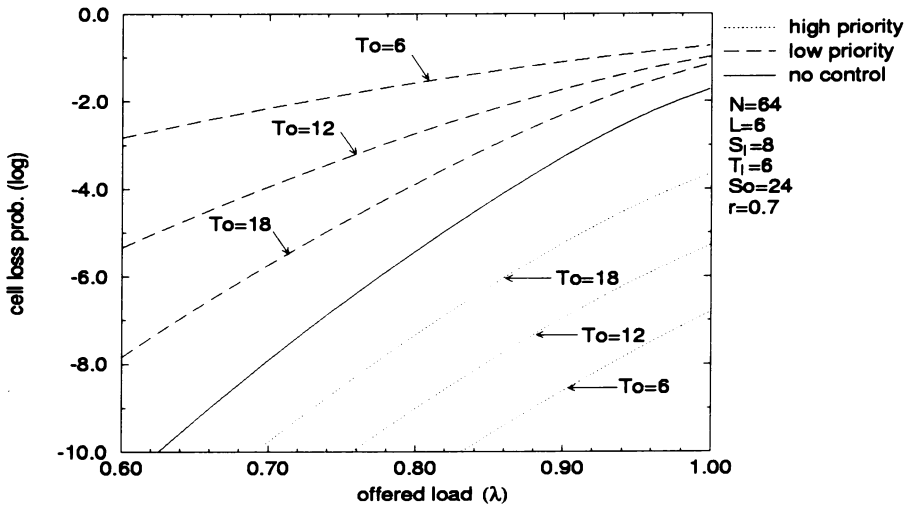


Fig. 8. Cell loss probabilities versus total offered load λ with various values of T_O.

Table 3. Maximum throughput versus L with high-priority cell loss probability 10^{-9} and low-priority cell loss probability 10^{-3}.

L	r = 0.3	r = 0.5	r = 0.8	no control
3	0. 8797	0. 8298	0. 7323	0. 6887
6	0.8729	0.7996	0.7035	0.6600

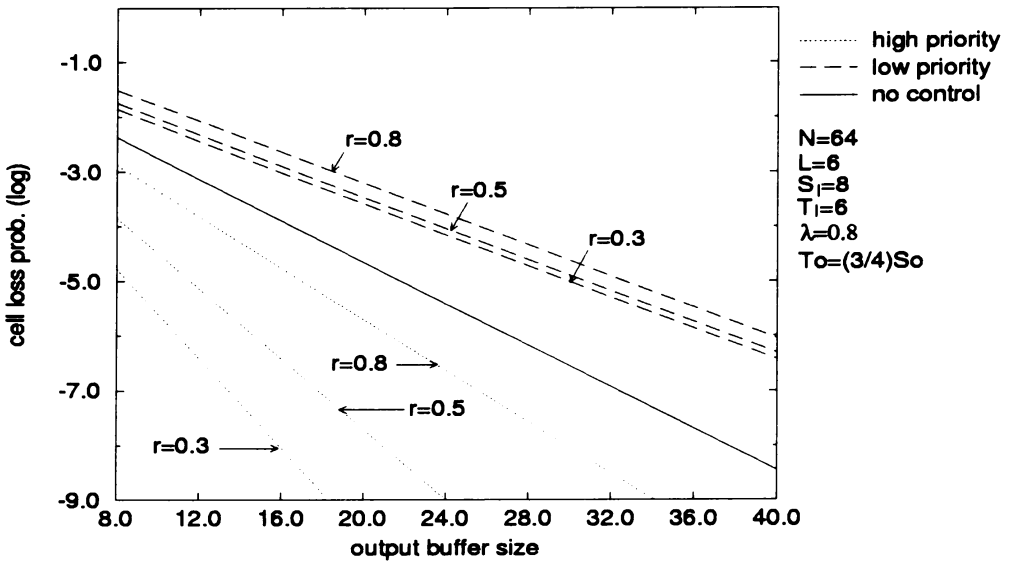


Fig. 9. Cell loss probabilities versus output buffer size S_O for various values of r.

Table 4 shows the cell delay in input/output buffers with various input loads; the loads of high- and low-priority traffic are the same (r = 0.5). From the table, the cell delay in input buffers is less than 0.01 of the time slot length, which corresponds to only 0.027μs when the line speed of the switch is 155 Mbps. Hence, we conclude that the fraction of delay due to input buffers is very small and has little impact on the overall delay of the switch. Morover, it reduces about 80% of (0.73828 vs. 3.74688) cell delay compared to the input-buffer switch (the L = 1 case) for λ = 0.6 and L = 6.

Table 4. Cell delay for N = 64, r = 0.5, S_I = 8, T_I = 6, S_O = 24, and T_O = 18.

λ	L = 6			L = 1
	Input buffers	Output buffers	system	system
0.60	0.00001	0.73822	0.73823	3.74688
0.70	0.00003	1.14819	1.14822	8.19693
0.90	0.00041	4.04769	4.04810	11.2208
0.99	0.00694	8.39032	8.39726	11.5700

In Table 5, we compare our switch with the Knockout switch. Basically, the throughputs and cell delays of both switches are quite similar. Under moderate load ($\lambda \leq$ 0.7), the cell loss probabilities of both high- and low-priority cells in our switch are much lower than those in the Knockout switch. Under heavy load (λ > 0.7), the cell loss probability of low-priority cells in our switch is slightly higher than that of the Knockout switch. However, the cell loss probability of high-priority cells in our switch is still lower than that of the Knockout switch.

Table 5. Comparison of proposed and Knockout switches for N = 64, r = 0.5, S_I = 8, T_I = 6, S_O = 24, and T_O = 18.

λ	Cell loss probability			Throughput		Cell delay	
	Proposed		Knockout	proposed	Knockout	proposed	Knockout
	high	low					
0.40	-21.0271	-13.2137	-6.35339	0.4	0.4	0.328122	0.328121
0.50	-17.2917	-10.3564	-5.80077	0.49999	0.49999	0.492173	0.492168
0.60	-14.1251	-7.94413	-5.35447	0.59999	0.59999	0.738231	0.738213
0.70	-11.3863	-5.85542	-4.98103	0.69999	0.69999	1.14823	1.14821
0.80	-8.99272	-4.03379	-4.59148	0.79999	0.79998	1.96274	1.96748
0.90	-6.94152	-2.49357	-3.23544	0.89816	0.89948	4.04810	4.29148
0.99	-5.50099	-1.47037	-1.82457	0.96899	0.97517	8.39727	11.0319

4. CONCLUSIONS

In this paper, we have presented a high speed ATM switch with input and output buffering which is based on a multiple-bus structure. A priority discarding scheme is adopted in each buffer which discards arrival cells of low-priority when its queue length is greater than a threshold value. We have analyzed the performance of the switch. The experimental results indicate that the priority discarding scheme is more effective when the ratio of high-priority traffic is small and the difference of cell loss probability constraints between the two traffic classes is large. The throughput of the switch with the priority discarding scheme, which satisfies the QOS of each traffic class, is much higher than that of the switch without the scheme. We have also shown that to satisfy the QOS of each traffic class, the buffer size required in a switch with the priority discarding scheme is smaller than that of a switch without the scheme. These results can also be used to determine the buffer size and the threshold value of the priority discarding scheme to satisfy the cell loss constraint of each traffic class.

In addition, the proposed switch also supports multicasting without adding extra cost. By using the broadcasting characteristic of the bus structure, the cell in any input port can be transmitted to each output port directly if it gets the transmission authority. This saves much time spent duplicating cells in comparison with other approaches which support multicasting. Simulation with VHDL has been performed to verify the functionality of the proposed ATM switch. Synthesis has also been conducted to evaluate the area and timing. The experimental and synthesis results indicate that our proposed ATM switch can meet a requirement for high speed and support QOS.

REFERENCES

1. R. Handel, M. N. Huber, and S. Schroder, *ATM Networks: Concepts, Protocols, Applications*, Addison-Wesley, 1994.
2. H. Ahmadi and W. E. Denzel, "A survey of modern high-performance switching techniques," *IEEE Journal Selection Areas on Communications*, Vol. 7, No. 7, 1989, pp. 1091-1103.
3. E. W. Zegura, "Architectures for ATM switching systems," *IEEE Communications*

Magazine, Vol. 31, No. 2, 1993, pp. 28-37.
4. R. Rooholamini, V. Cherkassky, and M. Garver, "Finding the right ATM switch for market," *Computer*, Vol. 27, No. 4, 1994, pp. 16-28.
5. C. Fayet, A. Jacques, and G. Pujolle, "High speed switching for ATM: the BSS," *Computer Networks and ISDN Systems*, Vol. 26, No. 9, 1994, pp. 1225-1234.
6. V. Benes, *Mathematical Theory of Connecting Networks*, Academic Press, New York, 1965.
7. C. L. Wu and T. Y. Feng, "On a class of multistage interconnection networks," *IEEE Transactions on Computers*, Vol. C-29, No. 8, 1980, pp. 696-702.
8. Y. S. Yeh, M. G. Hluchyj, and A. S. Acampora, "The knockout switch: a simple, modular architecture for high-performance packet switching," *IEEE Journal Selected Areas Communications*, Vol. 5, No. 8, 1987, pp. 1274-1283.
9. K. Y. Eng, M. G. Hluchyj, and Y. S. Yeh, "Multicast and broadcast services in a knockout packet switch," in *Proceedings of IEEE INFOCOM* '88, 1988, pp. 29-34.
10. G. Albertengo, "An optimal packet switch architecture for ATM," in *Proceedings of IEEE INFOCOM* '91, 1991, pp. 441-444.
11. B. Collier and H. Kim, "An output queuing batcher-banyan ATM switch architecture," in *Proceedings of IEEE MILCOM* '93, 1993, pp. 303-307.
12. K. Y. Kou, A. Arutaki, and S. Iwasaki, "The architecture and implementation of ATM switch for broadband ISDN," in *Proceedings of IEEE GLOBECOM* '93, 1993, pp. 8-13.
13. W. E. Denzel, A. P. J. Engbersen, and I. Iliadis, "A flexible shared-buffer switch for ATM at Gb/s rates," *Computer Networks and ISDN Systems*, Vol. 27, No. 4, 1995, pp. 611-624.
14. Y. S. Lin and C. B. Shung, "Queue management for shared buffer and shared multi-buffer ATM switches," in *Proceedings of IEEE INFOCOM* '96, 1996, pp. 687-695.
15. Y. Oie, M. Murata, K. Kubota, and M. Hideo, "Performance analysis of nonblocking packet switch with input and output buffers," *IEEE Transactions on Communications*, Vol. 40, No. 8, 1992, pp. 1294-1297.
16. H. Shi, C. Zukowski, and O. Wing, "VLSI design optimization of input/output-buffered broadband ATM switches," in *Proceedings of IEEE INFOCOM* '96, 1996, pp. 810-817.
17. W. Goralski, *ATM: The Future of High-Speed Networking*, Computer Technology Research Corp., 1994.
18. H. Kroner, G. Hebuterne, P. Boyer, and A. Gravey, "Priority management in ATM switching nodes," *IEEE Journal Selected Areas on Communications*, Vol. 9, No. 3, 1991, pp. 418-427.
19. D. L. Perry, *VHDL*, McGraw-Hill, 2nd Edition, 1993.
20. S. Mazor and P. Langstraut, *A Guide to VHDL*, Kluwer Academic Publishers, 1992.
21. J. Bhasker, *A VHDL Primer*, Prentice Hall PTR, 1994.
22. K. L. E. Law and A. Leon-Garcia, "Multicast and self-routing in ATM radix trees and banyan networks," in *Proceedings of IEEE INFOCOM* '95, 1995, pp. 951-959.
23. http://www.networking.ibm.com/82a/82aover.html, 1999.
24. J. J. Suh and C. H. Jun, "Performance analysis of the knockout switch with input buffers," *IEE Proceeding – Communications*, Vol. 141, No. 3, 1994, pp. 183-189.

Kuochen Wang (王國禎) received the B.S. degree in control engineering from National Chiao Tung University, Taiwan, in 1978, the M.S. and Ph.D. degrees in electrical engineering from the University of Arizona in 1986 and 1991, respectively. He is currently a Professor in the Department of Computer and Information Science, National Chiao Tung University. From 1980 to 1984, he worked on network management and on design and implementation of the Toll Trunk Information System at the Directorate General of Telecommunications in Taiwan. He served in the army as a second lieutenant communication platoon leader from 1978 to 1980. His research interests include computer networks, mobile computing and wireless Internet, fault-tolerant computing, and computer-aided VLSI design.

Hsin-Jung Wang (王欣嶸) received the B.S. degree in information science and engineering from Tatung Institute of Technology, Taiwan, in 1994 and the M.S. degree in computer and information science from National Chiao Tung University in 1996. His research interests include ATM networks, mobile computing, and telecommunication networks.