A Nonblocking Multi-stage ATM Switch using Cell-based Routing with a Hierarchical Cell Sorting Mechanism

David Santosöl, Seisho Yasukawa², Naoaki Yamanaka², Tetsuya Miki¹

¹ University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585, Japan
Tel: +81-424-90-7095
Fax: +81-424-90-7096
Email: {david-s, miki}@fifth.ee.uec.ac.jp

² NTT Network Service Systems Laboratories
3-9-1 Midori-cho, Musashino-shi, Tokyo 180-8585, Japan
Tel: +81-422-59-4769
Fax: +81-422-59-4819
E-mail: {Yasukawa.Seisho, Yamanaka.Naoaki}@nslab.ntt.co.jp

Abstract

A multi-stage switching architecture is a key technology for building a high-speed ATM switching system. An effective way to make a multi-stage switch nonblocking is to use cell-based routing. However, cell-based routing may cause cell-sequence disorder at the output of the switching fabric. This paper proposes a hierarchical cell-sorting (HCS) switch architecture, which is a nonblocking multi-stage ATM switch using cell-based routing technology. Each basic HCS switch performs cell sorting at every crosspoint, based on timestamp information in the cell-header. This arranges the cells in sequence at the output of each basic HCS switch, since the crosspoints are hierarchically interconnected from the input port to the output port of a basic HCS switch. A multi-stage HCS switch is constructed by interconnecting the input and output lines of these basic HCS switches in a hierarchical manner. Thus, the cell sequence in each final output of the multi-stage switch is preserved in a hierarchical manner. In this way, cell-based routing with 100% throughput is achieved, with no need for internal speed-up techniques.

1. Introduction

A multi-stage switching architecture is a feasible and cost-effective way to construct a high-speed ATM switching system. It is also the best architecture in terms of switch scalability because it can be easily expanded using additional switching blocks of the same type [1]. In a three-stage switching architecture, there are two types of virtual channel (VC) routing: connection-based routing and cell-based routing. The former routes cells of the same VC through the same path, and the latter routes cells of the same VC through all available paths. Cell-based routing is known to be an effective way to make a multi-stage switch nonblocking without using any internal speed-up techniques [2]-[5].

However, there are two main problems in a multi-stage switching system using cell-based routing: how to distribute input traffic to the second-stage switch fairly to keep the switch nonblocking and how to arrange the distributed cells in sequence at the output of the switching fabric in an effective way. In this paper we propose a dynamic cell distribution mechanism (CDM) in Section 2 to solve the first problem, and a hierarchical cell sorting (HCS) mechanism in Section 3 and a time-stamp distribution mechanism in Section 4 to solve the second problem. Finally, in Section 5 we describe the performance of this switch based on computer simulations.

2. Dynamic Cell Distribution Mechanism

In a cell-based routing architecture, cell (traffic) distribution over second-stage switches is a very important factor in keeping the switch nonblocking. For example, let us consider that the incoming cells...
Figure 1. Dynamic Cell Distribution Mechanism in three-stage HCS switch

Figure 2. Load distributions of multi-stage switches using connection and cell-based routing

(a) Connection-based routing and cell-based routing
(b) Random CDM
(c) Dynamic CDM

to different routes (different second-stage switches) every cell-time. Even if the route is randomly selected, there is no guarantee that in a given period of time each route is selected with the same frequency. Thus, the number of cells entering the second-stage switch buffers might become unbalanced and instantaneous load concentration might occur. If the load concentration continues to occur, it might cause blocking in a multi-stage switching system, which makes cell-based routing useless.

We propose a dynamic cell distribution mechanism (dynamic CDM) which considers the cell addresses when distributing cells at the first-stage switches. The main purpose of this mechanism is to balance the distributed instantaneous load in each second-stage switch buffer (Figure 1). The cell distribution mechanism is as follows. Each cell distributor (CD) placed at the input side of first-stage switches maintains a cell-distribution record of
which route (which second-stage switch) cells of the same group have been routed to during a given period of time. When a cell arrives, the cell distributor determines which group the cell belongs to (which third-stage switch the cell is going to). The switch then uses the cell-distribution record to arrange that the cells of the same group are fairly distributed to second-stage switches. This simple cell distribution mechanism enables cells to be fairly distributed in a dynamic way over second-stage switches, regardless of the type of input traffic.

To evaluate the load balancing performance of this mechanism, we performed three computer simulations of multi-stage switches using 1) connection-based routing, 2) cell-based routing with random CDM, and 3) cell-based routing with dynamic CDM. In the system using random CDM, cells are randomly distributed to different routes every cell-time, with no consideration of the cell address. Figure 2 shows the distributed load of a certain output buffer of second-stage switches when a particular traffic volume entered the switching system. With connection-based routing, the load was not balanced at all and blocking (average distributed load > 1.0) occurred at switches 1 and 3 of the second stage, even when the input traffic volume was below the total switching capacity. In the system using cell-based routing with dynamic CDM, blocking could be avoided because the load was fairly distributed over the second-stage switches. Thus, dynamic CDM makes a multi-stage switching system nonblocking for any type of input traffic pattern. The system using cell-based routing with random CDM is also a nonblocking system, since the average distributed load is below 1.0. However, from Figures 2(b) and 2(c) we can see that the dynamic CDM minimizes the variance of the distributed load of second-stage switches. This improves the delay performance of the proposed HCS switch.

3. Hierarchical Cell Sorting (HCS) Switch Architecture

To arrange the distributed cells in sequence at the output of the switching fabric, we propose a hierarchical cell sorting mechanism. The basic idea of this mechanism is to sort the distributed cells by arrival time in the switching system. In each basic switching block, we follow the FIFO (First In First Out) rule to determine which cell has to be sent first, to arrange the switched cells in sequence. By performing this cell sorting in each basic switch element of all stages, we can get the distributed cells in sequence at the output of the switching fabric.

To control the time-stamp and cell routing, we place a time-stamp controller (TR-CTRL) at the input side of the first-stage switches (Figure 1). Each controller has a counter that is synchronized with the cell transmission time; the counter is incremented by one each cell time. When a cell enters the switch, the value in the counter is written into the overhead of the cell as a time-stamp. This time-stamp records the time the cell was received by the switch and is used in cell sorting throughout the switching fabric.

Figure 3 shows the architecture of a basic HCS switch element. It performs cell switching and cell sorting at the same time. To avoid having to use internal speed-up techniques, we use crosspoint buffering in switching cells to an output port. Each crosspoint is constructed from an address filter (AF), a crosspoint buffer, a transit buffer, a time-stamp-based arbitration controller (CTRL), and a selector. A transit buffer can store several cells sent from either the upper crosspoint buffer or the upper transit buffer.

When the crosspoint buffer or transit buffer has at least one cell, it sends a transmission request (REQ) and time-stamp information to its controller. The controller then compares the time-stamps of the cells in the crosspoint and transit buffers for cell-selection and sends the result to the selector for cell-transfer. The transit buffer sends a NACK signal to the next-higher controller when it becomes full. When the upper controller receives this signal, it stops selecting cells to transmit. As long as no NACK signal is received from the next-lower transit buffer, cell-selection and cell-transfer are performed by the controller and the selector, respectively, once per cell time. The selected cell is sent to the next-lower transit buffer or to the output line at the lowest crosspoint.

The cell-selection is performed according to the time-stamp-based cell-selection rules given below. The controller uses these rules to determine which cell to send next. **Rule 1:** if both the crosspoint buffer and the transit buffer have a cell to send, then the cell with the earliest time-stamp is selected. If the time-stamp of the cell in the crosspoint buffer equals that of the cell in the transit buffer, the controller determines which cell to transmit by as follows [6]. Let us consider that crosspoint buffer $k$ and transit buffer $k$ ($k$ is the crosspoint number in a row) each have a cell with the same time-stamp. The cell in crosspoint buffer $k$ is selected with probability $I/k$, while the cell in transit buffer $k$ is selected with probability $(k-1)/k$ because the transit buffer $k$ is interconnected with $k-1$ upper
crosspoints, meaning that probability \((k-1)k\) represents cell transmission from \(k-1\) upper crosspoints. In this way, a cell is selected fairly when both cells have the same time-stamp. **Rule 2**: if neither the transit buffer nor the crosspoint buffer in a crosspoint has a cell to send, then the controller stops selecting cells to transmit, to avoid disrupting the cell sequence in the switching system.

By following these cell-selection rules, cells can be distributed in sequence across the switching system. However, **Rule 2** greatly increases the cell-transfer delay when the offered load is small, because in that case there is a greater chance of there being no cell from an input line. Therefore, we propose a time-stamp distribution mechanism to improve the cell-transfer delay performance of this switching system.

### 4. Time-stamp Distribution Mechanism

To reduce the cell-transfer delay, we introduce a time-stamp distribution mechanism. When no cell enters the first-stage switch during a cell time, the TR-CTRL at the input side of the first-stage switches creates a dummy cell, and the value of the counter is written into its overhead. The dummy cell is then passed to the following basic switches. The switches need the information carried by this dummy cell to maintain correct and effective time-stamp comparisons. A new cell arriving from the upper transit buffer or from an upper switch can overwrite a dummy cell waiting in a crosspoint buffer or transit buffer, because the new cell brings newer timestamp information. This prevents an excessive increase in the cell load.

As shown in Figure 4, when a cell enters a basic switch element, the incoming cell is copied and distributed to all output ports before its destination is checked. Then the AF in front of the crosspoint buffers checks the type of the incoming cell. If the cell type is REAL (meaning it is not a dummy cell) and the destination address corresponds to the AF number (which is equal to the output line number), the cell is written as a real cell in the following crosspoint buffer. If the cell type is REAL and the destination address does not correspond to the AF number, the cell is written as a dummy cell and given the same time-stamp as that of the incoming cell. If the cell type is DUMMY, the cell is written as a dummy-cell to the following crosspoint buffer.

As a consequence of using dummy cells, we need another rule for cell-selection to be successful. **Rule 3**: if both the crosspoint buffer and transit buffer have at least a dummy cell to send, then the cell with the earliest time-stamp is selected. If the time-stamp of the cell in the crosspoint buffer equals that of the cell in the transit buffer, the controller selects the non-dummy cell. If both are dummy cells, the controller determines which cell to send by using **Rule 2**. The introduction of dummy cells and this new cell-selection rule enables cell-selection to be
performed effectively regardless of the offered load, which dramatically improves the cell-transfer delay performance of the proposed switching system.

5. Performance of HCS Switch

We evaluated the performance of the HCS switch with the dynamic CDM and the time-stamp distribution mechanism by computer simulation. In our simulation, we modeled a three-stage HCS switch (switch size: $N^2 \times N^2$) composed of basic HCS switch (switch size: $N \times N$) elements arranged with close connection. The input traffic followed a Bernoulli process and the cell destination was given randomly from 1 to $N^2$.

Under these simulation conditions, we verified that cells were in sequence at the output of the basic HCS switches and also at the output of the three-stage HCS switch. This shows that the proposed hierarchical cell-sorting mechanism works well.

![Figure 4. Time-stamp distribution mechanism](image)

![Figure 5. Cell delay performance of HCS switch](image)

Figure 5 shows the average cell-transfer delay of a basic HCS switch and that of a three-stage HCS switch for $N=8$ and $N=12$. The results show that these switches can achieve 100% throughput, the ideal throughput performance of a single output-buffer-type switch [7]. They also show that a basic HCS switch has a good
delay performance due to the implementation of the timestamp distribution mechanism. The difference in delays between the switches with $N=8$ and switches with $N=12$ depended on the number of cell comparison steps in a basic HCS switch. In this simulation, we modeled a basic HCS switch architecture with $N/l$ comparisons. This explains why our basic HCS switch had at least $N/l$ extra delays compared to the conventional single output-buffer-type switch. Since the three-stage HCS switch was composed of basic HCS switch elements, it had at least three times as many extra delays as a basic HCS switch. In fact, comparing $n$ cells at a time can lower the number of cell comparison steps and also the delays to $N/n$.

6. Conclusions

We have proposed dynamic cell distribution, hierarchical cell sorting (HCS), and time-stamp distribution mechanisms to support a nonblocking multi-stage ATM switch that uses cell-based routing. The time-stamp distribution mechanism enables each basic HCS switch to perform effective cell sorting at its crosspoints in a hierarchical manner, which results in the distributed cells being arranged in sequence at its output. A three-stage HCS switch is composed of basic HCS switch elements, so the switched cells at each of its outputs are also in sequence. We verified by computer simulation that the HCS mechanism works well. The cell-transfer delay performance of this switch can be improved by reducing the number of cell comparison steps, if the hardware technology permits.

Computer simulations also showed that the dynamic cell distribution mechanism does a good job of distributing cells of any input traffic pattern over the second-stage switches. As a result, while the conventional output-buffer-type switch needs internal speed-up techniques to achieve 100% throughput, a multi-stage HCS ATM switch does not need any to achieve the same throughput performance. This makes the proposed multi-stage HCS switch an effective switching system that is applicable to future broadband ATM networks.

References