Modeling a Class of Priority-based ATM Communication Switch Designs

Jianxun Ding
Dept of Computer Science
Texas A&M University
College Station, TX 77843-3112

Hong Jiang
Dept of Computer Science & Engineering
University of Nebraska
Lincoln, NE 68588-0115

Abstract

Owing to its efficiency and flexibility, Asynchronous Transfer Mode (ATM) has emerged as the most promising transfer technique for wide-area broadband telecommunications that provide text, image, voice, and video information services. This paper deals with the modeling of a class of ATM communication switch designs. Various buffering mechanisms are considered in the switch design to alleviate cell contents. Analytical models for these design alternatives are developed for a case of a nonuniform traffic pattern, the favorite load, which can reflect some typical ATM applications. Performance comparisons are also conducted for a number of design alternatives.

1 Introduction

Asynchronous Transfer Mode (ATM), a cell based transmission, is the recommended transport technique for Broadband ISDN (BISDN) by CTTT [1]. Among ATM switch designs that have high transmission capabilities, the self-routing multistage interconnection networks (MINs), also known as the Banyan networks, are commonly considered one of the most effective network fabrics to build ATM switches in the high-speed transport systems supporting broadband services [1]. The very nature of BISDN suggests that a mixture of functionally different traffics be handled by the ATM switches, giving rise to priority-oriented control and non-uniform traffic patterns. In this paper, analytical models are developed to investigate such switch designs with the Banyan interconnection structure.

An \( N \times N \) Banyan network has \( \log_2 N \) stages, of which each has \( N/b \times b \times b \) crossbar switch elements (SEs). Conflicts of cells to an output are resolved by selecting one of the contending cells while buffering the rest. Depending on the way by which buffers are organized at the inlet side of SE, there can be either the conventional-buffered MINs (CMINs), where each inlet of an SE has one FIFO buffer, or split-buffered MINs (SMINs), where each input buffer is split into smaller ones, each associated with an outlet. While simple in its design and implementation, the former suffers from one major drawback: the cell at the head of a buffer will block all the succeeding ones destined to other possible idle outlets. The latter, also called as Crosspoint Buffered MIN [2], solves the problem by providing a buffer for each outlet. Depending on whether the split buffers are of fixed or variable sizes, SMINs can be further classified as either statically split-buffered MINs (SSMINs) or adaptively split-buffered MINs (ASMINs).

In papers [2, 3], the analytical model of SSMINs under the uniform input load was developed. However, as mentioned above, there are numerous services in BISDN. The traffic loads in the BISDN ATM switches are highly nonuniform. Hence we should not limit our analyses under the uniform traffic pattern for the ATM switch design. On the other hand, performance analyses cannot exhaust all nonuniform traffic patterns. Consequently, more and more researchers are focusing on typical nonuniform traffic patterns that can reflect communication applications more realistically than the uniform traffic patterns [4, 5, 6, 7].

In this paper, we study the performance of the priority-based SSMINs and ASMINs under the favorite load. The favorite load is a typical nonuniform traffic pattern and is more general than the uniform traffic pattern. It is one of the most common processor-memory communication patterns in the parallel processing environment [4]. It also can reflect some current and future ISDN applications. The paper is organized as follows. Section 2 describes the basic assumptions of SSMINs and ASMINs, the favorite load, and the priority-based selection strategies. Section 3 presents analyses based on the model developed in [8] for various priority-based SSMIN and ASMIN switch designs. Section 4 discusses the analytical results and shows the interesting performance behavior of the priority-based SSMINs and ASMINs. Finally section 5 concludes the paper.

2 General Assumptions and Features

The system under study assumes the following operational characteristics, as seen in most of the conventional-buffered MIN analyses.

(a) The network fabrics are operated in a time-slotted fashion, with fixed length cells progressing synchronously from stage to stage.

(b) Cells are generated at each network inlet with the same probability. If in a cycle a cell cannot be accepted by a first stage SE because the corresponding buffer is full, it will be rejected. In the next cycle, the arrival pattern of a cell will remain the same.

(c) Cells generated in a cycle are independent of the cells generated in the previous cycles.

(d) There is no blocking at the outlets of the network.
Under the uniform traffic pattern, cells are destined uniformly across all network output ports. In an \(N \times N\) multi-stage interconnection network, however, one network input is likely to send cells to a specific network output port most of the time [4]. If a network input port \(IN\) sends cells more often to the network output \(OUT\), \(IN\) will be called the favorite network input of \(OUT\). \(IN\), the favorite network output of \(IN\), and the cells from \(IN\) to \(OUT\), the favorite cells. If \(\lambda\) is the favorite rate, then a cell will be equally likely directed to any one of the \(N - 1\) non-favorite outputs with probability \((1 - \lambda)/(N - 1)\). This type of nonuniform traffic pattern is called the favorite load.

The favorite load may frequently appear in some BISDN applications where each network input sends a large amount of (image) data to its favorite network output and occasionally sends a small amount of (telephone) data to all other outputs. Note that in an \(N \times N\) network, the favorite load reduces to uniform load when cells from a network input \(IN\) go to its favorite network output with the favorite rate \(\lambda = 1/N\).

Since the switch fabric in question is permutation symmetric, we assume, without loss of generality, that the inlet \(IN\)‘s favorite outlet is \(OUT\), in a split-buffered Omega network as illustrated by the following example load matrix. It is easy to see that if the favorite cells always pass through the straight connections exclusively, a non-favorite cell sometimes, but not always, passes through the straight connections. In an \(SE\), a split buffer associated with the straight (cross) connection path is called the favorite (non-favorite) buffer. (see Fig. 1). Any cell passing through an \(SE\)‘s favorite buffer is called the favorite request to that \(SE\); the other cells passing through non-favorite buffers are called non-favorite requests to that \(SE\).

\[
\begin{bmatrix}
\frac{\lambda}{N-1} & \frac{\lambda}{N-1} & \cdots & \frac{\lambda}{N-1} \\
\frac{1-\lambda}{N-1} & \frac{1-\lambda}{N-1} & \cdots & \frac{1-\lambda}{N-1} \\
& \frac{1-\lambda}{N-1} & \frac{1-\lambda}{N-1} & \cdots \\
& \frac{1}{N-1} & \frac{1}{N-1} & \cdots & \frac{1}{N-1}
\end{bmatrix}
\]

### 3 The Analytical Models

As we mentioned in section 1, we will consider priority-based BISDN ATM switch designs. If there are more than one cells destined to the same \(SE\) outlet at the same time, priority can be assigned to the favorite or non-favorite buffers. As a result, the routing logic in an \(SE\) always selects cells from the split buffers with a higher priority. Also, since the buffer splitting need not to be even, depending on the traffic load and the service demands, the favorite buffers can adaptively have larger or smaller number of units than the non-favorite buffers. Fig. 1 shows the four possible combinations of the adaptive splitting with the priority-based selection policies.

We follow the notations defined in [8] and extend its model to the analyses of different selection policies. The selection policy can be random-based or priority-based: (1) If cells in the favorite or non-favorite buffers receive the equal priority, the selection policy is called the Equal Priority (EP) policy; (2) if cells in the favorite buffer receive higher priority than the non-favorite buffers, the selection policy is called the Prioritize Favorite buffers (PF) policy; and (3) if cells in the non-favorite buffers receive higher priority than the favorite buffer and conflicts among non-favorite buffers are solved randomly, the selection policy is called the Prioritize Non-favorite buffers (PNF) policy. In paper [8], the model for EP was developed. In the following subsections 3.1 and 3.2, we will derive analytical models for SSMINs with PF and PNF selection policies. Then in subsection 3.3, we will revise it to get the analytical model for ASMINs.

#### 3.1 PF Policy

Under the PF policy, the routing logic gives the favorite buffers higher priority than the non-favorite buffers. A cell in a favorite buffer is always selected whenever there is a conflict with the cells in the other \(b - 1\) dedicated non-favorite buffers. Hence, the probability that a cell is ready to a \((k + 1)\)th-stage \(SE\)‘s inlet due to a favorite request is equal to the probability that the preceding stage \(SE\)‘s favorite buffer is not empty, regardless of the statuses of the other non-favorite buffers. Therefore, the equation (2) in [8] about \(ed(k + 1)\) should be changed to:

\[
q(k + 1) \cdot ed(k + 1) = pf0(k).
\]

Hence,

\[
ed(k + 1) = \frac{pf0(k)}{q(k + 1)}. \tag{1}
\]

Similarly, under the PF policy, a cell in a favorite buffer need not to compete with cells in the other non-favorite buffers in order to be selected. However, if the favorite buffer is empty, a cell in a non-favorite buffer must compete with cells in other \(j (0 \leq j \leq b - 2)\) non-favorite buffers in order to be selected. So the corresponding equations of \(rf'(k)\) and \(rnf'(k)\) should be changed to

\[
rf'(k) = 1.0, \tag{2}
\]

\[
rf'(k) = pf0(k) \sum_{j=0}^{b-2} (j+1)(j+2)pmf0(k)^{j+2} - j \cdot \frac{1}{j+1} = \frac{pf0(k)}{(b-1)pmf0(k)} [1 - pmf0(k)^{b-1}]. \tag{3}
\]

#### 3.2 PNF Policy

Under the PNF policy, the routing logic gives non-favorite buffers higher priority than favorite buffers. A cell in a
favorite buffer cannot be selected unless all the \( b - 1 \) non-
favorite buffers dedicated to the same SE outlet are empty. Conflicts among cells in the non-favorite buffers are solved randomly. Hence, the probability that a cell is ready to a
\((k+1)\)th-stage SE's inlet due to a favorite request is equal to the probability that the preceding stage SE's favorite buffer is not empty AND all its \( b - 1 \) non-favorite buffers are empty. Therefore, the equations about \( ed(k+1) \) should be changed to:

\[
q(k + 1) \cdot ed(k + 1) = \frac{pf_0\(k\) \cdot pn\(f_0\)(k)}{q(k + 1)}. \tag{4}
\]

Similarly, under the PNF policy, a cell in a favorite buffer cannot be selected unless the corresponding \( b - 1 \) non-favorite buffers are all empty. A cell in a non-favorite buffer has higher priority to be selected than a cell in the favorite buffer; but it must compete with cells in other \( j \)  
\((0 \leq j \leq b - 2)\) non-favorite buffers. So the equations of \( rf'(k) \) and \( rnf'(k) \) should be changed to

\[
rf'(k) = \frac{pf_0\(k\) \cdot pn\(f_0\)(k)}{\sum_{j=0}^{b-1}(b-j)pn\(f_0\)(k)\cdot pn\(f_0\)(k)^{b-j-1}} = \frac{1}{(b-1)pn\(f_0\)(k)}[1 - pn\(f_0\)(k)^{b-1}]. \tag{6}
\]

### 3.3 Adaptive Splitting

One may naturally consider that under the favorite load, buffers should, instead of being equally split, be divided according to the ratio between the favorite cells and the non-favorite cells. We call this strategy adaptive splitting. Adaptive splitting can be either fixed for some specific applications, or dynamic for general applications. Tamir and Frazier reported an implementation of dynamic splitting in the UCLA ComCoBB chips [9]. In their implementation, they allowed packets to have varied lengths and used virtual cut-through technique to further increase the switch performance. The implementation and hardware cost issues are beyond the scope of this paper. Here we restrict our discussions on the analytical modeling for ATM packet (cells with fixed length) switching strategy. Our main purpose is to derive an analytical model to evaluate the performance behavior of the adaptive splitting in combination with the priority-based selection policies.

Recall that \( L(L \geq 3; L \leq 2) \) is meaningless for the adaptive splitting) represents the total number of buffer units associated with an SE's inlet, \( I_f \) represents the size of a favorite buffer, and \( lnf \) represents the size of a non-favorite buffer. There are many ways for the adaptive splitting depending on how to select the adaptive coefficient. We give one simple example here to show how to apply the analytical model. Let \( \delta \) be the adaptive coefficient.

\[
\delta = \frac{e(1)}{(1 - e(1))(b - 1)},
\]

\[
\text{set } \frac{I_f}{lnf} = C \cdot \delta,
\]

or

\[
\frac{I_f}{lnf} = C \frac{\delta}{\delta}, \tag{7}
\]

![Figure 2: Normalized Throughput versus Input Rate](image)

and, \( L = I_f + (b - 1) \times lnf \), \( C \) is an adjusting parameter and in this example it is selected as 1.0. For instance, in an ASMIN with 2 x 2 SEs, if the total buffer size in an SE inlet is 4 \((L = 4)\), and \( e(1) = 0.75\), then \( \delta = 1/3 \) and each SE's inlet will have a 3-unit favorite buffer and a 1-unit non-favorite buffer; each inlet will have a 1-unit favorite buffer and a 3-unit non-favorite buffer.

Changing the split buffer size \( l \) to \( I_f \) or \( lnf \) in the related equations, such as those of \( rf(k) \), \( rnf(k) \), \( pf_0\(k\) \), \( pn\(f_0\)(k) \), \( pf_1\(k\) \), \( pn\(f_1\)(k) \), \( Rf(k) \), \( Rnf(k) \), etc., we obtain the analytical model for the ASMINs.

### 4 Results and Comparisons

Our analytical model is validated through extensive simulations. Each simulation is performed until the confidence interval is smaller than 5% of the mean, using 95% confidence intervals. The analytical results generally match well with the simulation results. Note that most conventional-buffered models have the low accuracy under the heavy input load [10]. One of the major reasons for this inaccuracy stems from the re-distribution assumption — a blocked cell in an SE is uniformly re-distributed in the next cycle. In a split-buffer MIN, a blocked cell will stay in the "dedicated" split buffer and compete for the same outlet again in the next cycle. Hence this re-distribution assumption is no longer used in our model.

Fig. 2 compares the normalized throughput versus input rate for 256 x 256 sized SSMM and ASMM with the EP strategy. The favorite rate \( e(1) = 0.75 \) and the buffer size of an SE inlet is 4. We use 2 + 2 to represent the evenly split-buffered SSMM with each split buffer size \( l \) = 2. The figure shows that ASMM with the 2 + 2 splitting performs better than the SSMM with the 2 + 2 splitting. And the ASMIN with the 1 + 3 splitting has the lowest normalized throughput among the three splitting approaches. The results conform the intuition. The 3 + 1 splitting fits the favorite input load better than the 2 + 2 splitting because it can host more cells in the network. On the other hand, the 1 + 3 splitting is totally unsuitable for the favorite input load and thus has the lowest throughput.
Figures 3, 4, and 5 further depict the impact of the priority selection strategies. Fig. 3 shows the normalized throughput versus input rate for a 256 x 256 SSMIN with 2 x 2 switches. The favorite rate $f(1) = 0.75$ and the buffer splitting is 2 x 2. Three different priority-based selection strategies are compared. Overall, the PF strategy has the highest throughput. With the increase of the input load, the EP strategy has higher throughput than the PNF strategy under the heavy load. Fig. 4 and Fig. 5 further plot the normalized delay for the favorite and non-favorite cells under the same network scenario as in the Fig. 3. We can see that for the favorite cells, the PF has the shortest normalized delay since the PF strategy always selects the favorite requests first. However, the PF strategy ignores the non-favorite requests most of the time, causing intolerably long delays for non-favorite cells. This is shown in the Fig. 5. Both figures show that the EP strategy always has moderate performance between the two extreme strategies.

Fig. 6 depicts two scenarios of ASMINs: the 3 + 1 splitting and the 1 + 3 splitting. For the 3 + 1 adaptive splitting, the favorite buffers have 3 units and the non-favorite buffers have 1 unit. The 1 + 3 case splits buffers in the opposite way. We observe that the ASMINs only display small throughput differences for the three priority-based selection strategies under the heavy input load. But overall, the 3 + 1 splitting has higher throughput than the 1 + 3 splitting. As for the normalized delay, Fig. 7 and Fig. 8 show again that EP strategy always gives moderate performance between the two priority-based strategies, regardless how the buffers are split. The PF gives very small delay for the favorite cells in the 3 + 1 splitting but causes extremely large delays for the non-favorite cells. For the 1 + 3 splitting, because of the limited buffer space for the majority cells, a lot of cells have to be waiting outside of the network. So the normalized delay within the network seems very small. But the overall throughput for the 1 + 3 splitting is very low (see Fig. 6). The interesting result is that the PNF strategy does not cause extremely long delays compared with the PF strategy in both 3 + 1 and 1 + 3 cases. In fact, some applications needs this strategy: The small amount of exception/synchronization/telephone signals need to be handled in the higher priority than the large amount of normal/routine/image signals.

Overall, the PNF and EP strategies in both SSMINs and ASMINs do not cause extreme cell delays compared
with the PF strategy; the PF strategy in ASMINs can be used in some extreme situations under which favorite cells require very short delays in the expense of the extremely long delays of non-favorite cells. The adaptive splitting with more buffers for favorite cells generally performs better than the evenly splitting and the adaptively splitting with more buffers for non-favorite cells. It should be noted that all the performance gains are in the expense of hardware complexities. So the problem of choosing which combination of the cell selection and the buffer splitting strategy should be based on the QoS requirements of ATM cell transmissions. This paper only concerns the analytical modeling of the SSINs and ASINs. The hardware cost comparisons are beyond the scope of this paper and deserve another paper. Our ultimate goal is to develop a set of formal analytical models for various ATM switch designs and then find the most cost-effective switch fabrics.

5 Conclusions

It has become more and more evident that the topology of the switch fabric, the location of the cell buffers, and the contention resolution mechanism are the most significant aspects of an ATM switch design. We have developed an analytical model for a class of ATM designs with considerations in each of these three dimensions. In particular, we have analyzed the switch designs with static buffer splitting, adaptive buffer splitting, and the priority-based selection strategies for solving cell contentions. The analytical model has been developed for the favorite load traffic pattern. The favorite load is more general than, yet includes, the uniform load. It also reflects some typical ATM applications. With the help of this analytical model, we can investigate performance features such as network throughput and cell delay of favorite or non-favorite cells under different priority-based selection strategies. The analytical model has been validated through simulations and comparisons have been performed for a number of different priority-based ATM switch designs.

Acknowledgement

The authors wish to express their gratitude to Professor L. N. Bhuyan for his valuable comments and suggestions during the process of this study.

References