COST-PERFORMANCE TRADE-OFFS IN LARGE SCALE BANYAN-BASED ATM SWITCHES

by

ALIMUDDIN MOHAMMAD

B.E., Visvesvaraya College of Engineering, Bangalore, India, 1987
M.S., University of Petroleum and Minerals, Dhahran, Saudi Arabia, 1991

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

THE UNIVERSITY OF BRITISH COLUMBIA

March, 1998

© Alimuddin Mohammad, 1998
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-27203-6
Much research effort has been directed into the design and performance analysis of ATM switches to date. However, less work has been done in efficiently utilizing the switch resources (e.g. buffers and links) to achieve the required performance. The current techniques for designing and evaluating ATM switches focus mainly on achieving a specified global cell loss probability within a given maximum cell delay. While useful and necessary to ensure that the switch can provide an acceptable QoS, this approach cannot effectively measure other important qualities of an ATM switch such as resource utilization and implementation complexity. Further, as the use of the Internet increases and new multimedia applications emerge there is need for substantial improvement in capacity on Internet backbone links. The need for ever increasing packet delivery bandwidth makes it inevitable that switches will have to scale to handle larger aggregate bandwidth (quiet possibly in terabit-per-second) range and more importantly to maintain QoS delivery.

This thesis proposes a unifying framework for the design and analysis of large scale ATM switches based on enhanced banyan topologies. The banyan architecture has been chosen as the target architecture for developing the framework, since it is a logN depth network and is optimal in the use of switching elements. However, the standard banyan network suffers from internal blocking. In order to overcome blocking in the banyan network we have proposed the fat-banyan (FAB) architecture which employs gradual increase in dilation from the input stage to the output stage of the network. Gradually incrementing dilation has dramatic impact on the implementation and architectural scalability of the FAB switch. Further, the FAB switch is highly optimized in the utilization of the internal bandwidth by appropriate setting of the dilation parameter per stage. The FAB switch can scale to the terabit-per-sec range with reasonable increase in chip count. Also, there is minimal interaction between the different streams (VCs/VPs) traversing the switch, thus minimizing any effect on the QoS of a stream from other streams. The performance of the FAB is analyzed by analytical methods and by extensive computer simulations. It should be noted that the technique of optimizing
internal bandwidth can be applied in general to other ATM switching fabrics like the Benes network and the Fat-tree.

An important resource of an ATM switch is its buffers. The proliferation of best-effort services implies that ATM switches must contain large buffers to effectively manage congestion. Shared-memory switching uses smaller amount of buffering to achieve the same loss probability as that of dedicated buffering. However, shared memory architectures do not scale due to limitations of memory cycle time. A novel approach to scale shared-memory switches is proposed using small depth (truncated) self-routing fabrics. This approach provides an efficient realization of the growable switches concept which was proposed by other researchers. Three buffering schemes have been considered for the FAB switch: output buffering, shared output buffering and combined input with output buffering (dedicated as well as shared). The performance under these buffering schemes has been analyzed by extensive simulation. Pure output buffering is considered as it has been shown to provide optimal delay and throughput performance. Shared output buffering has been combined with the use of the memoryless truncated FAB network to design highly scalable shared-memory switch architectures. Input-output buffering with backpressure is shown to provide a cost-effective way of designing a lossless switching fabric.

This thesis also examines multicasting on the FAB switch. We have proposed a scheme to efficiently handle the replication of multicast packets in the FAB switch. It is shown that this technique can achieve very low cell loss even at very high offered loads and can also handle the case where the number of packet copies exceeds the switch size.

The various aspects of design and performance optimization developed in this thesis can serve as a general approach for designing and evaluating Terabit-per-second ATM switches.
# Table of Contents

Abstract ............................................................................................................................... ii

List of Figures ...................................................................................................................... vii

List of Tables ......................................................................................................................... xiii

Acknowledgment ................................................................................................................. xiv

Chapter 1 Introduction ........................................................................................................... 1
  1.1 ATM Switch Functionality ............................................................................................. 3
  1.2 Motivation and Scope of Thesis .................................................................................. 4
  1.3 Thesis contributions ...................................................................................................... 7
  1.4 Organization of the thesis ............................................................................................ 9

Chapter 2 ATM Switch Architectures ................................................................................... 11
  2.1 Introduction .................................................................................................................. 11
  2.2 Types of ATM Switches ............................................................................................... 11
    2.2.1 Shared-Memory Switches ...................................................................................... 11
    2.2.2 Shared-Medium Switches ...................................................................................... 14
    2.2.3 Space-Division Switches ....................................................................................... 15
  2.3 Banyan-Based Switch Architectures ............................................................................. 16
  2.4 Benes-based Networks ................................................................................................. 19
  2.5 Growable Switch .......................................................................................................... 21
  2.6 Drawbacks and Improvements Required ..................................................................... 23
  2.7 Conclusion .................................................................................................................. 24
<table>
<thead>
<tr>
<th>Chapter 3</th>
<th>Bandwidth Optimization in Banyan Switches</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>25</td>
</tr>
<tr>
<td>3.2</td>
<td>The Fat-Banyan Switching Fabric</td>
<td>27</td>
</tr>
<tr>
<td>3.3</td>
<td>Fat-Banyan Architectural Features</td>
<td>29</td>
</tr>
<tr>
<td>3.4</td>
<td>Chip Count Estimate for a Large Scale FAB Switch Implementation</td>
<td>36</td>
</tr>
<tr>
<td>3.5</td>
<td>Comparison of FAB Chipcount with other Switch Fabrics</td>
<td>40</td>
</tr>
<tr>
<td>3.6</td>
<td>Conclusion</td>
<td>44</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>Performance of The Fat-Banyan Switch</td>
<td>46</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>46</td>
</tr>
<tr>
<td>4.2</td>
<td>Performance Analysis under Independent Uniform Traffic</td>
<td>46</td>
</tr>
<tr>
<td>4.3</td>
<td>Cell Loss and Stage Dilation</td>
<td>49</td>
</tr>
<tr>
<td>4.4</td>
<td>Results</td>
<td>51</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Uniform Traffic</td>
<td>51</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Non-Uniform Traffic</td>
<td>55</td>
</tr>
<tr>
<td>4.5</td>
<td>Performance of the Output Buffered Fat-Banyan</td>
<td>61</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Uniform Traffic</td>
<td>61</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Bursty Traffic</td>
<td>64</td>
</tr>
<tr>
<td>4.6</td>
<td>The Input-Output Queued Fat-Banyan</td>
<td>67</td>
</tr>
<tr>
<td>4.6.1</td>
<td>Backpressure mode</td>
<td>67</td>
</tr>
<tr>
<td>4.6.2</td>
<td>Uniform traffic</td>
<td>69</td>
</tr>
<tr>
<td>4.6.3</td>
<td>Bursty traffic</td>
<td>69</td>
</tr>
<tr>
<td>4.7</td>
<td>Conclusions</td>
<td>72</td>
</tr>
</tbody>
</table>
Chapter 5
5.1 Introduction ........................................... 73
5.2 Considerations in Designing ATM Switches .......... 75
5.3 Truncated Switch Architectures ......................... 76
5.3.1 Generalized Truncated Switch Model ............... 77
5.3.2 Banyan Realization .................................. 78
5.4 Performance Analysis of the Truncated Fat-Banyan ... 82
5.4.1 Uniform Traffic ..................................... 82
5.4.2 Bursty Traffic ....................................... 82
5.4.3 Unbalanced Bursty Traffic .......................... 85
5.5 Truncated Fat-Banyan with Input Queues ............... 86
5.6 Conclusions ............................................ 88

Chapter 6
6.1 Introduction ............................................ 89
6.2 Multicasting in the FAB Switch ......................... 90
6.3 Multicast Performance .................................. 92
6.4 Conclusion ............................................. 94

Chapter 7
7.1 Conclusions ............................................ 97
7.2 Future research areas ................................... 99

Appendix A
List of standard acronyms ................................. 101

Bibliography ................................................ 102
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>ATM cell and Header formats at the UNI and NNI.</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>General Structure of an ATM switch.</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Shared-memory switch.</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>Shared-medium switch.</td>
<td>14</td>
</tr>
<tr>
<td>2.3</td>
<td>Banyan-based ATM switch architectures.</td>
<td>17</td>
</tr>
<tr>
<td>2.4</td>
<td>Benes network of size N=16.</td>
<td>21</td>
</tr>
<tr>
<td>2.5</td>
<td>The growable switch model</td>
<td>22</td>
</tr>
<tr>
<td>3.1</td>
<td>Two 16×16 banyans a) with 2×2 elements and b) with 4×4 elements.</td>
<td>26</td>
</tr>
<tr>
<td>3.2</td>
<td>(a)2×2 Buffered SE. (b)2×2 Fat SE with output concentrator. (3) 2×2 fully dilated Fat SE (no output concentrator required).</td>
<td>29</td>
</tr>
<tr>
<td>3.3</td>
<td>8×8 output-buffered FAB switch with dilation configuration as follows: DC=[2,3,4].</td>
<td>30</td>
</tr>
<tr>
<td>3.4</td>
<td>2 dilated 8×8 banyan network.</td>
<td>30</td>
</tr>
<tr>
<td>3.5</td>
<td>Generic package of BFSEs</td>
<td>31</td>
</tr>
<tr>
<td>3.6</td>
<td>Constructing 2×2 FSE's with L&lt;sub&gt;in&lt;/sub&gt; = 2 and L&lt;sub&gt;out&lt;/sub&gt; = 4 from BFSE's</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>(L&lt;sub&gt;in&lt;/sub&gt;: input link dilation, L&lt;sub&gt;out&lt;/sub&gt;: output link dilation)</td>
<td></td>
</tr>
<tr>
<td>3.7</td>
<td>Constructing 2×2 FSE's with L&lt;sub&gt;in&lt;/sub&gt; = 4 and L&lt;sub&gt;out&lt;/sub&gt; = 8 from BFSE's</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>(L&lt;sub&gt;in&lt;/sub&gt;: input link dilation, L&lt;sub&gt;out&lt;/sub&gt;: output link dilation)</td>
<td></td>
</tr>
<tr>
<td>3.8</td>
<td>Isomorphic representation of a banyan network</td>
<td>33</td>
</tr>
<tr>
<td>3.9</td>
<td>Board partitioning for a 8x8 FAB switch</td>
<td>33</td>
</tr>
<tr>
<td>3.10</td>
<td>Partitioning a BFSE among planes</td>
<td>37</td>
</tr>
<tr>
<td>3.11</td>
<td>BFSEs of varying sizes</td>
<td>38</td>
</tr>
<tr>
<td>3.12</td>
<td>The Knockout switch.</td>
<td>41</td>
</tr>
<tr>
<td>3.13</td>
<td>Building an N:L concentrator from smaller p:L concentrators.</td>
<td>41</td>
</tr>
</tbody>
</table>
Figure 4.1  Fat Switching Element (FSE)  ........................................... 48
Figure 4.2  Cell loss probability as a function of dilation for a FSE, \(S(8,2,L)\) for different input loads under uniform traffic (by analysis)  .................. 50
Figure 4.3  Maximum throughput for a fat-banyan switch with \(DC=[L_1=2, L_i=L_{i-1}+1, 2\leq i \leq \log N]\) and simple banyan network under the independent uniform traffic pattern with input load \(p=1.0\) (by analysis)  ........................................... 52
Figure 4.4  Maximum throughput for a fat-banyan switch with \(DC=[L_1=2, L_i=L_{i-1}+1, 2\leq i \leq \log N]\) under independent uniform traffic pattern (expanded scale) with input load \(p=1\) (by analysis)  ............... 52
Figure 4.5  Maximum throughput of a fat-banyan switch with different dilation configurations under independent uniform traffic with input load \(p=1\) (by analysis)  ........................................... 53
Figure 4.6  Concentrator action of a section of the FAB switch  ................. 54
Figure 4.7  Maximum throughput (by simulation) as a function of input load for a 8\(\times\)8 FAB switch with two different dilation configurations under independent uniform traffic  ........................................... 55
Figure 4.8  Cell loss probability (by simulation) as a function of input load for a 8\(\times\)8 FAB switch with two different dilation configurations under independent uniform traffic  ........................................... 55
Figure 4.9  Cell loss probability as a function of output dilation for a 16\(\times\)16 FAB switch for two different stage dilation configurations under independent uniform traffic with input load \(p=0.9\) (by simulation)  ....................... 56
Figure 4.10  An example of a maximum conflict pattern in a 16\(\times\)16 banyan network  ........................................... 56
Figure 4.11  Maximum throughput (by simulation) for a 8×8 FAB switch under mixed SSSD (point-to-point traffic) and background uniform traffic load of 1.0 for two different dilation configurations. . . . . . 58

Figure 4.12  Cell loss probability results (by simulation) for a 8×8 FAB switch under mixed SSSD (point-to-point traffic) and background uniform traffic load of 1.0 for two different dilation configurations. . . . . . 58

Figure 4.13  Routing of a Maximum Conflict Pattern in an 8×8 FAB switch with the following dilation configuration: DC=[2,3,4] . . . . . . . . 59

Figure 4.14  Cell loss probability for a 64×64 FAB switch with
DC=[2,4,8,8,8,8] as a function of output buffer size under independent uniform traffic with input load \( p=0.9 \) . . . . . . . . . 63

Figure 4.15  Cell loss probability (concentrator loss+buffer loss) for three different dilation configurations of a 16×16 FAB switch as a function of output buffer size under independent uniform traffic with input load \( p=0.9 \) (by simulation). . . . . . . . . . . . . . 64

Figure 4.16  Cell loss probability (concentrator loss+buffer loss) for two different dilation configurations of a 64×64 FAB switch as a function of output buffer size under bursty traffic conditions with \( p=0.7 \) and average burst length=15 (by simulation). . . . . . . . 65

Figure 4.17  Cell loss probability (concentrator loss+buffer loss) for two different dilation configurations of a 64×64 FAB switch as a function of output buffer size under bursty traffic conditions with \( p=0.9 \) and average burst length=15 (by simulation). . . . . . . . 66

Figure 4.18  Concept of LC backpressure in input-output queued fat-banyan switch with \( N=8 \) and dilation of last stage FSE's = 4. . . . . . . . . 68
Figure 4.19  Cell loss probability for FAB switches of size 32 and 64 as a function of input buffer size per output port under random uniform traffic with offered load $\rho=0.7$ and output buffer size=32 cells (by simulation). For $N=32$, DC= [2,4,4,4,4]. For $N=64$ DC=[2,4,4,4,4,4]. ........................................ 69

Figure 4.20  Cell loss probability for FAB switches of size 32 and 64 as a function of input buffer size per output port under random uniform traffic with offered load $\rho=0.9$ and output buffer size=32 cells (by simulation). For $N=32$, DC= [2,4,4,4,4]. For $N=64$ DC=[2,4,4,4,4,4]. ........................................ 70

Figure 4.21  Cell loss probability for FAB switches of size 32 and 64 as a function of input buffer size per input port under bursty uniform traffic with average burst length=15 and offered load $\rho=0.7$ (by simulation) and $B_{out}=256$. For $N=32$, DC= [2,4,4,4,4]. For $N=64$ DC=[2,4,4,4,4,4]. ........................................ 70

Figure 4.22  Average delay versus offered load ($\rho$) for FAB switch of size 32 under bursty uniform traffic with average burst length=15.
DC=[2,4,4,4,4,4], $B_{out}=256$ (by simulation). ............... 71

Figure 5.1  The general truncated switch model .................. 77

Figure 5.2  The growable switch model .............................. 79

Figure 5.3  Truncated fat-banyan (TFAB(8,2)) switch with output shared memory modules and DC=[2,3]. ......................... 81

Figure 5.4  Cell loss probability as a function of output buffer size for a $16\times16$ TFAB switch for different dilation configurations under independent uniform traffic with input load $\rho=0.9$ (obtained by simulation). ... 83
Figure 5.5 Cell loss probability (buffer loss) for two $16 \times 16$ TFAB switches with no concentrator loss as a function of buffer size per output port under bursty traffic with average burst length=15 and offered load $\rho=0.5$ (by simulation). ........................................... 84

Figure 5.6 Cell loss probability (buffer loss) for two $16 \times 16$ TFAB switches with no concentrator loss as a function of buffer size per output port under bursty traffic with average burst length=15 and offered load $\rho=0.7$ (by simulation). ........................................... 84

Figure 5.7 Cell loss probability (buffer loss) for two $16 \times 16$ TFAB switches with no concentrator loss as a function of buffer size per output port under bursty traffic with average burst length=15 and offered load $\rho=0.9$ (by simulation). ........................................... 85

Figure 5.8 Cell loss probability (buffer loss) for a $16 \times 16$ TFAB switch for two output group sizes as a function of buffer size per output port under unbalanced bursty traffic with offered load $\rho=0.71$ and average burst length=15 (by simulation). ........................................... 86

Figure 5.9 Concept of LC backpressure in input-output queued TFAB switch. ........................................... 87

Figure 5.10 Cell loss probability for $64 \times 64$ TFAB switch with input queueing and LC backpressure under bursty uniform traffic with average burst length=15 and offered load $\rho=0.7$ (by simulation). Bout=256. ........................................... 87

Figure 6.1 A multicast packet switch consisting of a copy network and a point-to-point switch. ........................................... 89

Figure 6.2 Balanced tree approach to copy number division per stage. . . . 91
Figure 6.3  Output interval assignment approach with examination of two
bits at each stage. .......................................................... 93

Figure 6.4  Simulation results for a 128×128 copy network: Cell loss
probability versus Output offered load for different dilation
configurations with mean fanout =2. ............................... 95

Figure 6.5  Simulation results for a 128×128 copy network: Cell loss
probability versus Output offered load under overload conditions
for different dilation configurations with mean fanout =2. ....... 95
List of Tables

Table 1.1  Types of ATM services ........................................... 2
Table 1.2  Terabit Experimental Switch Implementations ............... 6
Table 3.1  Comparison of interconnections required for FAB and dilated banyan networks. .................................................. 35
Table 3.2  Comparison of FAB and dilated banyan networks built with 2 dilated $2 \times 2$ FSE's. ....................................................... 35
Table 3.3  Chip count for various FAB switch sizes with external link rate of 622 Mb/s. .......................................................... 39
Table 3.4  Chip count for various FAB switch sizes with external link rate of 2.5 Gb/s. ........................................................... 39
Table 3.5  Chip count for various FAB switch sizes with external link rate of 10 Gb/s. ........................................................... 39
Table 3.6  Chip count for Knockout switch (as reported in ............. 42
Table 3.7  Chip count for a Banyan switch ............................... 42
Table 3.8  Chip count for the Tandem Banyan switch .................. 43
Table 3.9  Chip count for the Dilated banyan. ............................ 43
Table 3.10 Chip count for the FAB switch with C=128 and R=1.0 Gb/s. . . . 44
Table 3.11 Chip count for the FAB switch with C=240 and R=2.5 Gb/s. . . . 44
Table 4.1  Maximum throughput and cell loss probability results (by simulation) for a $8 \times 8$ FAB switch with dilation configuration as follows: DC=$[2,3,4]$ under hot-spot traffic. ...................... 60
Table 4.2  Maximum throughput and cell loss probability results (by simulation) for a $8 \times 8$ FAB switch with dilation configuration as follows: DC=$[2,4,6]$ under hot-spot traffic. ...................... 60
Acknowledgment

I would like to express sincere gratitude to Dr. H. M. Alnuweiri for his constant guidance and advice in supervising my Ph.D. research and to Dr. R. W. Donaldson for his interest and support in this project.

This research was partially supported by a grant from the Canadian Institute for Telecommunications Research under the NCE program of the Canadian Government, and by a Canadian Natural Sciences and Engineering Research Council grant.

Finally I would like to thank my beloved wife, Gazala for her support in my Ph.D. study.
Chapter 1 Introduction

Asynchronous transfer mode (ATM) has been adopted as the standard for Broadband Integrated Services Digital Network (B-ISDN) by CCITT. ATM is a fast packet switching and connection-oriented transfer mode technology based on statistical time division multiplexing. Emerging ATM/BISDN's will support four main classes of communication services as shown in Table 1.1 using different protocols for the ATM adaptation layer (AAL). ATM uses fixed length packets called cells, consisting of a user information field (48 bytes) and a header (5 bytes). The 48 bytes are used by AAL and the user payload. The ATM cell is configured slightly differently for the UNI (User Network Interface) and the NNI (Network Network Interface) as shown in Figure 1.1. At the UNI interface a GFC (Generic Flow Control) field is defined since various OAM (Operation and Maintenance) functions operate at the UNI only. The cell header mainly consists of the virtual circuit labels of VPI and VCI. At the UNI 8 bits are assigned to VPI and 16 bits assigned to VCI. For the NNI, 12 bits are assigned to the VPI and 16 bits are assigned to the VCI. The PT (payload type) field identifies the type of traffic residing in the cell. The cell may contain user traffic or management/control traffic. The C (cell loss priority) field is a 1–bit value. If C is set to 1, the cell may be discarded by the network during congestion. A C bit value of 0 indicates a higher priority cell and is so treated during congestion. The HEC (header error control) field is an error check field, which can also correct a 1-bit error. It is calculated on the 5-octet ATM header, and not on the 48-octet user payload.

The key component of an ATM network is the ATM switching node, which performs the fast packet switching function. ATM is designed to support serial link rates from a few kilobits per second to tens of gigabits per second (T1/E1, T3, OC-3, OC-12, OC-48, OC-192) [1, 2, 3]. The major challenge in designing ATM switches is to be able to
Chapter 1. Introduction

Figure 1.1. ATM cell and Header formats at the UNI and NNI.

Table 1.1. Types of ATM services

<table>
<thead>
<tr>
<th>Class</th>
<th>A - Constant bit rate (CBR)</th>
<th>B - Variable bit rate (VBR)</th>
<th>C - Connection oriented data traffic</th>
<th>D - Connectionless data traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection mode</td>
<td>Connection oriented</td>
<td>Connection oriented</td>
<td>Connection oriented</td>
<td>Connectionless</td>
</tr>
<tr>
<td>Bit rate</td>
<td>Constant</td>
<td>Variable</td>
<td>Variable</td>
<td>Variable</td>
</tr>
<tr>
<td>Example</td>
<td>H.261 video</td>
<td>VBR voice and video</td>
<td>Fax</td>
<td>Electronic mail</td>
</tr>
<tr>
<td>ATM Adaption Layer</td>
<td>AAL 1</td>
<td>AAL 2</td>
<td>AAL 3 or 5</td>
<td>AAL 4 or 5</td>
</tr>
</tbody>
</table>
handle aggregate bit rates greater than 10 Gb/s and into the terabit-per-second range. Building efficient switch architectures and fast enough digital electronics to handle such high bit rates is a major challenge. Another problem that any ATM switching system must solve is that of buffering i.e., storing incoming ATM cells that cannot be delivered upon arrival until the resources blocking their delivery (e.g. internal bandwidth of the switch, queue length at the input/output) is freed. This thesis addresses the architectural issues of design and evaluation of large scale switching systems.

1.1. ATM Switch Functionality

The general structure of an ATM switch is shown in Figure 1.2. The ATM switch receives statistically multiplexed cell stream on its input links. Each cell in the stream is identified to belong to a virtual connection by its VPI/VCI value. The input controller processes the header and may typically perform some or all of the following functions: identify the type of cell (Signalling, OAM, data, etc.), add a routing tag for routing through the switch fabric to the appropriate output, add information from the connection table like connection priority, perform a VPI/VCI translation and control input buffers if they are provided. The control processor deals with functions like connection establishment, connection release and bandwidth allocation and management. The output controller performs cell processing like removing information (routing tag, connection priority, etc) added at the input controllers and output buffer management. In this thesis we are

Figure 1.2. General Structure of an ATM switch.
mainly concerned with the ATM switching fabric and the buffering strategy employed in
the switch. An ideal ATM switch fabric should have the following three characteristics:
1. Capability to route all packets from its input lines to the requested output lines
   without loss.
2. Capability to deliver the packets to the requested output lines with minimum transit
delay.
3. Capability to preserve the same order of the packets at the output, as their arrival
   order at the switch inputs.

Thus, the functionality required of an ATM switch fabric is quite simple, but the
challenge as mentioned earlier is to achieve these characteristics at high speeds and to
satisfy the Quality of Service (QoS) requirements of both real-time and non-realtime
traffic, which would traverse these switches in a B-ISDN environment. The details of
the various switch architectures that have been proposed in the literature are discussed
in chapter 2.

1.2. Motivation and Scope of Thesis

Much research effort has been directed into the design and performance analysis
of ATM switches to date [4, 5, 6, 7]. However, less work has been done in efficiently
utilizing the switch resources (e.g. buffers and links) to achieve the required performance.
The current techniques for designing and evaluating ATM switches focuses mainly on
achieving a specified global cell loss probability within a given maximum cell delay.
While useful and necessary to ensure that the switch can provide an acceptable QoS, this
approach cannot effectively measure other important qualities of an ATM switch such
as resource utilization and implementation complexity. In our work an important design
goal is the optimization of internal links and buffers of the switch. We show that the
optimization of links has a profound impact on the interconnect complexity of the switch.
Since ATM switches generally employ large buffers, especially since the introduction of
available bit rate (ABR) services, the optimization of buffers is important in order to reduce the memory cost component of the switch.

As the use of the Internet increases and new multimedia applications emerge there is need for substantial improvement in capacity on Internet backbone links. The Internet growth in the last several years has been phenomenal. Personal computers are approaching the performance of workstations and it is only a matter of time before 155Mbps to the desktop will be common place within the professional environment. All of these factors will drive the deployment of ATM but it is not known when the explosion will occur. Additionally, it can only be speculated as to the sufficient ATM switch size required to meet the end user’s demands. Will 32, 64 or 128 155Mbps ports be the ideal sized LAN based switch? What about the WAN switches? Present state-of-the-art commercial switching equipment is capable of operating at 2.5 Gb/s to 10 Gb/s with 50 Gb/s to 100 Gb/s equipment beginning to be available. Table 1.2 lists a few experimental switch implementations which reportedly can handle multi Gb/s links. However, the need for ever increasing packet delivery bandwidth makes it inevitable that switches will have to scale to handle larger aggregate bandwidth (say in the terabit-per-second) range and, more importantly to maintain QoS delivery. In this thesis an important design consideration is the modular growth of the switch to handle such high bandwidth with modest cost increase and without adversely impacting the QoS provided to all flows through the switch. In the following we consider few important factors that influence the design and performance of ATM switches.

Routing: The routing algorithm used in the switch fabric greatly influences the speed of switching. Since a full crossbar implementation is not feasible for modular growth, a multistage interconnection network seems to be the most feasible solution towards modular growth of the switch. Routing schemes within the switch fabric can be classified as centralized routing schemes, partially distributed routing schemes and
self-routing schemes. Centralized routing schemes do not scale with large port numbers or high link speeds. Partially distributed routing schemes are faster than centralized routing schemes but may not scale well with large port numbers or high link speeds. Self-routing schemes use truly distributed routing algorithms and are the ideal choice for routing within the switch fabric. In this thesis, only self-routing switching fabrics are considered, although they are combined sometimes with shared-memory switching.

Flow Segregation and QoS: Another important issue in the design of an ATM switch fabric is that of interaction of different traffic streams within the switch fabric. This phenomenon is called QoS coupling. QoS coupling leads to the interference of a connections cell loss, cell delay and cell delay variation by other connections traversing

<table>
<thead>
<tr>
<th>Switch Name and Topology</th>
<th>Link Rate (Gb/s)</th>
<th>Aggregate Rate (Gb/s)</th>
<th>Stage of Development</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thunder and Lightening (Crossbar)</td>
<td>10-80</td>
<td>160-640</td>
<td>4-port Prototype</td>
<td>ECL</td>
</tr>
<tr>
<td>WUGS (Washington Univ. Gigabit Switch) (Buffered Benes Network)</td>
<td>2.5</td>
<td>80</td>
<td>64 port Prototype</td>
<td>CMOS 1.2 Gb/s G-link (HP)</td>
</tr>
<tr>
<td>Tiny Tera (Crossbar)</td>
<td>10</td>
<td>320</td>
<td>32-port Prototype</td>
<td>0.25-micron CMOS 2 Gb/s serial-links (TI)</td>
</tr>
<tr>
<td>Rerouting Banyan (Overlapped Banyan)</td>
<td>10</td>
<td>1000</td>
<td>128 ports</td>
<td>GaAs MESFET 10 Gb/s serial-links</td>
</tr>
<tr>
<td>iPPoint/3D (illinois Pulsar-based Optical INTerconnect switch) (Crossbar)</td>
<td>0.155 - 2.5</td>
<td>10 - 40</td>
<td>4 ports</td>
<td>FPGA/ASIC</td>
</tr>
<tr>
<td>Growable Switch (Clos Network)</td>
<td>0.155 - 1.0</td>
<td>25</td>
<td>32 ports</td>
<td>CMOS</td>
</tr>
</tbody>
</table>
the switch. In the worst case a few highly bursty connections may severely affect the QoS of well behaved connections. This problem is mainly due to the introduction of ABR/UBR services which is not subject to connection admission control (CAC) and has the potential to cause QoS coupling within the fabric. An important design criterion is hence the minimization of QoS coupling within the fabric. Ideally a fabric with no QoS coupling would require a crossbar implementation with no buffers in the fabric. In this thesis we effectively eliminate QoS coupling in the switching fabric. In our switch models different flows do not impact each others cell delay or cell delay variation. It should be noted however that QoS coupling management has to be done at the periphery (either at the input or output buffers) of the switch. This can be handled by efficient buffer management [8, 9] and cell scheduling techniques which employ the Fair Queueing algorithm or one of its variations [10].

Multicasting: Applications like teleconferencing and videoconferencing require the setting up of multicast virtual circuits. Implementing efficient multicast channels through an ATM switch is a major challenge. There are generally two approaches to multicasting in an ATM switch. In the first approach, the ATM switch is preceded by a separate copy network which creates the required number of copies of a multicast cell. These copies are then input to the “unicast” ATM switch which routes them to the appropriate destinations. The second approach is to incorporate the multicasting function in the switch core itself. In this thesis we propose an efficient way of implementing multicasting on our proposed switch architectures.

1.3. Thesis contributions

Broadly speaking this thesis proposes several techniques for designing large-scale ATM switches (operating with Tb/s aggregate capacity) that also employ high-speed links (e.g. OC-48=2.5 Gb/s or OC-192=10 Gb/s). In the following we elaborate on several specific contributions of the thesis:
Chapter 1. Introduction

1. This thesis proposes several switch architectures which provide high utilization of the two main resources of the switch namely, the interconnect and the buffers. Interconnect and wiring complexity becomes a major bottleneck for large-scale implementations, where the switch fabric spans several boards and shelves. Therefore minimizing interconnect complexity is an important design issue. Memory is a significant component affecting both cost and internal delays. Switches with high-speed links (OC-48) may require very large cell buffers in the fabric to guarantee the QoS of cell streams passing through the fabric. Hence, optimization of memory (cell buffers) in ATM switches is another important design goal.

2. We provide analysis of the performance of the proposed fabrics with theoretical methods and computer simulations under independent and bursty traffic conditions, and under general traffic patterns such as non-uniform and hot-spot. The effect of various parameters which affect cell loss, such as internal link dilation, depth of the fabric, and shared memory size are considered in the performance analysis.

3. QoS (Quality of Service) coupling is an important performance measure of the fairness of a switching fabric. QoS coupling occurs when cells from different streams (VCs/VPs) affect each others cell loss and cell delay. The architectures proposed in this thesis do not use internal buffers thus eliminating QoS coupling in the form of delay effects between different streams. Instead, they strategically employ internal link dilation to reduce cell loss.

4. We generalize our switch design strategies to provide an efficient method to develop scalable shared memory switches. This is based on the use of a truly self-routing memoryless interconnect. By themselves, shared memory switches do not scale to large number of ports or high port speeds since memory access time becomes a bottleneck. We also study the impact of different buffering strategies on the proposed architectures. We considered dedicated output buffering, shared output buffering
and a combination of input with output buffering (both dedicated and shared). We also introduce the notion of link-capacity back pressure mode of operation for the case of input-output buffering. This method eliminates cell loss in the memoryless interconnection part of the switch.

5. Since multicast traffic will be a significant portion of the overall traffic we consider the performance of one of the proposed switch architectures as a copy network and analyze its performance under multicast traffic, both uniform and bursty by simulation methods. The proposed copy network achieves very low cell loss even at very high output offered loads.

6. We propose a very effective modular implementation of the proposed switch architectures and have analyzed the implementation complexity, both in terms of the interconnection complexity and the chip count required for their implementation. We show that the architectures proposed can be scaled to large dimensions with reasonable increase in interconnection complexity and chip cost per port.

1.4. Organization of the thesis

Chapter 2 presents a comprehensive survey of existing switch architectures. It takes an in-depth look into the existing ATM switch fabrics and highlights their drawbacks and the improvements required for large scale switching.

Chapter 3 looks at the architecture of the FAB in terms of the interconnection complexity and partitioning for large scale implementations. It also addresses the issue of modular growth and scalability of the FAB switch.

Chapter 4 introduces the output buffered fat-banyan. The performance under uniform traffic conditions is analyzed. Simulation results are presented for general traffic patterns such as non-uniform and hot-spot. The performance of the output-buffered banyan is further enhanced by using small input buffers and a backpressure mechanism. Simulation results under independent uniform and bursty traffic conditions are presented.
Chapter 1. Introduction

Chapter 5 looks at the issue of growability of the switch fabric. The idea of truncated switch fabrics is introduced. System performance is analyzed by computer simulations.

Chapter 6 addresses the issue of multicasting. The FAB network is used as a copy network to perform the function of multicasting. A novel aspect of this copy network is that very low cell loss is achieved even when the output offered load is high.

Chapter 7 concludes the thesis and outlines future research problems.
Chapter 2 ATM Switch Architectures

2.1. Introduction

Large scale ATM switch architectures support high aggregate bit rates in the range of hundreds of Gb/s to tens of Tb/s. Such architectures cannot be realized as single stage networks of either shared-memory or shared-bus using current technology. Some form of multistage interconnection network is to be used to handle aggressive link rates in the range of OC-48 (2.5 Gb/s) to OC-192 (10 Gb/s). These link rates can support bandwidth demanding applications. The high link rates also ensure very efficient statistical multiplexing of applications with peak rates in the range of tens of Mb/s [1]. As ATM networks are installed they may begin to revolutionize business practices, which may in turn lead to greater demands on the network. Both of these questions can only be answered as ATM adapts to the ever changing user requirements over time. Therefore, ATM switches which are designed today should adopt a core switching architecture which can easily adapt and scale to all future networking requirements. In this chapter we first look at the different types of switch architectures that have been proposed in the literature. We also survey various Banyan and Benes based multistage architectures and discuss their relative merits and demerits. Finally, we discuss the important concept of Growable switches, initially proposed by Karol et al [11].

2.2. Types of ATM Switches

ATM switch architectures can be broadly classified into the following three types: Shared-memory switching, Shared-medium switching and Space-division switching. In the following subsections, a brief review of these switch types is given.

2.2.1 Shared-Memory Switches

A shared-memory switch consists of a single dual-ported memory shared by all input and output lines [12, 13, 5, 6]. The incoming cells are multiplexed into a single stream and
Chapter 2. ATM Switch Architectures

are written into the shared memory as shown in Figure 2.1. The memory can be logically organized as either fully shared, partially shared or completely partitioned with respect to the destination output ports. Full sharing leads to minimum memory requirements to achieve a certain cell loss probability; but may be unfair to other ports when the traffic is bursty, causing degradation in their performance. In complete partitioning, each output port has a dedicated queue and requires larger memory to achieve the same cell loss probability as the full sharing scheme. Partial sharing can be used to achieve optimal size and performance. At the outputs, cells are read out in a single stream, demultiplexed and transmitted on the output lines. Therefore, the control logic in the shared memory architecture must be able to handle N incoming cells, queue them in the proper addresses in memory and select N outgoing cells in each time slot. The main bottleneck, which restricts the number of input-output lines of a single switch, is the memory cycle time. In order to accommodate all input and output traffic in a single time slot, the memory bandwidth, which is defined as the average number of words accessed per second, must be at least the sum of the bandwidths of the incoming and outgoing lines. For a given memory cycle time, the number of input-output lines is constrained by the following equation:

\[
\text{Memory cycle time} = \frac{\text{cell length}}{2 \times N \times \text{link speed}}
\]

For a memory cycle time of 10 ns, link speed of 155 Mbps, and cell length of 53 bytes, the maximum possible switch size is \(N=136\). If the link speed is 2.4 Gbps then \(N=8\).

Three basic types of shared memory switches have been proposed and analyzed in the literature. The Prelude switch [12] was the first shared-memory type switch developed.
for ATM and based on the complete partitioning approach. One of its drawbacks is that it ties the number of input/output ports to be equal to the number of bytes per packet. The second type of shared-memory switches discussed in the literature are the linked-list based switches [13], where the buffer memory is logically organized as N linked lists, one for each output. The linked-list based approach achieves complete sharing of the memory, and hence lesser memory is required to achieve a given cell loss probability. The third type is the hybrid (shared and dedicated output buffer) switch [5, 14]. The advantage of this switch is that overloading of the shared buffer by some output ports during bursts can be easily controlled, by controlling the queue sizes at each output. An architecture which does away with the multiplexing and demultiplexing stages by replacing them by crosspoint space-division switches, thus increasing the memory bandwidth, is described in [5]. However, the hardware complexity of the switch is increased due to the introduction of the crosspoint switches, and hence this design is not suitable for large switches. Better switch scalability can be achieved using single-stage interconnection [15] and multistage interconnection [5] of shared-memory switches. The major challenge in designing these switches is minimizing the hardware complexity to achieve the desired performance.

In summary, a single shared-memory switch has the best hardware utilization, because the two required functions of a packet switch, queueing and switching, are carried out via buffering. Hence, shared-memory switches are of particular interest to small to medium sized switches with OC-3 or OC-12 links. However in order to make them viable for large switches, improvement is required in the following aspects:

1. Schemes to improve the memory cycle time. For this purpose cost-effective parallel access schemes need to be investigated.

2. Methods to efficiently interconnect the shared-memory switches so that maximal (optimal) sharing of the buffer-memory is achieved.
2.2.2 Shared-Medium Switches

A shared-medium switch consists of a common high-speed medium (typically a high-speed bus) onto which the input lines are synchronously multiplexed as shown in Figure 2.2. The bandwidth of the bus is equal to $N$ times the rate of a single input line. At the output, there is an address filter and a FIFO buffer for each output line. The function of the address filter is to extract the packets which are destined to a particular output and to store them in the FIFO buffer for that output. As the number of links attached to the medium and their speeds increase, the medium speed becomes a bottleneck. Not many shared-medium architectures have appeared in the literature. The first shared medium switch for ATM was the ATOM (ATM Output Buffer Modular) switch [16]. It uses a bit-slice organization to alleviate the bottleneck of the medium speed. The PARIS (Packetized Automated Routing Integrated System) shared-medium switch [16] was designed for variable length packets. More recently a bus structure based on sequentially scheduling the inputs and parallel transfer [17] has been proposed. However a parallel bus of 425 wires is used, which seems to be excessive. Many current and earlier commercial switches are based on shared-bus architectures, including switches from FORE systems and Newbridge. The shared-medium architecture results
in separate queues for each output, and hence requires more memory in order to achieve 
a required cell loss probability. However, the shared medium supports broadcasting and 
multicasting in a natural way.

2.2.3 Space-Division Switches

Space-division switches can be broadly classified into two major categories based on 
their routing capabilities:

1. Non-blocking switches: Non-blocking switches are switches in which internal block-
   ing within a switch will not occur due to the existence of a large number of non 
   overlapping internal paths (When two or more cells contend for an internal link, only 
   one can pass through the link, the rest have to be dropped. This is referred to as 
   internal blocking). However, blocking may occur at the output of the switch when 
   more than one packet are destined to the same output line.

2. Blocking switches: In these switches internal (as well as output) blocking will occur 
   when two or more packets contend for the same link within the switch.

   Internal blocking and output contention cause degradation in throughput. In order 
to reduce the amount of throughput degradation, buffers may be provided at the input 
ports, output ports or internally at the switching elements, respectively referred to as input 
buffering, output buffering, and internal buffering [18]. Input buffering suffers from low 
throughput (about 58.6% with FIFO buffers) due to head-of-line blocking effects. With 
output buffering, all cells contending for the same output port are stored at the output 
ports until they can be read out. Output buffering increases the throughput over input 
buffering, since more than one cell can be delivered to the output when output contention 
occurs, which is not possible with input buffering. Internal buffering is used to alleviate 
internal blocking. But it has several limitations; the complexity of the switching elements 
is increased and also buffers introduce random delay within the switching fabric, causing 
undesired cell-delay variation in the network.
2.3. Banyan-Based Switch Architectures

The self-routing property of banyan networks provides great simplicity in the control of the switching elements and hence makes them attractive for implementing high speed switching nodes. An $N \times N$ banyan network comprising $2 \times 2$ switching elements consists of $N/2 \times \log N$ switching elements, interconnected such that there is an unique path from any input to any output. However, because of the internal blocking property of banyan networks, they cannot be directly used for switching purposes. Therefore various topologies based on the banyan network have been studied in the literature [7, 6] few of which are illustrated in Figure 2.3. The work on these architectures has mainly focused on the performance enhancement of the banyan fabric with no significant work on the resource utilization or scalability of the architecture. Several of the enhanced networks are described below.
Figure 2.3. Banyan-based ATM switch architectures.

**Dilated Banyan Switch:** Dilated banyans [19, 20, 21] employ fixed dilation of the input and outputs of a switching element. Dilation results in improved performance of the banyan network by increasing the internal bandwidth. However, the internal bandwidth of the switch fabric is not optimally utilized, being under-utilized in the initial stages and the degree of dilation may not be appropriate for the later stages.

**Buffered Banyan Switch:** Turner [7] was one of the pioneers in proposing a design for a fast packet switching network. The main component of Turner's switching network was a buffered banyan network, tailored to the needs of a fast packet-switching
Chapter 2. ATM Switch Architectures

network. When an internal conflict occurs, the buffered-banyan network buffers the blocked packets, which are resubmitted for contention in the next cycle. However, the buffered-banyan network has a limited throughput [22, 23, 24, 25] which decreases rapidly for increasing $N$, for an $N\times N$ network. At $N=1024$, throughput equals 0.45 for a single-buffered banyan. The throughput increases with increasing buffer size, and reaches to around 0.6 for a buffer size of four. Beyond buffer size of four, improvement is negligible. Moreover, from the cost and delay point of view, buffered-banyan networks are not optimum. Hence the buffered-banyan model is not an appropriate model for fast large-scale packet switching.

**Tandem Banyan Switch:** The Tandem Banyan network [26] is constructed by placing several banyan networks in series. The idea is to misroute the packets when a contention occurs, and start a new routing at the subsequent banyan network. The Tandem Banyan switch achieves output queueing (and a high throughput as well) and reduces the size of the concentrators, by making it proportional to $k$, where $k$ is the number of banyans in series ($k$ is much smaller than $N$). Under independent uniform traffic with $N=32$, input load $p=1$, nine banyans (i.e., $k=9$) are required to achieve a cell loss probability of $10^{-6}$, and for $N=1024$, fourteen banyans ($k=14$) are required. However, the tandem banyan is not optimal in the use of the hardware, requiring $k\times(N\log N)^1$ switching elements. For large $N$, placing $k$ banyans in tandem would be excessive from a hardware point of view. Further, the series connection leads to synchronization problems between banyan networks on multiple chips and boards.

**Overlapped Banyan Switch:** The overlapped banyan [27] reduces the number of stages compared to the tandem banyan by embedding several banyan networks within the switch fabric. The basic idea is that, whenever a contention occurs, the cell is misrouted and a new routing is started immediately from the next stage of switching elements.

\footnote{All logarithms are assumed to be base-2 unless indicated otherwise.}
Chapter 2. ATM Switch Architectures

(SE’s), unlike the tandem banyan where the misrouted packet starts a new routing only after arriving at the output of one of the banyan networks in cascade. Once the packet reaches the required position in the overlapped banyan, it is passed to the output via bypass links. For \( N=64 \), \( p=1 \), 28 stages of SE’s are required to achieve a cell loss probability of \( 10^{-6} \). For \( N=1024 \), 54 stages are required, whereas for the tandem-banyan 140 stages (in 14 banyans) would be required. Though the overlapped banyan requires a smaller number of stages compared to the tandem banyan, the number of stages required is still quite large considering the fact that the switching elements are a little more complex than those of the tandem-banyan. Further, both the tandem banyan and the overlapped banyan have to deal with the packet resequencing problem, which introduces a delay overhead.

**Batcher-Banyan Switch**: This is another class of architectures based on the Batcher-banyan network [28, 29, 30], which make use of the property that a banyan network becomes internally non-blocking if the inputs are sorted with respect to the destination requests. In order to achieve the sorting of the incoming packets, a Batcher sorter, based on the principle of bitonic sorting is employed. An \( N \)-input Batcher sorting network consists of \((\log N \times (\log N + 1))/2\) stages of comparators which is rather large, considering the fact that the banyan network itself employs \( \log N \) stages. Moreover, the Batcher sorting network does not contribute to any improvement in throughput when multiple requests are destined to the same output. Additional networks like a trap and a concentrator [30] are required to filter out the excessive packets and resubmit them for routing in the next time slot. Thus, the hardware overhead is large to achieve the required performance.

### 2.4. Benes-based Networks

An \( N \times N \) Benes network consists of \( 2\log_2 N - 1 \) stages of switching elements and provides \( N/2 \) different paths for any input-output pair. A Benes network is illustrated in
Figure 2.4. Two basic types of Benes switches have been proposed. These are the Input buffered Benes and the Buffered Benes.

**Input Buffered Benes:** In this switch each input port is equipped with an input port controller (IPC) which contains an input buffer, a path table for that port, and a link-flags vector [31]. The IPC’s of the different input ports are linked in a circular manner by links that allow each IPC to communicate with its two neighbor IPCs. Starting with a particular input port (e.g. port 0) a path to an output is searched from a window of the first \( w \) cells from the input FIFO (first-in first-out) queue. When a path is determined, the bits of the link-flags vector corresponding to the selected links is marked accordingly. The link-flags vector is passed on to the IPC of the following input port and the search and marking procedure is repeated until all the ports have been visited. Once a path is determined, it is encoded using a routing tag which is used to self-route the cell from the input queue to the output port. The switch hence, achieves a non-blocking operation. With \( w=1 \) the throughput is limited to 0.58 due to HOL (head of line) blocking. Higher throughput is achieved by using \( w>B/2 \) where \( B \) is the input buffer size per port. This architecture does not scale to large switch sizes or high link rates. One of the bottlenecks is the centralized scheduling which would be very slow for large switch sizes and also would not work at high link rates. The other bottleneck is the window searching scheme at the input FIFOs which again slows down the routing operation through the switch.

**Buffered Benes:** The buffered Benes network has also been used as an ATM switching fabric [1]. The WUGS (Washington University Gigabit Switch) switch prototype which was developed at Washington University by Jonathan Turner and others is a prime example of the buffered Benes switch. More recently switch fabrics marketed by IgT Inc. fall under this model. The switching element (SE) design in the WUGS switch is based on a highly parallel shared-buffer architecture. The SE’s are equipped with a backward (back-pressure) signalling capability that eliminates the possibility of cell loss.
in any SE. The cells are routed to their destinations using a two-phase technique. In the first phase, cells are routed randomly to the outputs of the SE's at stage $\log_2 N$ (i.e., the outputs of the banyan subnetwork) such that these outputs are evenly loaded. In the second phase, cells are routed to their respective destinations using the reverse banyan subnetwork. This technique leads to cells arriving in an out-of-sequence order at the outputs. This is dealt with by providing resequencing buffers at the switch output. Also, this switch supports multicasting. Multicasting is supported by successive application of copy-by-two routing function in the Benes network \[32\]. A connection with $m$ destinations requires $\log_2 m$ passes through the switch fabric. The main disadvantage of this fabric is that it is internally buffered and could lead to QoS coupling in the cell delay and cell delay variation of the connections passing through it.

![Benes network of size $N=16$.](image)

**Figure 2.4.** Benes network of size $N=16$.

### 2.5. Growable Switch

Some networks based on the Clos network topology \[33, 11\] have been proposed for ATM switching. Of these the most important one is the growable switch model which is discussed in this section.
The growable packet switch is based on a 3-stage $N \times N$ Clos network topology, with the first two stages providing a memoryless interconnection capable of routing up to $m$ packets to each group of $n$ output ports, where $m \leq n$. There are $N/n$ output groups, and an $m$-to-$n$ packet switch module provides the buffering and final routing for each group of $n$ output ports. The growable packet switch is illustrated in Figure 2.5. The main drawback of this architecture is that the distributed routing scheme used is not self-routing and therefore will not scale with high port speeds or larger port sizes. The details of the routing algorithm can be found in [11]. The routing is performed essentially by passing "link-reservation" vectors among the cross-bar switches of the first stage of the memoryless fabric. The vectors are updated as they are passed from one cross-bar to another. This requires an additional bus to connect the cross-bar switches of the first stage. The advantage of this interconnection network is that it is highly modular as it employs crossbar switches only in a two-stage memoryless fabric. Also, shared memory modules are employed to serve groups of output ports.

Figure 2.5. The growable switch model
2.6. Drawbacks and Improvements Required

In the previous section we have mainly considered three types of multistage interconnection networks and their variations. The Banyan based packet switches are attractive because of their self-routing property. However, unbuffered or buffered Banyan switches are known to suffer from severe blocking which leads to limited throughput [22, 34, 6, 24]. A wide variety of techniques studied in the literature were presented in the previous section. The Benes network when used as an input buffered switch does not scale with switch size or port speeds. The buffered Benes has been shown to perform well under balanced traffic conditions [1]. However, it suffers from QoS coupling within the fabric. Finally, the three-stage growable switch has its advantage of modularity but does not scale due to the limitation of its routing algorithm.

An ideal multistage interconnection network for an ATM switch which can support high link rates and large number of ports, must have the following properties:
1. Uses a fast self-routing scheme.
2. Low interconnection density.
3. Very low QoS coupling (in terms of average delay, cell delay variation and cell loss).
4. Is capable of modular growth.
5. Is scalable in a cost-effective (low chipcount per port) manner.

With the above criteria in mind we have proposed a high-capacity switching fabric based on incremental internal dilation called the fat-banyan (FAB) [35, 36, 37]. The FAB is discussed in detail in chapter 3. A modified version of the FAB called the Truncated FAB (TFAB) [38, 39] is also presented in chapter 5. The TFAB makes effective use of available memory technology and provides a scalable solution to the design of shared-memory switches as will be shown in chapter 5.

An unbuffered Benes with its second half (the reverse banyan subnetwork) dilated
like the FAB may serve as a very good switching fabric as it eliminates QoS coupling (in terms of delay and cell delay variation) and also achieves load balancing among the internal links within the fabric. The internal load balancing may lead to reduction in the dilation required to achieve a certain cell loss. This study however is not part of this thesis and has been recommended for future work.

2.7. Conclusion

In this chapter we discussed space division architectures and commented on their suitability for large scale ATM switching in the range of 10–100 Tb/sec. The characteristics required of such fabrics were outlined in section 2.6. In the subsequent chapters we will evaluate and analyze in detail the FAB network and show that it is capable of large scale ATM switching in a cost effective (high in resource utilization) and scalable manner.
Chapter 3 Bandwidth Optimization in Banyan Switches

3.1. Introduction

The internal blocking of the banyan network (see Figure 3.1) can be improved by placing internal buffers or by increasing the internal link capacity as was seen in the previous chapter. The question of whether to use internal buffers or to increase the internal link capacity depends on a number of factors including:

1. The extent of performance improvement in terms of cell loss, delay, and throughput.
2. Feasibility and cost of implementation which includes such factors like interconnection and I/O density, switch modularity, and growability.
3. Impact on QoS coupling and cell sequence integrity.

In this chapter we first proceed to carefully examine these crucial factors in some detail.

*Buffered-banyan networks* [7, 22, 24, 40, 23] are based on using internal buffers. In these networks buffers are placed at the inputs (or outputs) of each switching element. Performance studies for both uniformly distributed output requests [24] and non-uniformly distributed output requests [23] have been reported. By using multiple buffers a saturation throughput of 0.7 [7] is obtained under uniform traffic conditions. Under non-uniform traffic conditions, the performance of these networks is severely degraded, and for very large networks [23] the throughput may drop to nearly zero. One solution that has been suggested is the use of a randomizing network (also called distribution network) as in Benes switches [41] to break up any traffic patterns which may cause severe congestion to persist. But such randomization may lead to QoS coupling within the switch fabric. For example with multipath networks, such as the buffered Benes, randomization is carried out in the first half of the network, which causes different flows to affect each other's...
Figure 3.1. Two 16×16 banyans a) with 2×2 elements and b) with 4×4 elements. cell delay and cell loss within the fabric. Thus, buffered-banyan networks suffer from performance limitations.

The other resource of the switch is its internal link capacity. Dilated banyan networks [20, 19, 42, 35, 38] are based on increasing the internal link capacity. This, in effect, increases the internal speed up of the switch by using parallel paths. Alternatively, researchers have considered increasing the number of ports of a switching element [7, 19] to increase bandwidth as shown in Figure 3.1. However, this approach increases the bandwidth within a switching element only but it does not increase the bandwidth
between different stages of the switch. Dilated networks do not efficiently utilize the internal bandwidth, while networks with large number of ports per switching element, do not provide sufficient bandwidth. The main bottleneck in a banyan network is the limited bandwidth of the inner links. From a performance point of view, dilated networks are attractive as they provide sufficient bandwidth. Also the QoS coupling of different flows within the switch fabric is limited to cell loss which can be made arbitrarily small by employing sufficient dilation. While the performance of the dilated banyan has been studied for random uniform traffic \[20, 19\], a comprehensive investigation of such factors as the impact of different traffic patterns, use of shared output queueing (refer to chapter 5) and various levels of dilation is lacking. In this chapter we propose a unifying model for banyan networks with variable degrees of dilation and introduce a new model called the \textit{fat-banyan} (FAB) switch and provide detailed description of its architectural features, modular construction, and feasibility of constructing large capacity (Terabit-per-second) switches from basic modules. We also provide useful estimates on the number of chips and the interchip wiring required to construct switches of a given size and external link capacity. The loss performance of the unbuffered fabric as a function of dilation and switch element size will be discussed in the next chapter. Although originally conceived as an output-buffered switch, the FAB switching fabric can be optimized for input-buffering, input-output buffering, and shared buffering. We also defer any discussion on buffer placement and sizing to the next chapters.

\section*{3.2. The Fat-Banyan Switching Fabric}

The \textit{fat-banyan} (FAB) switch has switching elements (SEs) with multiple input and output links per port. The number of input and output links per port may not be equal, and we call such an SE as a \textit{fat SE} (FSE). Figure 3.2 illustrates the difference between a FSE and a standard buffered SE. In a \(2 \times 2\) buffered SE if an incoming cell cannot be routed to one of the two output ports because of contention, it will be buffered. In
the FSE, cells can be routed to dilated output port based on availability of links. In the former case cells will be lost if buffers are full, and in the latter case cells will be lost if the links are not available. For the FAB switch, the number of input and output links will grow in the first few stages of the switch. However, in the remaining stages, the number of links can remain fixed or even decrease depending on the type of traffic pattern. The incremental increase in dilation aims at eliminating the need for internal buffering by increasing internal bandwidth. This is based on the observation that in multibuffered-banyan networks, performance improvement has been achieved for buffer sizes of up to four cells. With buffer sizes greater than that, the improvement in performance is negligible [24, 25]. Moreover, theoretical analysis on congestion in banyan networks has shown that under uniform traffic conditions, the congestion of all but $1/N^\alpha$ of the routing problems with a single packet per input in a banyan network of size $N\times N$ is at most $O(\alpha)+o(\alpha\log N)$, for any $\alpha$ [43]. Since the congestion is a function of $\log N$, the number of links in the FSE's of the FAB switch would not grow to be very large.

There are several advantages of the FAB switch over a multibuffered-banyan. Mainly, there are no queueing delays inside the FAB switch, and the throughput of the FAB switch reaches nearly unity under uniform traffic conditions (as will be shown in a later chapter), while the throughput of the multibuffered-banyan saturates at around 0.7 [24]. A detailed analysis of the fundamental complexities of the crossbar-based (knockout switch) and banyan-based switches has been reported. It has been shown that [21] for a given cell loss probability requirement, the dilated-banyan network has the lowest order of complexity among the space-division switches that have been proposed to date. By utilizing variable dilation, the FAB switch achieves a lower order of complexity than the banyan switches with constant dilation, and also provides a more general banyan switch model [36] in which buffer-bandwidth trade-offs can be fully investigated and optimized. In any particular stage of the FAB switch the FSE's are of the same type. The notation $L_i$
Chapter 3. Bandwidth Optimization in Banyan Switches

Figure 3.2. (a) 2x2 Buffered SE. (b) 2x2 Fat SE with output concentrator. (3) 2x2 fully dilated Fat SE (no output concentrator required).

where $i$ is the stage number will be used to denote the output link dilation of the FSE's in stage $i$. Also any given FAB network is characterized by a dilation configuration (DC) represented by the set DC =$\{L_1,L_2,L_3,...\}$. Figure 3.3 illustrates an 8x8 FAB switch with dilation configuration DC =$\{L_1,L_2,L_3\} = [2,3,4]$.

3.3. Fat-Banyan Architectural Features

The FAB architecture can be viewed as a generalized version of the dilated banyan architecture (a 2 dilated (DC=$[2,2,2]$) 8x8 banyan network is shown in Figure 3.4). The dilated banyan can be operated in two ways: one is to route cells over any one of the multiple links at each input and the other is to route only over one link at each input then use more links at subsequent stages [20]. In the latter case the performance of the dilated architecture is similar to that of the FAB architecture. However, the gradual increase in dilation of the FAB topology leads to a much lower construction cost not
Figure 3.3. 8×8 output-buffered FAB switch with dilation configuration as follows: DC=[2,3,4].

Figure 3.4. 2 dilated 8×8 banyan network.

only because of savings in cross-connect hardware but, more importantly, because of the resulting reduction in interchip and interboard wiring and the resulting modularization of the architecture as will be explained next.

Modular Expansion of Switching Elements: Figure 3.5 shows our proposal for constructing FSEs of arbitrary input and output dilation using basic FSEs of fixed size and dilation pattern. In its simplest form a 2×2 FSE with an input link dilation of $L_{in}=1$
Chapter 3. Bandwidth Optimization in Banyan Switches

and output link dilation of $L_{out}=2$ is called a basic FSE (BFSE). A parallel arrangement of BFSEs can be used to construct $2 \times 2$ FSEs of varying input and output dilations. Configuring a FSE of a given size and dilation can be done simply by pin assignment to a generic chip containing several BFSEs as shown in Figure 3.5. Two pin assignments for two types of FSEs are shown in Figures 3.6 and 3.7. Each line in the above Figures represents one link which may consist of several wires routed through several I/O pins. With this construction a fully dilated FSE is obtained. An FSE is said to be fully dilated if the number of links per output port is equal to the total number of its input links.

Other variations of this construction can be devised to implement FSE's which are not fully dilated. Output concentrators will be required within each FSE to reduce output dilation in this case.

![Figure 3.5. Generic package of BFSEs](image)

Figure 3.5. Generic package of BFSEs
Banyan Partitioning Strategy: It should be emphasized that the reduction in interconnection complexity in the initial stages of the FAB has a significant impact on reducing the complexity of inter board wiring, which can lead to improved packaging for large scale implementations. An isomorphic representation of the banyan network is shown in Figure 3.8, where $D_{n,d}$ denotes a delta network [19, 44] with $n$ inputs and outputs constructed of $d \times d$ crossbars. This representation is particularly useful for partitioning
large banyans among multiple chips or multiple boards. For the FAB network, the best strategy is to horizontally slice through the network as shown in Figure 3.9. The FAB architecture allows the modular growth to a large scale ATM switch. For a large scale ATM switch with say a throughput of the order of terabits/sec an important issue is to break the design into modular units and interconnect these units. This is done by putting the modular units onto boards, which are placed in shelves which in turn interconnect the boards through backplanes. A rack of shelves may be needed to house these boards with additional cabling required to interconnect the shelves.

Figure 3.8. Isomorphic representation of a banyan network

This partitioning strategy is justified in view of the following facts:
Chapter 3. Bandwidth Optimization in Banyan Switches

1. The wiring becomes more and more localized to within smaller subsets of adjacent FSEs as we go from the input stage to the output stage because of the banyan topology.

2. The number of wires required for interconnecting the FSE's on different boards is limited to the first few stages which have the smallest levels of link dilation.

In general, the FAB switch can be partitioned among multiple boards by horizontal slicing of the network and placing each slice on a distinct board. Partitioning a \( N \times N \) FAB among \( K \leq N \) boards is done by placing \( N/K \) input ports, \( N/K \) output ports and \( N/2K \) FSEs from each stage of the network on a single board. Assuming that \( N=2^n \) and \( K=2^k \), the maximum number of external links between two boards is

\[
\sum_{i=1}^{\log_2 k} (N/K)L_i
\]

where \( L_i \) is the dilation of stage \( i \). As an example consider a FAB network of size \( N=256 \) with dilation configuration, \( \text{DC}=[2,4,8,8,8,8,8] \). If the FAB is partitioned among 2 boards the maximum number of external wires between two boards is 256. The maximum number of wires interconnecting two boards will be 384 if the FAB is partitioned among 4 boards and 448 if the FAB is partitioned among 8 boards. Table 3.1 compares the maximum number of external wires crossing two boards. We see that the interboard wiring for the FAB switch is greatly reduced compared to the dilated banyan.

In comparison to the dilated banyan network, the FAB network does provide the required internal bandwidth while significantly reducing the interchip/interboard interconnection compared to other switch architectures such as the Knockout switch [45], Buffered Benes [1, 31] and 3-stage Clos [11]. Additionally the FAB offers a modest saving in the number of FSEs. Table 3.2 compares the dilated banyan and FAB architectures of different sizes assuming that the network is built from 2 dilated FSEs (\( L_{in}=2 \), \( L_{out}=2 \)). It should be noted that 2 dilated FSE's require \( 4 \times 2 \) concentrators at each output to provide arbitration if the number of cells contending for an output is greater than 2. For a \( 16 \times 16 \) FAB switch with \( \text{DC}=[2,4,8,8] \) the total number of 2 dilated FSEs is given
Chapter 3. Bandwidth Optimization in Banyan Switches

by: 8+16+32+32=88. It should be noted that in the first three stages of the 16×16 FAB only one of the input links of the 2 dilated FSE’s need to be used, or a much simpler implementation based on the FSE’s built from BFSE’s shown in Figure 3.5 can be used. For a 16×16 dilated banyan switch with a dilation of 8, the number of 2 dilated 2×2 SEs required is given by: 32×4=128.

Table 3.1. Comparison of interconnections required for FAB and dilated banyan networks.

<table>
<thead>
<tr>
<th>Number of boards</th>
<th>Maximum number of external wires interconnecting two boards</th>
<th>Maximum number of external wires interconnecting two boards</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>256</td>
<td>1024</td>
</tr>
<tr>
<td>4</td>
<td>384</td>
<td>1024</td>
</tr>
<tr>
<td>8</td>
<td>448</td>
<td>768</td>
</tr>
</tbody>
</table>

Table 3.2. Comparison of FAB and dilated banyan networks built with 2 dilated 2×2 FSE’s.

<table>
<thead>
<tr>
<th>Switch Size</th>
<th>Dilation Configuration for Fat-Banyan</th>
<th>Number of 2 dilated 2x2 FSEs (Lin=2, Lout=2)</th>
<th>Number of 2 dilated 2x2 FSEs (Lin=2, Lout=2) forming an 8 dilated banyan</th>
</tr>
</thead>
<tbody>
<tr>
<td>16×16</td>
<td>2,4,8,8</td>
<td>88</td>
<td>128</td>
</tr>
<tr>
<td>64×64</td>
<td>2,4,8,8,8,8,8</td>
<td>608</td>
<td>768</td>
</tr>
<tr>
<td>256×256</td>
<td>2,4,8,8,8,8,8,8,8</td>
<td>3840</td>
<td>4096</td>
</tr>
<tr>
<td>4096×4096</td>
<td>2,4,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8</td>
<td>88064</td>
<td>98304</td>
</tr>
</tbody>
</table>

† Note that the FSEs in the first two stages do not require output concentrators and hence have simplified implementations.
Chapter 3. Bandwidth Optimization in Banyan Switches

3.4. Chip Count Estimate for a Large Scale FAB Switch Implementation

In this section we develop a simple but useful cost model that evaluates cost of a switch in terms of pin-limited VLSI routing chips. In this model the cost depends on the FAB size as well as the dilation configuration used. This cost model will be used to evaluate the scalability of the core fabric in terms of the growth in number of chips required per port of the switch.

Cost Model: The main element of our cost model is a basic routing chip that contains multiple BFSEs as was shown in Figure 3.5. In this model the parts of each BFSE can be accessed through the chip I/O. The main limiting factor in our model is the number of I/O pins per chip. This is justified in light of the fact that such a routing chip includes mainly multiplexers, demultiplexers and a few bytes of buffers, and such units will use little area in silicon.

The use of multiple planes to allow the switch to keep up with the external link speed has implications for additional hardware at the inputs (to the separate the cell stream into multiple planes) and outputs (to combine the separate planes).

Large scale switches are expected to support link rates of 2.5 Gb/s to 10 Gb/s or even up to 40 Gb/s [46]. However, with current microelectronic technology I/O speeds are far lower than such rates. Therefore, to support a given high link rate of magnitude \( R \) bits/sec, several I/O pins of rate \( r \) bits/sec (where \( r < R \)) may have to be dedicated to the link. In this case \( \lceil R/r \rceil \) are needed per link of an FSE. In our model if the number of \( \lceil R/r \rceil \) I/O pins is large (e.g. \( \geq 40 \)) the pins can be allocated to multiple planes of chips. We demultiplex the bits from a single external link over \( s \) planes of chips. Hence, each chip in a given plane would have \( \lceil R/rs \rceil \) pins for a single external link. We denote this quantity as

\[
p = \lceil R/rs \rceil.
\] (3.2)
Figure 3.10. Partitioning a BFSE among planes

For example if the link rate \( R = 10 \text{ Gb/s} \) and the I/O rate (per pin) is \( r = 200 \text{ Mb/s} \), and the number of planes of chips used is \( s = 5 \), then the number of pins per chip (per plane) is \( p = 10 \). Figure 3.10 illustrates the method of partitioning a single BFSE among \( s \) planes.

This section provides formulas for partitioning the FAB switch among VLSI chips with \( C \) I/O pins. The partitioning assumes the use of \( d \times d \) BFSEs with input dilation \( L_{\text{in}} = 1 \) and output dilation \( L_{\text{out}} = d \), as shown in Figure 3.11 (for \( d = 2 \) and \( d = 4 \)). It is assumed that a BFSE can actually be partitioned among different chips or to enable the layout of multiple BFSEs on the same chip. We use the following parameters for our calculations:

- External link rate: \( R \) bits/sec
- Number of input links per BFSE: \( d \)
- Number of I/O pins/link per chip: \( p \)
- I/O pin rate: \( r \) bits/sec

37
Number of planes of chips into which a BFSE is partitioned: s
Total I/O pins per chip: C
Number of BFSEs per chip: i

The total number of BFSEs for the FAB switch: T

\[ T = \frac{N}{2} [1 + 2 + 4 + 8 \times (\log_2 N - 3)] \]  

Figure 3.11. BFSEs of varying sizes

The number of I/O pins per link per chip is given by

\[ p = \lceil R/rs \rceil. \]  

(3.3)

Consider a BFSE of size \( d \times d \) as the basic module. The number of BFSEs per chip (per plane) is given by \( i \). It is straightforward to determine that \( i \) is the largest integer which satisfies

\[ idp + id^2p \leq C. \]  

(3.4)

For an \( N \times N \) switch the total number of BFSEs is

\[ T = \frac{N}{2} [1 + 2 + 4 + 8 \times (\log_2 N - 3)]. \]  

(3.5)

The total chip count is given by

\[ \lceil Ts/i \rceil. \]  

(3.6)

Tables 3.3, 3.4, and 3.5 summarize the chip count for a few practical FAB switch sizes

and dilation configurations for external link rates of 2.5 Gb/s and 10 Gb/s. We see that

\[ \text{The results reported in these tables are based on the formula (3.2) which assumes that each BFSE has full access to chip I/O.} \]

\[ \text{Alternatively, multiple stages of BFSEs can be packaged together to further reduce I/O requirements.} \]
as the switch size increases the chip count per port increase in a sub-linear manner. For instance at 2.5 Gb/s in order to go from a switch of size $64 \times 64$ to $256 \times 256$, a scaling by a factor of 4; the number of chips/port increases by a factor of 1.5 only.

<table>
<thead>
<tr>
<th>Switch Size</th>
<th>Dilation Configuration</th>
<th>Calculated parameters</th>
<th>Total Chip count</th>
<th>Chips/port</th>
</tr>
</thead>
<tbody>
<tr>
<td>64x64</td>
<td>2.4.8.8.8.8</td>
<td>$p=1$, $T=992$</td>
<td>124</td>
<td>1.94</td>
</tr>
<tr>
<td>256x256</td>
<td>2.4.8.8.8.8.8</td>
<td>$p=3$, $T=6016$, $i=13$</td>
<td>752</td>
<td>2.94</td>
</tr>
<tr>
<td>4096x4096</td>
<td>2.4.8.8.8.8.8.8.8.8.8.8</td>
<td>$p=1$, $T=161792$, $i=13$</td>
<td>20224</td>
<td>4.94</td>
</tr>
</tbody>
</table>

Table 3.3. Chip count for various FAB switch sizes with external link rate of 622 Mb/s.

<table>
<thead>
<tr>
<th>Switch Size</th>
<th>Dilation Configuration</th>
<th>Calculated parameters</th>
<th>Total Chip count</th>
<th>Chips/port</th>
</tr>
</thead>
<tbody>
<tr>
<td>64x64</td>
<td>2.4.8.8.8.8</td>
<td>$p=3$, $T=992$, $i=13$</td>
<td>382</td>
<td>5.96</td>
</tr>
<tr>
<td>256x256</td>
<td>2.4.8.8.8.8.8</td>
<td>$p=3$, $T=6016$, $i=13$</td>
<td>2314</td>
<td>9.04</td>
</tr>
<tr>
<td>4096x4096</td>
<td>2.4.8.8.8.8.8.8.8.8.8.8</td>
<td>$p=3$, $T=161792$, $i=13$</td>
<td>62228</td>
<td>15.19</td>
</tr>
</tbody>
</table>

Table 3.4. Chip count for various FAB switch sizes with external link rate of 2.5 Gb/s.

<table>
<thead>
<tr>
<th>Switch Size</th>
<th>Dilation Configuration</th>
<th>Calculated parameters</th>
<th>Total Chip count</th>
<th>Chips/port</th>
</tr>
</thead>
<tbody>
<tr>
<td>64x64</td>
<td>2.4.8.8.8.8</td>
<td>$p=10$, $T=992$, $i=4$</td>
<td>1240</td>
<td>19.375</td>
</tr>
<tr>
<td>256x256</td>
<td>2.4.8.8.8.8.8</td>
<td>$p=10$, $T=6016$, $i=4$</td>
<td>7520</td>
<td>29.375</td>
</tr>
<tr>
<td>4096x4096</td>
<td>2.4.8.8.8.8.8.8.8.8.8.8</td>
<td>$p=10$, $T=161792$, $i=4$</td>
<td>202240</td>
<td>49.375</td>
</tr>
</tbody>
</table>

Table 3.5. Chip count for various FAB switch sizes with external link rate of 10 Gb/s.
3.5. Comparison of FAB Chipcount with other Switch Fabrics

In the previous section the chipcounts were derived based on the assumption that each BFSE has full access to chip I/O. In this section we use an alternative partitioning approach for the switch fabric. This approach is based on partitioning the fabric to minimize the chip count by packaging several banyan stages within the switch fabric. This approach has been used previously in [44] where chipcount equations for various switch fabrics: Knockout, Tandem banyan etc. are derived. In this section we compare the chipcount of the FAB switch with the Knockout, Dilated banyan and the Tandem banyan using the above approach. The parameters for the switching network are so chosen that the cell loss performance under independent uniform traffic conditions is of the order of $10^{-6}$. For all the networks considered, in order to match the external data rate $[R/r]$ parallel planes are required, each a bit-serial network.

**Knockout Switch:** As we have seen previously the Knockout switch is a crossbar switch. A simplified block diagram of this switch is shown in Figure 3.12. The details of the chipcount derivation for the Knockout switch can be found in [44]. Each bit-plane of the Knockout network has $N$ Knockout concentrators with $N$ inputs and $L$ outputs. An $N:L$ concentrator can be built from a tree of $p':L$ concentrators as shown in Figure 3.13. The number of levels of $p':L$ chips required is given by the smallest integer $i$ satisfying

$$N \left( \frac{L}{p'} \right)^i < L \quad (3.7)$$

and the number of chips per level is given by

$$\left( \frac{N}{L} \right) \left( \frac{L}{p'} \right)^j \quad \text{where} \quad 1 \leq j \leq i. \quad (3.8)$$

Assuming $p'=C$, the total I/O pins per chip, the chipcount for a one-bit knockout concentrator, $C_k$, is

$$C_k = \frac{N}{L} \sum_{j=1}^{i} \left( \frac{L}{C} \right)^j. \quad (3.9)$$
There are $N$ Knockout concentrators per plane and hence the total number of chips for the switch is

$$N \lceil R/r \rceil C_k. \quad (3.10)$$

Table 3.6 lists the chipcount for two switch sizes.
Table 3.6. Chip count for Knockout switch (as reported in [44])

**Banyan Switch:** Consider an \(N\times N\) banyan switch constructed from \(d\times d\) crossbars. In order to package this using chips with total I/O of \(C\) we require \(\frac{N}{C/2}\) chips to accommodate \([\log_{d}C/2]\) stages [44]. There are a total of \(\log_{d}N\) stages. Hence, the total number of chips required is

\[
R = \frac{1}{r} \cdot \frac{N}{C/2} \cdot \left[ \frac{\log_{d}N}{\log_{d}C/2} \right].
\]

The chipcount estimates are shown for two switch sizes in Table 3.7.

Table 3.7. Chip count for a Banyan switch

**Tandem Banyan Switch:** Consider an \(N\times N\) Tandem banyan switch constructed from \(d\times d\) crossbars with \(k\) banyan networks (of size \(N\times N\)) connected in cascaded. The total number of chips required are the sum of the chips required for the cascade of \(k\) banyan networks plus the number of chips required for packaging the output concentrators [44]. Hence, we have the total chip count as

\[
k \left[ \frac{R}{r} \cdot \frac{N}{C/2} \cdot \left[ \frac{\log_{d}N}{\log_{d}C/2} \right] \right] + C_{out}.
\]

\(C_{out}\) is the number of chips required to package the output concentrators and their values are calculated based on the pinout requirement and is shown in Table 3.8.
Chapter 3. Bandwidth Optimization in Banyan Switches

<table>
<thead>
<tr>
<th>Switch Size</th>
<th>Total Chip count</th>
<th>Chip count/port</th>
</tr>
</thead>
<tbody>
<tr>
<td>256x256, k=9, C_out=3</td>
<td>363</td>
<td>1.41</td>
</tr>
<tr>
<td>1024x1024, k=14, C_out=5</td>
<td>2245</td>
<td>2.19</td>
</tr>
</tbody>
</table>

Table 3.8. Chip count for the Tandem Banyan switch

**Dilated Banyan Switch:** Consider an \( N \times N \) dilated banyan switch with \( d \times d \) switch elements each with a dilation of \( L \). In order to package this using chips with total I/O of \( C \) we require \( \frac{Nd}{C/2} \) chips to accommodate \( \lfloor \log_{d}C/2 \rfloor \) stages. There are a total of \( \log_{d}N \) stages. Hence, the total number of chips required is

\[
\left\lfloor \frac{R}{r} \right\rfloor \left\lfloor \frac{Nd}{C/2} \right\rfloor \left\lfloor \frac{\log_{d}N}{\log_{L}C/2} \right\rfloor.
\]

(3.13)

The chip count estimates are shown for two switch sizes in Table 3.9.

<table>
<thead>
<tr>
<th>Switch Size</th>
<th>Dilation Configuration</th>
<th>Total Chip count</th>
<th>Chip count/port</th>
</tr>
</thead>
<tbody>
<tr>
<td>256x256</td>
<td>8.8,8.8,8.8,8.8,8.8</td>
<td>1280</td>
<td>5</td>
</tr>
<tr>
<td>4096x4096</td>
<td>8.8,8.8,8.8,8.8,8.8,8.8</td>
<td>30720</td>
<td>7.5</td>
</tr>
</tbody>
</table>

Table 3.9. Chip count for the Dilated banyan.

**Fat-Banyan Switch:** Consider an \( N \times N \) FAB switch with dilation of the first two stages being 2 and 4 respectively. There is a modest savings in chipcount when compared to the dilated banyan since more number of FSE's can be packed per chip in the initial stages. The first two stages have an average dilation of 3 and the next stage has an average dilation of 6, the rest of the stages are similar to the dilated banyan. The chipcount estimates for different switch sizes and parameters are shown in Tables 3.10 and 3.11.

From the tables above we see that the FAB switch (Table 3.10) has a significantly reduced chipcount compared to the Knockout and modest savings compared to the dilated banyan network. The Tandem banyan network has a very low chipcount. The FAB
Table 3.10. Chip count for the FAB switch with $C=128$ and $R=1.0$ Gb/s.

<table>
<thead>
<tr>
<th>Switch Size</th>
<th>Dilation Configuration</th>
<th>Total Chip count</th>
<th>Chip count/port</th>
</tr>
</thead>
<tbody>
<tr>
<td>256x256</td>
<td>2.4,8,8,8,8,8,8,8</td>
<td>1000</td>
<td>3.9</td>
</tr>
<tr>
<td>1024x1024</td>
<td>2.4,8,8,8,8,8,8,8</td>
<td>5200</td>
<td>5.1</td>
</tr>
<tr>
<td>4096x4096</td>
<td>2.4,8,8,8,8,8,8,8,8,8</td>
<td>25920</td>
<td>6.32</td>
</tr>
<tr>
<td>256x256</td>
<td>2.4,4,4,4,4,4,4,4,4,4</td>
<td>300</td>
<td>1.2</td>
</tr>
<tr>
<td>1024x1024</td>
<td>2.4,4,4,4,4,4,4,4,4,4,4,4</td>
<td>1520</td>
<td>1.48</td>
</tr>
</tbody>
</table>

Table 3.11. Chip count for the FAB switch with $C=240$ and $R=2.5$ Gb/s.

chipcount with DC=[2,4,4,...] is better than that of the Tandem Banyan. As will be shown in the next chapter this dilation configuration when combined with input-output buffering yields an excellent loss performance. Additionally, the FAB has the advantage of lower delay overhead compared to the Tandem banyan.

3.6. Conclusion

In this chapter the FAB switch was introduced and shown to require lesser number of internal links compared to an equivalent uniformly dilated banyan switch. It was shown that incremental dilation of internal links has a dramatic impact on the implementation and scalability of the switch. It was shown that the FAB chipcount depends on the packaging strategy used. Two strategies were proposed. The first is based on the assumption that the ports of each BFSE has full access to the chip I/O. The second approach packs as many stages of FSE's as possible and yields a lower chip count than the first (refer to Tables 3.4 and 3.11). However, the latter strategy is less flexible in constructing a FAB
switch of a given size. The FAB chipcount is significantly better than the crossbar based Knockout and better than the dilated banyan according to the estimates presented in [44].
Chapter 4 Performance of The Fat-Banyan Switch

4.1. Introduction

This chapter discusses the performance of the Fat-Banyan (FAB) switch. The performance of the switch is evaluated by simulation and analysis. The performance of a specific ATM switch depends on the traffic pattern according to which cells arrive at its inputs as well as the distribution of output requests. The traffic pattern is determined by: 1) the arrival process of the cells at the inputs of the switch, and 2) the distribution of the output requests of the arriving cells. In this chapter we have considered the following arrival processes a) Bernoulli process and b) the Interrupted Bernoulli process. Basically two types of distributions have been considered for the output requests i.e., a) Uniform and b) Non-Uniform.

In this chapter we first analyze the performance of the bufferless FAB switch under independent uniform traffic. The performance of the bufferless FAB switch is shown to be very efficient in terms of cell loss probability. The cell loss is easily controlled by choosing an appropriate dilation configuration. Analysis and simulation results under uniform traffic are presented followed by simulation results for non-uniform and bursty traffic patterns. Next, the output buffered FAB switch is considered. Analysis and simulation results are discussed for this case. Finally, we present the performance results for the input-output queued FAB switch with backpressure. This has the advantages that the fabric is rendered lossless, the dilation is reduced and small input buffers are required to achieve a similar cell loss performance as that of the purely output-buffered case.

4.2. Performance Analysis under Independent Uniform Traffic

In this section we consider the simplest traffic pattern, called the independent uniform traffic pattern. The arrival process of this traffic pattern is a Bernoulli process with parameter $p$, at an input port. Also, for this traffic pattern the arrivals at an input are
Chapter 4. Performance of The Fat-Banyan Switch

independent of the arrivals at any other input. Further, the distribution of the destination requests of the arriving cells is uniform over all the output ports for this traffic pattern.

The throughput of the switch is defined as the rate $R(p)$ of cells which reach their requested destinations. The parameter $p$ represents the load offered to the switch. The normalized throughput is defined to be $R(p)/p$, and the mean cell loss rate is given by $1-R(p)/p$.

The analysis is based on the approach presented in [47] for determining the performance of an unbuffered banyan network. The following assumptions are made for the analysis of the FAB switch:

1. The traffic at the input ports conforms to the independent uniform traffic pattern.
2. The performance of a stage can be approximated by that of a single FSE in that stage.
3. The number of input and output links per port of an FSE may vary from stage to stage.
4. Each link of an input port of a FSE can connect independently to either one of its output ports.
5. Cells are removed immediately at the output ports of an FSE.
6. The cells which are blocked at an FSE (cannot reach their destination port) are discarded.
7. No errors are introduced in the operation of the FSEs and the entire switch.

For the analysis we consider a FAB switch with $N$ input ports and $N$ output ports. A FSE can be characterized as a function of three parameters $S(M, D, L)$, where $M$ denotes the total number of input links to each FSE, $D$ is the number of output ports and $L$ is the number of links per output port. We will use the general notation $M_i$, $D_i$, and $L_i$ to denote an FSE in the $i$th stage ($1 \leq i \leq \log N$) of the FAB switch. However, whenever we refer to an arbitrary FSE of the FAB switch we will disregard the subscripts for the
Chapter 4. Performance of The Fat-Banyan Switch

Figure 4.1. Fat Switching Element (FSE).

sake of simplicity. Figure 4.1 illustrates such an FSE. For the purpose of the analysis, a knowledge of the total number of input links per FSE is sufficient, without the need to distinguish between the number of input ports and the number of links per input port.

Let $p_{ij}$ be the probability that $j$ cells are destined to a particular output port of a FSE, given that there were $i$ cell arrivals. Then

$$p(i, j) = \binom{i}{j} \left(\frac{1}{D}\right)^j (1 - \frac{1}{D})^{i-j}$$

(4.1)

Note that an incoming cell has an equal probability of $1/D$, of being destined to any output port of an FSE. An output port can accept only $\min(j, L)$ cells destined to it. If $j > L$ then $j - L$ cells are lost. We define $E(i)$ as the expected number of cells that the FSE passes successfully to its outputs per clock cycle, given that $i$ cells arrived at the beginning of that cycle. Thus,

$$E(i) = D \sum_{j=0}^{i} p(i, j) \min(j, L)$$

(4.2)

The mean number of cells at the output of the FSE per clock cycle is then given by

$$E = \sum_{i=0}^{M} E(i) q(i)$$

(4.3)

where $q(i) = \binom{M}{i} p^i (1 - p)^{M-i}$, is the probability of $i$ arrivals in one clock cycle.

We now define two output rates: 1) the output rate per output link, $O_L$ and 2) the output rate per output port, $O_p$. Our interest in $O_L$ is due to the fact that the input load
Chapter 4. Performance of The Fat-Banyan Switch

to the next stage is equal to the link rate \( O_l \) of the previous stage output. Thus, we have

\[
O_l = \frac{E}{(D \times L)}
\]  

(4.4)

\[
O_p = \frac{E}{D}.
\]

However, for the output of the last stage, we are interested in the output rate \( O_p \) (per port). This is consistent with our initial goal of replacing buffers with ports of multiple links. The output rate of the last stage can be calculated recursively using the above formulas. For a FAB switch of size \( N \times N \) the normalized throughput

\[
R(p) = \frac{O_{pn}}{p}
\]

(4.5)

where \( O_{pn} \) is the output rate per output port of the final stage.

4.3. Cell Loss and Stage Dilation

The blocking probability of dilated banyan networks under permutation and random request assumptions have been analyzed [19, 48] for the dilated banyan based interconnection networks. Also, the blocking probability of a dilated switching element under random uniform cell based traffic has been discussed in [20]. In this section we apply it to the case of a FSE. The problem of interest is to determine the cell loss rate as a function of stage dilation. Under the independent uniform traffic pattern, all the FSE’s of a stage are equally loaded. Hence, the average loss rate is the same for all FSE’s in the same stage. The following analysis is therefore for a single FSE \( S(M, D, L) \) only and is applicable to all FSEs in the same stage. The arrival rate at an input link is \( p \). In order to simplify the analysis, we consider the joint probability of a cell arriving at an input being bound to any output. The joint probability is given by \( p/D \) since the cell is equally likely to be destined to any output. The probability that \( j \) out of \( M \) packets will be destined to a particular output is given by

\[
p_j = \binom{M}{j} (p/D)^j (1 - p/D)^{M-j}
\]

(4.6)
Chapter 4. Performance of The Fat-Banyan Switch

The probability of cell loss for an FSE is given as the ratio of the average number of cells lost at an output of an FSE to the average number of cells at an output of an FSE if there were no cell loss. The probability of cell loss for an FSE is

\[
Pr(\text{cell loss}) = \frac{(D/pM)}{\sum_{k=1}^{M-L} k \left( \frac{M}{L+k} \right) (p/D)^{L+k} (1 - p/D)^{M-(L+k)}}
\]

(4.7)

The cell loss probability is plotted as a function of the dilation \( L \) in Figure 4.2, for different input traffic rates and for a FSE \( S(8,2,L) \). It can be seen that the dilation required to achieve a certain cell loss rate is a function of the input traffic rate. For input rates greater than 0.4, a dilation of 8 is necessary to achieve a cell loss probability under \( 10^{-6} \). At traffic rates of 0.2 and 0.3 a dilation of 7 is sufficient. The dilation required at any stage can thus be determined as a function of the traffic rate at the inputs of the FSE’s in that stage. In the fat-banyan, under uniform traffic assumption, the load is maximum at the inputs and reduces as we move towards the output. By keeping the initial dilation high it is possible to reduce the traffic rate quickly and thereafter maintain the dilation constant.

![Figure 4.2. Cell loss probability as a function of dilation for a FSE, S(8,2,L) for different input loads under uniform traffic (by analysis).](image-url)
4.4. Results

The simulation results presented here are based on the following arbitration mechanism at an FSE. Arbitration is required when the number of cells contending for an output of an FSE exceeds the output dilation of the FSE. Arbitration is done by giving priority to the cells from top to bottom at the output concentrators of the FSE. This arbitration mechanism is similar to the knockout switch concentrators [45] where priority is given from left to right of the concentrator inputs. Alternatively cells could be chosen randomly at the concentrator, but this would require a random number generator for each concentrator which would add to the complexity of the FSE. Hence, we have chosen this simple arbitration mechanism. The simulation results are obtained with 95% confidence intervals using the method of independent replications (refer to [49] for a discussion on this method).

4.4.1 Uniform Traffic

The performance results of the FAB switch under uniform traffic are discussed in this section. In Figure 4.3, the maximum achievable throughput is shown as a function of the number of stages, where each FSE has 2 input and 2 output ports (i.e., \( D=2 \)), under independent uniform traffic pattern for the simple banyan network and the FAB switch. For the FAB switch we have used \( M_1=D_1=L_1=2 \) for the first stage; \( M_i=D_{i-1} \times L_{i-1}, D_i=2 \) and \( L_i=L_{i-1}+1 \) (\( 2 \leq i \leq \log_2 N \)) for the subsequent stages (i.e., \( DC=[L_1, L_1+1, L_1+2, \ldots, L_1+\log_2 N-1] \)). In Figure 4.4, the throughput of the FAB switch is plotted on a different scale to show more clearly the variation of the maximum throughput as a function of the number of stages. It should be noted that this throughput is only for illustrating the improvement in throughput obtained by gradual dilation. Further, improvement in throughput and cell loss probability is achieved by using an appropriate dilation configuration as discussed below.
Chapter 4. Performance of The Fat-Banyan Switch

Figure 4.3. Maximum throughput for a fat-banyan switch with $DC=[L_1=2, L_i=L_{i-1}+1, 2\leq i \leq \log N]$ and simple banyan network under the independent uniform traffic pattern with input load $p=1.0$ (by analysis).

Figure 4.4. Maximum throughput for a fat-banyan switch with $DC=[L_1=2; L_i=L_{i-1}+1, 2\leq i \leq \log N]$ under independent uniform traffic pattern (expanded scale) with input load $p=1$ (by analysis).

The high throughput of the FAB is due to the gradual increase in the internal
bandwidth. This causes the load on the FSE's to gradually reduce as we move downstream from the input stage towards the output stage. For a load of $p=1$ at the inputs (i.e., the first stage), the load on the inputs of the FSE's in the second stage is only 0.5. This gradual reduction in load ensures that the FSE's in the intermediate stages deliver a high local throughput at each stage. Since the load on the FSE's decreases as we move towards the output stage, it is unnecessary to keep increasing the degree of dilation of the FSE's to the final stage.

![Graph](image)

Figure 4.5. Maximum throughput of a fat-banyan switch with different dilation configurations under independent uniform traffic with input load $p=1$ (by analysis).

We have investigated the effect of increasing the dilation up to a certain stage, and from there on maintaining the number of input and output links constant. Figure 4.5 plots cell loss probability versus switch size for three dilation configurations. We see that for the configuration with a dilation of 8 from the third stage onwards, the cell loss probability is of the order of $10^{-6}$ even for a switch size of 4096. For the configuration with a dilation of 12 from the fourth stage onwards, the cell loss is of the order of $10^{-11}$. This result is comparable to the result obtained by a non-blocking switch like the knockout. Figure 4.6 illustrates the concentrator action of a section of the FAB switch.
Figure 4.6. Concentrator action of a section of the FAB switch.

For the Knockout switch it has been shown that under independent uniform traffic, it is statistically unlikely that more than a specified number of packets, say \( L \), will be destined to the same output simultaneously. For achieving a loss probability of \( 10^{-6} \) the required concentrator output size \( L = 8 \), independent of the switch size \( N \). Thus, the FAB switch achieves a comparable performance to the Knockout switch, but using the banyan structure with significantly lower hardware complexity.

Figures 4.7 and 4.8 show simulation results for the 8×8 FAB switch with two different dilation configurations. With dilation configuration \( \text{DC}=[2,3,4] \) the cell loss probability is of the order of \( 10^{-4} \) at full load (\( p=1 \)). With dilation configuration \( \text{DC}=[2,4,6] \) the cell loss is of the order of \( 10^{-6} \) at full load. Figure 4.9 is a plot of cell loss probability versus the switch output dilation for two different dilation configurations of a 16×16 FAB switch. For configuration 1 (\( \text{DC}=[2,3,4,\text{variable}] \)) the cell loss probability is of the order of \( 10^{-2} \), and for configuration 2 (\( \text{DC}=[2,4,6,\text{variable}] \)) cell loss probability of the order of \( 10^{-6} \) can be achieved with a dilation of 8. For both switch configurations a dilation of greater than 8 does not result in any significant improvement in cell loss probability. Hence, for uniform traffic conditions a dilation of 8 serves to keep the cell loss probability within practical limits.
Figure 4.7. Maximum throughput (by simulation) as a function of input load for a $8 \times 8$ FAB switch with two different dilation configurations under independent uniform traffic.

Figure 4.8. Cell loss probability (by simulation) as a function of input load for a $8 \times 8$ FAB switch with two different dilation configurations under independent uniform traffic.

4.4.2 Non-Uniform Traffic

It is useful to study the performance of the FAB switch under certain non-uniform traffic patterns which occur in practice. By non-uniform traffic we mean that a cell is likely to require an output address according to a non-uniform distribution. We first
Figure 4.9. Cell loss probability as a function of output dilation for a 16×16 FAB switch for two different stage dilation configurations under independent uniform traffic with input load p=0.9 (by simulation).

Figure 4.10. An example of a maximum conflict pattern in a 16×16 banyan network consider single-source to single-destination (SSSD) non-uniform traffic pattern [23] in which each input source sends all its cells to a single destination. This type of traffic is also called as point-to-point traffic. An example of a point-to-point traffic is broadband traffic involving a large file transfer or an image transfer.

The performance of an 8×8 single-buffered banyan network under point-to-point traffic [7, 23] has shown that for the maximum conflict pattern (This traffic pattern has
the maximum conflicts among all possible traffic patterns consisting of \( N \) (for a switch of size \( N \times N \)) point to point traffic paths involving distinct inputs and outputs) shown in Figure 4.10 the maximum throughput that can be achieved is 0.49. For an \( 8 \times 8 \) multi-buffered banyan network the maximum throughput for the maximum conflict pattern is only slightly greater than 0.5. This low throughput in a buffered banyan network can be attributed to the conflicts arising in the innermost shared links. However, for an \( 8 \times 8 \) FAB switch with dilation configuration \( DC=[2,3,4] \) (or an equivalent dilated banyan switch) there is absolutely no conflict within the switch fabric as shown in Figure 4.13 and a throughput of 1 is achieved. This is because the FAB switch has been designed to reduce internal conflicts by increasing the internal bandwidth of the banyan network.

A more general case of non-uniform traffic case would be a few SSSD paths embedded in a uniform traffic pattern. Here we consider the case of a single SSSD path embedded in a uniform traffic pattern [23] which could be used to model a single dedicated video channel with several voice or data channels. Such a traffic pattern would be represented as a load matrix shown below in which \( \lambda \) represents the fraction of uniform traffic directed to the non-SSSD outputs and \( \lambda_{sssd} \) represents the point-to-point dedicated traffic between an input and the corresponding output.

\[
\begin{bmatrix}
\lambda & 0 & \lambda & \ldots & \lambda \\
0 & \lambda_{sssd} & 0 & \ldots & 0 \\
\lambda & 0 & \lambda & \ldots & \lambda \\
\ldots & \ldots & \ldots & \ldots & \ldots \\
\lambda & 0 & \lambda & \ldots & \lambda
\end{bmatrix}
\]  

(4.8)

Simulation results obtained for an \( 8 \times 8 \) FAB switch with dilation configuration: \( DC=[2,3,4] \) and \( DC=[2,4,6] \) for the above traffic pattern are shown in Figures 4.11 and 4.12. The results indicate that as the throughput of the embedded SSSD path increases, the throughput of the background uniform traffic decreases and hence the cell loss probability increases. The cell loss probability is of the order of \( 10^{-4} \) for dilation configuration 1. The cell loss probability can be improved by increasing the dilation of the FSE's. The cell loss probability is improved to the order of \( 10^{-7} \) as shown in Figure
4.12 for a 8×8 FAB switch with dilation configuration 2. Therefore, with a small increase in dilation the cell loss probability is greatly improved for this non-uniform pattern.

Figure 4.11. Maximum throughput (by simulation) for a 8×8 FAB switch under mixed SSSD (point-to-point traffic) and background uniform traffic load of 1.0 for two different dilation configurations.

Figure 4.12. Cell loss probability results (by simulation) for a 8×8 FAB switch under mixed SSSD (point-to-point traffic) and background uniform traffic load of 1.0 for two different dilation configurations.
Chapter 4. Performance of The Fat-Banyan Switch

A more severe case of non-uniform traffic is the hot-spot pattern in which many input sources try to send their cells to one destination. An example of such a situation is when several callers try to call a popular location on a telephone network. Another example is a LAN (local area network) consisting of many diskless computers and a single file server. When several computers simultaneously try to access the file server a hot-spot situation results. The hot-spot traffic is modelled as follows. A fraction \( h \) of cells arriving at an input are directed towards the hot-spot. If \( p \) is the arrival rate at an input of the FAB switch then \( hp \) cells are directed to the hot-spot and \( (1-h)p \) cells are directed uniformly to the \( N \) outputs. Table 4.1 shows simulation results for the \( 8 \times 8 \) FAB switch with dilation configuration: DC=[2,3,4] under hot-spot conditions. It can be seen that the effect on cell loss is severe even at low loads. This effect on cell loss can be alleviated considerably by increasing the dilation of the FSE’s in the FAB switch as illustrated in Table 4.2. However, the hot spot pattern is a problem with any switch fabric including the crossbar based Knockout switch. The solution to handling hot-spot traffic lies in limiting its intensity.

![Figure 4.13. Routing of a Maximum Conflict Pattern in an 8×8 FAB switch with the following dilation configuration: DC=[2,3,4]](image-url)
Table 4.1. Maximum throughput and cell loss probability results (by simulation) for an 8×8 FAB switch with dilation configuration as follows: DC=[2,3,4] under hot-spot traffic.

<table>
<thead>
<tr>
<th>Hot-spot percentage (h)</th>
<th>Throughput</th>
<th>Cell loss probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0125</td>
<td>0.99944991</td>
<td>5.500912e-04</td>
</tr>
<tr>
<td>0.025</td>
<td>0.99944088</td>
<td>5.5912e-04</td>
</tr>
<tr>
<td>0.05</td>
<td>0.99938337</td>
<td>6.166312e-04</td>
</tr>
<tr>
<td>0.1</td>
<td>0.99905955</td>
<td>9.404475e-04</td>
</tr>
</tbody>
</table>

Table 4.2. Maximum throughput and cell loss probability results (by simulation) for an 8×8 FAB switch with dilation configuration as follows: DC=[2,4,6] under hot-spot traffic.

<table>
<thead>
<tr>
<th>Hot-spot percentage (h)</th>
<th>Throughput</th>
<th>Cell loss probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0125</td>
<td>0.99999919</td>
<td>8.1375e-07</td>
</tr>
<tr>
<td>0.025</td>
<td>0.99999917</td>
<td>8.3375e-07</td>
</tr>
<tr>
<td>0.05</td>
<td>0.99999897</td>
<td>1.0275e-06</td>
</tr>
<tr>
<td>0.1</td>
<td>0.99999743</td>
<td>2.56875e-06</td>
</tr>
</tbody>
</table>

Table 4.1. Maximum throughput and cell loss probability results (by simulation) for an 8×8 FAB switch with dilation configuration as follows: DC=[2,3,4] under hot-spot traffic.

Table 4.2. Maximum throughput and cell loss probability results (by simulation) for an 8×8 FAB switch with dilation configuration as follows: DC=[2,4,6] under hot-spot traffic.
4.5. Performance of the Output Buffered Fat-Banyan

In this section the performance of the output buffered switch is assessed by analysis and simulation. The simulation results are obtained with 95% confidence intervals using the method of independent replications (refer to [49] for a discussion on this method).

4.5.1 Uniform Traffic

The FAB switch can be implemented as a pure output-buffered switch. The analysis of a non-blocking output queued switch under independent uniform traffic has been dealt with in [50]. Here we use the analysis for the FAB switch and validate the simulation results against the analytical results. Two parameters are required in determining the relation between cell loss and buffer size, namely the arrival rate and the buffer size. The arrival rate at an input link of an FSE at a particular stage of the FAB switch is the output link rate, $O_1$ from the previous stage. For finite number of output links per port, $L$ and finite buffer size, $b$, the output queue can be modelled as a finite-state, discrete-time Markov chain. Let $Q_m$ denote the number of cells in an output queue of the FAB switch at the end of the $m$th time slot, and $A_m$ denote the number of packet arrivals during the $m$th time slot. Then, we have

$$Q_m = \min\{\max(0, Q_{m-1} + A_m - 1), b\}. \quad (4.9)$$

The max term inside the bracket ensures that the queue size is non-negative and the overall min term ensures that the queue size can never be greater than $b$. Cells are lost if queue size exceeds $b$. Defining the random variable $A$ as the number of cell arrivals at an output port of the FAB switch in a given time slot, we have

$$a_k = \Pr[A = k] = \binom{L}{k} (O_1)^k (1 - O_1)^{N-k}. \quad (4.10)$$
Chapter 4. Performance of The Fat-Banyan Switch

The state transition probabilities \( P_{ij} = \Pr[Q_m = j \mid Q_{m-1} = i] \) are given by:

\[
P_{ij} = a_0 + a_1 \quad i = 0, j = 0
\]
\[
= a_0 \quad 1 \leq i \leq b, j = i - 1
\]
\[
= a_{j-i+1} \quad 1 \leq j \leq b - 1, 0 \leq i \leq j
\]
\[
= \sum_{m=j-i+1}^{j} a_m \quad j = b, 0 \leq i \leq j
\]
\[
= 0 \quad \text{otherwise.}
\] (4.11)

The steady-state queue size can be obtained from the Markov chain global balance equations. We get

\[
q_1 = \Pr[Q = 1] = (1 - a_0 - a_1)q_0/a_0
\]
\[
q_n = \Pr[Q = n] =
\]
\[
\left( q_{n-1} \left[ 1 - P_{(n-1)(n-1)} \right] - \sum_{k=0}^{n-2} q_k P_{k(n-1)} \right) / P_{n(n-1)}
\]
\[
2 \leq n \leq b
\] (4.12)
\[
q_0 = \Pr[Q = 0] =
\]
\[
1 - \sum_{n=1}^{b} q_n = 1/\left( 1 + \sum_{n=1}^{b} q_n/q_0 \right).
\]

In order to determine the cell loss rate introduced by the buffers we need to determine the switch throughput at the buffer outputs. A cell is not transmitted in the \( m \)th time slot from an output buffer if and only if \( Q_{m-1} = 0 \) and \( A_m = 0 \). Denoting \( \rho_0 \) as the normalized switch throughput obtained at the buffer outputs, we have

\[
\rho_0 = 1 - q_0 a_0. \quad (4.13)
\]

A cell emerging from the FAB switch will be lost if the output queue already contains \( b \) packets. The buffer cell success probability is obtained by dividing the normalized switch throughput at the buffer outputs by the output port rate of the FAB switch. Hence, the cell loss probability is given by

\[
\Pr[\text{cell loss}] = 1 - \rho_0 / \rho_n. \quad (4.14)
\]
Figure 4.14. Cell loss probability for a 64×64 FAB switch with DC=[2,4,8,8,8] as a function of output buffer size under independent uniform traffic with input load \( p = 0.9 \).

Figure 4.14 is a plot of cell loss which includes the buffer loss for DC=[2,4,8,8,8] and input load \( p = 0.9 \). The analysis results are optimistic compared to the simulation results, but provide a good approximation for small buffer sizes.

Figure 4.15 is a plot of cell loss probability (concentrator loss+buffer loss) as a function of buffer size for three different dilation configurations for a 16×16 FAB switch. The concentrator loss is dependent on the dilation configurations. The concentrator losses for each of the configurations are shown alongside the curves. The concentrator loss is the highest for dilation configuration 1 (DC=[2,3,4,8]). For dilation configuration 3 (DC=[2,4,8,10]) the concentrator loss is less than \( 10^{-6} \) for a load of 0.9 under independent uniform traffic, and the loss curve tracks that for a lossless switch fabric such as the Knockout. The cell loss probability curves flatten out after a buffer size of 70 for dilation configurations 1 and 2 (DC=[2,4,6,8]), since the buffer loss is negligible after that, and throughput is limited by the concentrator loss which is determined by the dilation configuration.
Figure 4.15. Cell loss probability (concentrator loss+buffer loss) for three different dilation configurations of a 16×16 FAB switch as a function of output buffer size under independent uniform traffic with input load p=0.9 (by simulation).

4.5.2 Bursty Traffic

A bursty traffic model in which each source alternates between active and idle periods is used for the performance evaluation. The durations of the active and idle periods are both geometrically distributed with parameters α and β respectively. In the active period it is assumed that cells arrive in consecutive time slots. The burst lengths are assumed to be statistically independent. The mean burst length is $L_{\text{active}}=1/\alpha$, the mean idle duration is $L_{\text{idle}}=1/\beta$ and the offered load $\rho=L_{\text{active}}/(L_{\text{active}}+L_{\text{idle}})$. For our simulations we have assumed $L_{\text{active}}=15$ cells.

Figures 4.16 and 4.17 plot the cell loss versus output buffer size for two different loads, for two dilation configurations of a 64×64 FAB switch. For the switch with DC=[2,4,4,4,4,4] the cell loss probability is high due to the concentrator loss. With DC=[2,4,8,8,8,8] the concentrator loss is negligible and is dominated by the buffer loss and the overall cell loss probability improves as the output buffer size increases. As in the case of the random traffic the dilation configuration DC=[2,4,8,...] which has a
maximum dilation of 8 gives the best concentrator loss. Further, improvement in cell loss probability is achieved by using larger buffer sizes. Truncated architectures with shared memory can be used to reduce the memory size required to achieve a given cell loss probability. This is the topic of the next chapter.

Figure 4.16. Cell loss probability (concentrator loss+buffer loss) for two different dilation configurations of a 64×64 FAB switch as a function of output buffer size under bursty traffic conditions with $\rho=0.7$ and average burst length=15 (by simulation).
Figure 4.17. Cell loss probability (concentrator loss+buffer loss) for two different dilation configurations of a 64×64 FAB switch as a function of output buffer size under bursty traffic conditions with $\rho=0.9$ and average burst length=15 (by simulation).
4.6. The Input-Output Queued Fat-Banyan

Input queueing in ATM switches is easy to implement, however, it suffers from serious throughput degradation due to the head-of-line (HOL) blocking effect in FIFO buffers. Throughput can be enhanced if buffering schemes other than FIFO are employed. Recently, a number of cell scheduling algorithms for input queuing were proposed to overcome the HOL blocking effect [51, 52, 53]. These algorithms are iterative and require the availability of fast scheduling hardware and the use of non FIFO buffers. Totally eliminating HOL blocking requires each input port to allocate a distinct buffer for each output port, a technique dubbed VOQ (virtual output queueing). Obviously, the implementation of such a scheme can be very costly [2, 3]. In this section, we examine the performance of input-output queueing in the FAB switch as a means for overcoming the HOL blocking effect. In the FAB switch with input-output queueing we use simple FIFO buffers for the input queues and multiport FIFO buffers for the output queues. We use the notation $Bin$ for input buffer size per port and $Bout$ for output buffer size per port. Back pressure can be employed to enforce no cell loss in the memoryless fabric and/or output buffers. The backpressure mode of operation for these switches is discussed next followed by its performance under uniform and bursty traffic conditions. It is shown that the backpressure (along with small input buffers) technique we employ renders the fabric lossless while achieving a better cell loss performance than the output buffered case.

4.6.1 Backpressure mode

Backpressure is a technique which is used to control cell loss within a switching fabric. Backpressure can be used to prevent cell loss in a buffer (or a group of buffers) or to prevent cell loss in the memoryless fabric. The former case is called the buffer-capacity (or BC) backpressure and the latter is called link-capacity (or LC) backpressure. BC back pressure aims at eliminating cell loss in the output buffers. On the other hand, LC back pressure aims at eliminating cell loss in the memoryless interconnection fabric.
Chapter 4. Performance of The Fat-Banyan Switch

In the following we focus on LC back pressure as a method for making the memoryless interconnection network of the FAB lossless. Figure 4.18 shows an example of how LC back-pressure is implemented on an input-output buffered switch. In the example, six HOL cells from input queues 0, 1, 2, 3, 4, and 7, are destined to output port 0, but the output dilation of the last stage is 4. The HOL cell from input queue 3 is blocked by the top FSE at stage 2, and the HOL cell from input queue 7 is blocked by the top FSE at stage 3. Note that an FSE propagates back-pressure signals on a reverse path back to the concerned input queues if it can not route cells on its outgoing links. This type of operation requires simple distributed control that can be easily incorporated in FSE hardware. With LC backpressure, the number of cells, $k$, that are switched to an output port is equal to the dilation of the FSEs in the last stage of the switch.

Figure 4.18. Concept of LC backpressure in input-output queued fat-banyan switch with $N=8$ and dilation of last stage FSE’s = 4.
4.6.2 Uniform traffic

From Figures 4.19 and 4.20 we see that small input buffer sizes (2–3 cells) result in improving the performance of the switch while eliminating the loss in the fabric. We also note that as size of the switch \( N \) increases the cell loss probability increases slightly. Hence, the LC backpressure technique further enhances the capability of the FAB switch. The additional complexity of adding small input buffers and using backpressure results in reducing the dilation within the fabric, a lossless fabric and better cell loss performance than the purely output-buffered case.

4.6.3 Bursty traffic

From Figure 4.21 we see that input buffers have a similar effect to the case of independent uniform traffic. However, larger input buffers in the range of 4–8 cells are required for the bursty traffic case. Finally, in Figure 4.22 we see that the delay performance by adding small input buffers is not affected much when compared to the pure output queueing case (Bin=0).

![Figure 4.19. Cell loss probability for FAB switches of size 32 and 64 as a function of input buffer size per output port under random uniform traffic with offered load \( \rho = 0.7 \) and output buffer size=32 cells (by simulation). For \( N=32 \), DC= [2,4,4,4,4]. For \( N=64 \) DC=[2,4,4,4,4].]
Chapter 4. Performance of The Fat-Banyan Switch

Figure 4.20. Cell loss probability for FAB switches of size 32 and 64 as a function of input buffer size per output port under random uniform traffic with offered load $\rho = 0.9$ and output buffer size=32 cells (by simulation). For $N=32$, $DC=[2,4,4,4,4]$. For $N=64$ $DC=[2,4,4,4,4,4]$.

Figure 4.21. Cell loss probability for FAB switches of size 32 and 64 as a function of input buffer size per input port under bursty uniform traffic with average burst length=15 and offered load $\rho = 0.7$ (by simulation) and $B_{out}=256$. For $N=32$, $DC=[2,4,4,4,4]$. For $N=64$ $DC=[2,4,4,4,4,4]$.
Figure 4.22. Average delay versus offered load ($\rho$) for FAB switch of size 32 under bursty uniform traffic with average burst length=15. DC=[2,4,4,4,4], Bout=256 (by simulation).
4.7. Conclusions

The performance of the FAB switch has been shown to achieve very low cell loss probability (of the order of $10^{-6}$ or lesser). The cell loss is easily controlled by choosing an appropriate dilation configuration. For the purely output buffered switch (without any input buffers) $DC=[2,4,8,...]$ gives a cell loss probability of the order of $10^{-6}$ under independent uniform traffic conditions for switch sizes as large as $4096 \times 4096$. For the bursty traffic case with uniform output destinations the switch with $DC=[2,4,8,...]$ gives better performance as in the previous case. However, larger output buffers are required and the cell loss is very much higher at high input loads. Another version of the FAB switch with both input and output buffers was also considered. This version employs a backpressure technique termed LC backpressure to make the fabric lossless. This has the advantage of reduced dilation, a lossless fabric and better cell loss performance than that of the purely output-buffered case. Also, we have shown by simulation that the delay performance is not affected by using input buffers and employing backpressure because the input buffers are very small in size.

We also considered some examples of the non-uniform traffic case. For the case of SSSD traffic with independent uniform background traffic we have shown that a good cell loss performance can be obtained with increased dilation. The extent of improvement in performance depends on the level of dilation that we are willing to accomodate before the switch becomes very expensive. With hot-spot traffic the overall cell loss probability for a given hot-spot intensity is improved by using a slightly higher dilation.
Chapter 5  Design of Scalable Shared Memory Switches

5.1. Introduction

The design of ATM switches has evolved in a number of ways over the past few years, but several key issues remain to be resolved. One major issue is scalability and modularity, because many solutions which are efficient for small size switches are not scalable to large switches. Output buffered switches, such as the Knockout switch [54] provide optimal delay-throughput performance, however, they may require excessive buffering resources to handle general types of input traffic and their expandability to larger size is questionable due to their high interconnection complexity. Both routing and buffering resources would be severely under-utilized when the cell traffic distribution at the output ports is unbalanced. In this case, cell loss in heavily loaded output buffers will be high even if sufficient buffer space is available at other output ports [55, 56]. The "growable" ATM switch architecture proposed in [11] is a generalization of the Knockout switch that uses shared memory buffers for groups of outputs, instead of buffers dedicated to each output. This concept has two major advantages. The first is that output buffer sharing results in smaller overall buffer memory size than the dedicated output buffer case for a given cell loss rate. The second, and more important advantage, is that the size of each shared memory module remains fixed for a given cell loss rate (under the uniform input traffic assumption) for any switch size, thus providing a truly scalable switch architecture.

In this chapter we propose and evaluate a scalable ATM switch architecture based on the FAB switch which employs shared-memory buffers as basic building blocks. The proposed architecture is completely parameterized in terms of the shared-memory size and the number of ports sharing a memory buffer. Thus, at one extreme we can have one (output) port per memory module, which is the case when no buffer sharing is employed.
Chapter 5. Design of Scalable Shared Memory Switches

At the other extreme, all ports share the same memory module, which is the case of a pure shared memory switch. Thus our techniques provide a unified methodology for designing and evaluating a wide range of ATM switches ranging from space-division switches with no buffer sharing to shared-memory switches. We evaluate the performance trade-offs resulting from placing shared memory buffers at the outputs of a switch. The main motivation for a shared memory design strategy is two fold:

1. By taking advantage of well developed and commercially available shared memory switching (SMS) technology, ATM switch design can be simplified to determining a suitable SMS module size, and identifying a proper interconnection among the modules. In this way, switch architectures can be reusable and able to evolve as memory technology advances.

2. Employing shared memory greatly enhances buffer space utilization, allows the implementation of weighted fairness for multiple services, and supports multipriority traffic efficiently. The simplest form of complete memory sharing suffers from unfairness caused by unbalanced logical queue length in the shared memory. In such a scenario, certain output ports of the SM (shared memory) switch will have queues that consume most of the memory space, thus causing heavy cell loss to cells destined to other ports. This problem can be solved by a number of buffer management schemes [8, 9]. A buffer management scheme is needed only when the SM is full (or over a given threshold). A particularly efficient scheme is push-out buffer management policy[8]. When the SM is full, the push-out scheme will select for replacement the cell at the tail of the longest queue. Thus the push-out policy guarantees fairness to all output ports. The scheme can be implemented by associating a length counter with each logical queue such that whenever a cell enters (or leaves) the SM, the counter associated with its logical queue is incremented (decremented). The push-out policy can then identify the longest queue by a suitable
fast method that inspects the counters only (for example, through fast bus arbitration).

There are several other methods for buffer management[8, 9] but they all attempt to approximate the push-out scheme.

In the next section we discuss the key design issues to be considered in designing ATM switches. In section 5.3 we introduce the truncated switch model and its realization using the truncated fat-banyan switch. This is followed by a discussion of the performance analysis of the truncated fat-banyan switch. Finally, we consider the input-output queued truncated fat-banyan and study its performance.

5.2. Considerations in Designing ATM Switches

Buffering is the resource that dominates both the complexity and cost of an ATM switch. This is particularly true for switches that will collaborate with network-wide congestion control schemes with or without feedback congestion control. The recently standardized rate-based congestion control scheme requires the exchange of congestion information between the source and destination for each ABR connection. In this scheme, based on congestion information (propagating from source to destination), the destination sends special resource management (or RM) cells back to the source informing it as to whether it must increase or decrease its cell rate. The relatively long delays incurred in this type of congestion control implies that ATM switches must have sufficiently large buffers to prevent excessive cell loss for ABR connections especially during initial bursts. In the steady state, explicit-rate algorithms are known to require small buffer sizes. Traffic management schemes that do not use feedback information demand large buffer resources. Recent experiments for implementing TCP/IP over ATM with partial or early packet discard show that a buffer space of few thousand cells [57, 58] may be required to provide an acceptable effective throughput for TCP connections. In comparison to buffers, wires and memoryless interconnection stages pose much less problems. Therefore, our
approach is based on buffer optimization in ATM switch design. This involves several key design considerations:

1. The number of buffer stages should be minimized to reduce cell delay. If this cannot be achieved then cut-through routing should be employed. This type of routing allows cells to “bypass” empty buffers.

2. Buffer sharing should be utilized as much as possible, to reduce the amount of total buffering, to allow efficient control policies to be applied to multiple connections sharing a buffer, and to take advantage of available memory technology.

3. Use effective control and scheduling techniques for buffer management. This is justified by the fact that since large buffers will dominate the cost of a switch, then it is worthwhile to employ complex controllers that enhance the performance and utilization of such buffers for delay-sensitive and loss-sensitive services.

4. Interconnection fabrics must be self-routing and almost non-blocking even in the presence of unbalanced traffic. This feature is needed to restrict cell loss to buffer overflow only and not because of interconnection concentration loss.

5.3. Truncated Switch Architectures

In output buffered switches, incoming cells are routed to the output buffers through some interconnection network. If this network is non-blocking then it can be shown that such switches provide optimal delay-throughput performance at the expense of using larger buffers than other schemes. However, since most practical (cost-effective) fabrics are generally blocking, part of the cell loss is expected to occur in the bufferless interconnection network (this will be referred to as concentration loss). The remaining part of cell loss will occur in the output buffers. The main drawback of this buffering scheme is that cell loss can occur in certain overloaded buffers even if plenty of buffer space is available at other outputs. As mentioned earlier, this problem can be alleviated by bundling outputs into distinct groups and assigning a shared memory switch to buffer the
cells for each group of outputs as well as perform the final routing to individual output ports. In [38], we proposed the class of truncated switch fabrics as a very effective implementation for ATM switches.

5.3.1 Generalized Truncated Switch Model

An $N \times N$ truncated switch model can be represented by the general model shown in Figure 5.1. The term “truncated” refers to the fact that the bufferless interconnection stage is derived from standard multistage networks through removal of the last few stages of switching elements (SE’s). The truncated output-buffered switch model consists of two major stages, the first stage represents a truncated self-routing interconnect and the second stage consists of shared-memory switching modules. The first stage mainly concentrates traffic from the $N$ input links into $N$ multiple streams each of bandwidth $N''$ (links). Each concentrated stream (of $N''$ links) from the first stage is fed directly to an $N'' \times N'$ shared-memory switch. Note that switch models other than shared-memory can be used to realize the $N'' \times N'$ switching task. However, the shared memory switches will reduce the buffer-size requirements of the last stage, and consequently of the entire switch.

![Diagram of the general truncated switch model](image_url)

Figure 5.1. The general truncated switch model
Chapter 5. Design of Scalable Shared Memory Switches

The destination address field normally used in banyan type networks can be divided into two subfields: the higher-order subfield contains the address of the output SM switch (SMA) and the lower order subfield contains the output port address (SPA) within an SM. The field SMA is used to route cells up to one of the switches, then an SM switch uses the address field SPA to route cells to their final output destinations. The truncated switch model is based on the growable switch model proposed in [11]. The growable switch model generalizes the Knockout principle so that it applies for a group of outputs instead of a single output. If the arrivals on different input lines are statistically independent, then the probability of more than a handful of simultaneous packets say $N''$ arriving at a group of $N'$ outputs where $N' \leq N''$ is exceedingly small even for arbitrarily large $N$. For example, $10^{-6}$ cell loss can be achieved with $N''=33$ and $N'=16$ for an independent uniform load of 0.9 and arbitrarily large $N$. The growable switch model has been realized using a Clos network topology and is shown in Figure 5.2. The first two stages of the Clos network are used as the memoryless interconnect fabric and the last stage is realized by shared memory switching modules. Two types of losses occur in the interconnect fabric. The first is the loss due to the concentrator action of the fabric when more than $m$ cells are destined to the same output module. The second type of loss called the scheduling loss is due to the use of a distributed but not truly self-routing algorithm within the interconnect fabric. Optimal centralized routing algorithms are known for the Clos network [59] but they are slow and unsuitable for fast packet switching.

5.3.2 Banyan Realization

The scheduling loss suffered in the interconnect fabric of the growable switch can be entirely avoided by using a true self-routing interconnect such as a banyan network. However, the standard banyan is known to have a severely blocking topology. The Truncated Fat-Banyan (TFAB) switch which is introduced in this section employs self-routing through the fabric and hence avoids this loss.
The TFAB topology is derived from the FAB topology by removing one or more switching stages starting from the last (or output) stage and replacing the output buffers by shared-memory switches. Thus the TFAB provides an efficient realization of the truncated switch model since it achieves the same cell loss performance but with less hardware and much less wiring. The number of stages in the TFAB switch can vary from 1 to $\log N - 1$ for an $N \times N$ switch fabric. The notation TFAB$(N,S-1)$ will be used to denote an $N \times N$ TFAB switch with $S$ stages, the $S$th stage being the shared memory stage. The lesser the number of stages the more is the truncation. If the number of stages is $\log N - 1$ then the last stage is truncated; if the number of stages is $\log N - 2$ then the last two stages are truncated, and so on. Figure 5.3 illustrates a TFAB$(8,2)$ with dilation configuration DC=[2,3]. The routing in the truncated fabric is similar to the normal self-routing algorithm except that only those destination bits required for routing through the truncated fabric are used and the rest of the bits are used by the shared memory modules which provide the buffering and final routing. The TFAB switch has the following features:
1. **Low cost**: One advantage of the TFAB interconnection network over other fabrics (such as the knockout or crossbar based switches) is its cost-effectiveness (according to the cost function used in chapter 3). In the TFAB network, cell concentration functions are distributed among the switching elements and are, thus, shared among the switch outputs. Additionally, the TFAB network has the advantage of using the same simple self-routing algorithm used with banyan network topologies.

2. **QoS separation**: The FSE’s in the FAB are essentially bufferless, and implement some type of wormhole routing (as opposed to a store-and-forward routing) that maintains cell sequence integrity. More importantly, flows from different input ports do not affect each other’s delays. Although flows from different input ports can affect each other’s cell loss, the amount of loss can be reduced to arbitrarily small levels through a proper setting of link dilations. Such is not the case for example with other multipath networks, such as the buffered Benes [1] in which different flows can affect each other’s cell delay as well as cell loss. QoS separation is very important not only for achieving fairness, but also for providing QoS guarantees to the various flows crossing the switch. The TFAB employs shared memory switches in the last stage. In the shared memory switch logical queues are maintained for the output ports they serve. The logical output queues avoid QoS coupling in the form of delay interactions between the cell streams for different output ports. Further, unfairness in the buffer occupancy of any queue (and hence buffer loss) can be avoided by using a buffer management strategy [8, 9].

3. **Multicasting capabilities**: The TFAB switch has excellent multicasting capability. The internal dilation allows the bufferless TFAB network to maintain 100% throughput even when the multicast traffic load offered to the output port exceeds unity. Simulation results reported in [60] for the FAB switch show that, under random uniform traffic with fanout (number of packet copies) given by a truncated geometric
distribution, high throughput can be maintained even when the instantaneous output offered load is increased to 200%. Of course large output buffers are needed under such conditions. For multicast connections, the FAB network can be viewed as composed of multiple multicast trees which share their roots and some of the early links. However, the tree paths become distinct as one moves towards the output ports. The TFAB switch can implement multicasting using a two-phase procedure. In the first phase, multicasting is achieved by splitting trees routed at the input ports and performing cell replications at the appropriate internal FSE’s. Because of internal dilation, a single FSE can be a member of multiple multicast trees without any conflicts. In the second phase, one of several shared-memory multicasting schemes, such as those reported in [61] can be employed.

![Figure 5.3. Truncated fat-banyan (TFAB(8,2)) switch with output shared memory modules and DC=[2,3).](image-url)
5.4. Performance Analysis of the Truncated Fat-Banyan

In the following, simulation results are presented to demonstrate the efficiency of TFAB switches under various input traffic loads and conditions. The simulation results are obtained by the method of independent replications (refer to [49] for a discussion on this method) using 95% confidence intervals. The confidence intervals are very tight and not shown in the figures.

5.4.1 Uniform Traffic

We consider a TFAB(16,3), i.e., a FAB switch with its last stage truncated. Figure 5.4 illustrates the performance curves of cell loss probability as a function of the buffer size per output port for the three different dilation configurations considered previously. We can see that the buffer size per output port is very much reduced in comparison with pure output buffering. With pure output buffering, a buffer size of approximately 57 per output (refer to Figure 4.15) is required to achieve a cell loss probability of $10^{-6}$. However, for the TFAB(16,3) switch the buffer size is about 30 per output, which is about half the requirement for the pure output buffered case. However, the memory speed [5] would be much slower in this case.

5.4.2 Bursty Traffic

This subsection examines the performance of the TFAB switch under bursty traffic conditions. The same traffic model as in chapter 4 is used for the simulations with $L_{active}=15$ cells.

The simulation results are shown in Figures 5.5 to 5.7 for three different bursty traffic loads. For each load two different TFAB switches are considered. TFAB(16,3) is obtained by truncating the last stage of a $16 \times 16$ FAB switch, and using shared-memory modules each of which is shared among two output ports. Hence, a TFAB(16,3) switch requires 8 shared-memory modules. TFAB(16,2) is obtained by truncating the last two stages, and using 4 shared-memory modules, each of which is shared among four output
Figure 5.4. Cell loss probability as a function of output buffer size for a 16×16 TFAB switch for different dilation configurations under independent uniform traffic with input load $p=0.9$ (obtained by simulation).

We see that in all the cases the cell loss probability improves as the number of outputs sharing an output buffer increases. For example, with offered load $\rho=0.5$ and a buffer size of 60, TFAB(16,2) achieves a cell loss probability of $10^{-5}$, whereas with TFAB(16,3) the cell loss probability is only about $10^{-3}$. Hence, by adjusting the degree of truncation we can provide the required sharing of the output buffers and achieve a specified cell loss probability.
Chapter 5. Design of Scalable Shared Memory Switches

Figure 5.5. Cell loss probability (buffer loss) for two $16 \times 16$ TFAB switches with no concentrator loss as a function of buffer size per output port under bursty traffic with average burst length=15 and offered load $\rho=0.5$ (by simulation).

Figure 5.6. Cell loss probability (buffer loss) for two $16 \times 16$ TFAB switches with no concentrator loss as a function of buffer size per output port under bursty traffic with average burst length=15 and offered load $\rho=0.7$ (by simulation).
Figure 5.7. Cell loss probability (buffer loss) for two 16×16 TFAB switches with no concentrator loss as a function of buffer size per output port under bursty traffic with average burst length=15 and offered load $\rho=0.9$ (by simulation).

5.4.3 Unbalanced Bursty Traffic

In this traffic model the switch is assumed to be loaded with bursty traffic that is balanced at the inputs, each carrying a load of $\rho$. The traffic pattern is unbalanced with respect to the outputs in such a way that the outputs can be divided into two groups named “hot outlet group” and “cold outlet group”. Figure 5.8 shows cell loss probability versus buffer size per port, for the case of two equal sized (8 in this case) destination groups, and the probability of a cell being destined to group 1 is $p_{11}$ and the probability of a cell being destined to group 2 is $p_{12}$. The hot outlet group is more heavily loaded than the cold outlet group and hence suffers from more loss. Also, the cold outlet group benefits more from buffer sharing due to its lighter load than the hot outlet group.
Figure 5.8. Cell loss probability (buffer loss) for a 16×16 TFAB switch for two output group sizes as a function of buffer size per output port under unbalanced bursty traffic with offered load $\rho=0.71$ and average burst length=15 (by simulation).

5.5. Truncated Fat-Banyan with Input Queues

We have seen in chapter 4 that the input-output queued FAB switch with LC (link-capacity) backpressure is used to overcome the loss within the switch fabric. The same technique can be used to enhance the performance of the TFAB switch. Figure 5.9 illustrates the same example as in chapter 4 for the TFAB switch. We note that in this case the HOL cell from input queue 7 is not blocked due to the truncation of the last stage, and provided there is enough space in the shared memory module SM1 it will not be dropped. Figure 5.10 illustrates the performance of a TFAB(64,4) switch (an output grouping of 4 ports per SM). We see that in comparison to the input-output queued FAB switch the cell loss performance is greatly improved. The improvement is in two ways. Firstly, since the fabric is truncated the cell loss in the input queues due to LC backpressure is reduced. Secondly, the buffer loss in the output buffers is reduced due to buffer sharing.
Chapter 5. Design of Scalable Shared Memory Switches

Figure 5.9. Concept of LC backpressure in input-output queued TFAB switch.

Figure 5.10. Cell loss probability for 64×64 TFAB switch with input queueing and LC backpressure under bursty uniform traffic with average burst length=15 and offered load $\rho=0.7$ (by simulation). $B_{out}=256$. 
Chapter 5. Design of Scalable Shared Memory Switches

5.6. Conclusions

This chapter introduced the truncated fat banyan (or TFAB) switch model as a basic distribution network in a growable switch fabric. The TFAB satisfies all the properties of a growable fabric as specified in [11], and has the additional advantage of employing self-routing schemes. As far as the authors know, the TFAB was the first truly growable fabric with self-routing capability. The chapter also introduced the link-capacity back-pressure technique as a means of realizing an economical lossless version of the FAB which employs both input and output buffers. With link-capacity backpressure cell loss can occur only in the input or output buffers.
Chapter 6 Multicasting

6.1. Introduction

Multicasting is an important service to be provided by an ATM broadband network. Many applications like videoconferencing and entertainment video require the setting up of point-to-multipoint connections. An efficient multicasting architecture is hence essential in the ATM switches to setup such point to multipoint connections.

A multicast ATM switch can be built by cascading a copy network with a point-to-point switching network as shown in Figure 6.1. The copy network creates multiple copies of the input packets and delivers them to the inputs of the point-to-point switching network which follows it. The copy network provides a fast mechanism of replicating the packets without worrying about the routing to the output destinations. The point-to-point switching network does the final routing to the appropriate destinations. At the output of the copy network, packets of a particular multicast connection have to be distinguished so that the subsequent point-to-point switch can route them to the appropriate outputs. This is achieved by using an index reference (IR) [7]. The packets of different multicast connections are distinguished by a broadcast channel number (BCN). During call setup, the point-to-point switch output for a particular BCN and IR are stored at each output of the copy network. Efficient ways of reducing the memory requirements for storing this information at each output of the copy network are proposed in [62].

![Figure 6.1. A multicast packet switch consisting of a copy network and a point-to-point switch.](image-url)
This chapter presents an efficient copy network architecture. Essentially the previously proposed fat-banyan (FAB) [35] is used as a copy network. When compared to other banyan-based copy networks [62, 34, 63] the FAB switch can implement multicasting with minimal delay and at reasonable cost. In particular, the FAB network has superior delay performance compared to banyan networks with deflection routing. Liew [63] showed how to enhance cell loss performance by adding more stages to the banyan. For example, for a banyan size $128 \times 128$ to achieve a cell loss of $10^{-4}$ with an output offered load of 0.6 and mean fanout of 2 at the inputs, about 16 stages are required (versus 7 in the standard banyan). In order to achieve cell loss of the order of $10^{-6}$ a speed up of 2 is required in Liew’s copy network. Lee [34] proposed a nonblocking, self-routing copy network with constant latency. In Lee’s method in order to build a copy network with $2^M$ inputs and $2^N$ outputs ($N>M$), $2^{N-M}$ banyan networks each with $M$ stages are required. In addition $2^M (N-M)$-stage binary trees are required. Thus, in order for Lee’s network to handle packet overflows (i.e., total number of copies in excess of output ports) considerable hardware is required. Alternatively Lee’s architecture would require speedup to handle packet overflows.

In section 6.2 we review the multicasting techniques proposed in the literature. In section 6.3 we present the performance results of the FAB copy network under multicast traffic followed by conclusions in section 6.4.

6.2. Multicasting in the FAB Switch

Basically two approaches have been proposed in the literature for the replication of packets in multistage interconnection networks. These are the balanced tree approach and the interval splitting approach. In both the approaches a multicast packet has a copy number field which stores the number of copies (CN) desired for that packet. In the balanced tree approach proposed in [7] the first duplication of the packet happens at
stage $i = 1 + \log_2 N - \lfloor \log_2 CN \rfloor (CN>1)$. The CN of the upper output packet is set to $[CN/2]$ and the copy number of the lower output packet is set to $[CN/2]$, or vice versa. Here contention for output links of a switching element (SE) can be dealt by buffering [7] one of the packets and allowing the other packet to be duplicated or by dropping one of the packets. Dropping of packets may be acceptable only if the cell loss of the multicast packet copies can be kept very low. The balanced tree approach is illustrated in Figure 6.2. Input 3 receives a multicast packet with 6 copies to be generated. In the first stage (stage 0) the packet splits into two with CN of each of the copies set to 3. In the next stage each of the packet copies generated in the previous stage is split again with the CN of the upper packet set to 1 and the CN of the lower packet set to 2. In the last stage packets with CN=2 are split while packets with CN=1 are arbitrarily sent to either the upper or lower output.

![Diagram of balanced tree approach](image)

**Figure 6.2. Balanced tree approach to copy number division per stage.**

The second approach is the output interval assignment approach and is illustrated
in Figure 6.3. In this approach a multicast packet with copy number CN is assigned an output interval (Min, Max) where Min and Max are the minimum and maximum of the output interval. The packet copies are to be sent to the outputs of the copy network specified by this output interval. This scheme can be explained as follows: Let the binary representation of the address interval be \((m_1 m_2 ... m_n, M_1 M_2 ... M_n)\). At any stage \(k\), bits \(m_k\) and \(M_k\) are examined and routing decisions made as follows:

1. If \(m_k = M_k = 0\), then send the packet on the upper link; if \(m_k = M_k = 1\), then send the packet on the lower link.

2. If \(m_k = 0\) and \(M_k = 1\), then duplicate the packet and send it out on both links with the header of the duplicated packets modified as follows:

   a. For the packet sent out on the upper link, Min remains unchanged, and Max is set to \(M_1 ... M_{k-1}101...1\).

   b. For the packet sent out on the upper link, Max remains unchanged, and Min is set to \(M_1 ... M_{k-1}010...0\).

It should be noted that when splitting occurs the new intervals are contiguous to each other, and together they cover the original interval.

In our work we have used the output interval assignment approach of packet replication. The output address intervals are assigned randomly in the range \((i, CN-i-1)\) where \(i \leq N-CN\). Also the first splitting of a packet is delayed as long as possible and occurs at stage \(i = 1 + \log_2 N - [\log_2 CN]\).

6.3. Multicast Performance

In this section we highlight the performance of the FAB network as a copy network. Multicast traffic is modelled using a geometric interarrival distribution with the number
Chapter 6. Multicasting

Figure 6.3. Output interval assignment approach with examination of two bits at each stage.

of packet copies described by a geometric distribution. The following equations then hold

\[ q(Y_i = y) = \text{probability that the number of copies requested by an incoming packet is } y \]

\[ p(X_i = x) = \text{probability that the number of copies generated is } x \]

\[ p(X_i = k) = 1 - \rho \quad k = 0 \]

\[ \rho q(Y_i = k) \quad k = 1, 2, ... \]  

(6.1)

Assuming that \( Y_i \) is distributed according to the truncated geometric distribution with the parameter \( q \), we have

\[ q(Y_i = k) = \frac{(1 - q)q^{k-1}}{1 - q^N} \text{ for } 1 \leq k \leq N \]  

(6.2)

Note that \( N \) is the maximum number of allowable copies since an incoming packet will multicast to at most all of the switch outputs. The average number of copies per
incoming packet is

\[ E(Y_1) = \frac{1}{1 - q} - \frac{Nq^N}{1 - q^N} \quad (6.3) \]

The output offered load is given by

\[ \rho E(Y_1) \quad (6.4) \]

With the following traffic parameters we obtain the simulation results with 95% confidence intervals using the method of independent replications.

In Figure 6.4 cell loss probability is plotted for a 128×128 FAB network as a function of output offered load with mean fanout, \( E(Y_1) = 2 \) for different dilation configurations. For a given load, the cell loss probability improves as the dilation increases. With an output offered load of 0.6 cell loss probability below \( 10^{-4} \) can be achieved with dilation configuration \( DC = [2,3,4,5,5,5,5] \). In [63] where the extended shuffle exchange network is used about 16 stages (an extra 9 stages) are required to achieve a similar cell loss with the same output offered load. The extra stages however do not solve the problem of output request overflow which occurs when the sum of the packet copies exceeds \( N \). The FAB copy network is designed to handle this kind of overflow problem by gradual internal bandwidth expansion. Figure 6.5 shows the case when the output offered load is > 1. It can be seen that with configuration \( DC = [2,4,8,8,8,8,8,8] \) the cell loss for an output offered load of 1 is of the order of \( 10^{-6} \) and when the output offered load is 2 it is of the order of \( 10^{-4} \).

6.4. Conclusion

We have presented an efficient copy network for multicasting based on the fat-banyan architecture. An unique feature of this architecture is that it can support total packet replications > \( N \). With a maximum dilation of 5 the copy network achieves cell loss probability of \( 10^{-4} \) for an output offered load of 0.6 for a 128×128 switch. With a
Figure 6.4. Simulation results for a 128×128 copy network: Cell loss probability versus Output offered load for different dilation configurations with mean fanout = 2.

Figure 6.5. Simulation results for a 128×128 copy network: Cell loss probability versus Output offered load under overload conditions for different dilation configurations with mean fanout = 2.
Chapter 6. Multicasting

maximum dilation of 8 the cell loss probability is below $10^{-6}$ even when the output offered load is 1. In terms of latency the FAB copy network performs very well since FSE’s in any stage examine only a bit each of the output multicast address range bits, [Min, Max], which are present in the header of the multicast cell.
Chapter 7  Conclusions and future research areas

7.1. Conclusions

In this thesis we have provided an in-depth investigation of Banyan-based packet switches. We proposed a new generalized banyan model called the fat-banyan (FAB) [35, 64] switch model characterized by a flexible interstage dilation scheme that can be used to optimize switch cost with respect to internal bandwidth utilization. The optimization of internal bandwidth leads to significantly reduced interchip/interboard interconnection as was shown in Table 3.1. The main features of the FAB switch are 1) amenability to efficient hardware implementation (Table 3.10), 2) modularity and 3) scalability to very large capacities in the range of Terabits/sec as discussed in chapter 3. The performance of the switch shows that it is comparable to the Knockout switch under uniform, non-uniform [36, 37] and bursty traffic patterns. Another important feature of the FAB switch is that the cells from different connections do not affect each other's delays (i.e., no QoS coupling).

The FAB also provides a very cost-effective and flexible realization of Growable switches [11]. By adjusting the depth of the fabric (truncating the fabric) the desired number of outputs (in multiples of $d$ for a FAB switch made with $dxd$ switching elements) can be grouped. For the group of outputs shared memory switches (or any other existing switches) can be used to provide a scalable switching fabric based on the growable switch concept. The truncated fabric concept [38, 39] was introduced in this thesis is based on designing a fabric which is highly optimized both with respect to bandwidth and buffer utilization. It should be noted that the FAB model is a generalized model for designing switches highly optimized for bandwidth and buffer utilization. This concept can be
applied to any multistage interconnection network (eg. Fat-tree, Benes network, Clos Network).

The output buffered FAB switch provides a cost-effective way of realizing a space-division switch which has a comparable performance to the Knockout switch. The TFAB switch provides a cost-effective architecture for realizing the growable switch. For large scale ATM switch implementations it is desirable to reduce the interconnect complexity as much as possible while maintaining minimal cell loss through the switch. Hence, the FAB and TFAB switches were further enhanced by using input-output buffering and backpressure to obtain a lossless fabric [39]. The backpressure technique results in reduction of the dilation requirement within the fabric to obtain a specified cell loss probability. This further reduces the chipcount and interconnection complexity of the switching fabric (refer to Table 3.10). This technique of LC backpressure can be applied to any switching fabric to obtain a lossless fabric.

In order to support applications like telephone conferencing, video conferencing and downloading of video programs, ATM provides a service called point-to-multipoint connections. This requires the ATM switch to create multiple copies of incoming multicast cells. In Chapter 6 we proposed an efficient cell replication method for point-to-multipoint connections by using a copy network based on the FAB architecture [60]. The copy network achieves very low cell loss even at very high output offered loads as shown in Chapter 6. The FAB copy network performs multicasting with minimal delay and reasonable cost in comparison to the extended shuffle exchange network with deflection routing. Moreover, it is capable of producing copies in excess of the number of input/output ports due to its gradual incrementing of internal bandwidth. In contrast, the shuffle exchange network cannot handle the case where the number of copies requested is greater than the number of input/output ports.
Chapter 7. Conclusions and future research areas

7.2. Future research areas

The work in this thesis can be extended to yield some very interesting research topics to further enhance the design of ATM switches. As a continuation of the work in this thesis the following areas can be investigated in future research work.

Efficient support of multicasting within the switch fabric

We have investigated the use of a copy network architecture for replicating cells of point-to-multipoint connections. For large scale switching systems it may be more efficient to incorporate this mechanism within the switching fabric. Efficient mechanisms are needed to control replication of cells within the fabric in order to maintain good performance of the switch. The recursive copy generation technique [32, 65] has been used in other switch fabrics to combine copying and routing of multicast cells. This technique is based on generating a limited number of copies inside the fabric, and then recycling some output copies of a cell at the input of the switching fabric such that the input lines are equally loaded. Further study should enable the feasibility of including this feature in the FAB switch. For the TFAB switch an approach based on combining the recursive copy generation with one of the techniques proposed in [61] for multicasting in shared-memory switches may prove to be efficient.

Efficient support of multicasting of ABR/UBR traffic in an ATM switch

In ATM environment CBR and VBR traffic are subject to connection admission control (CAC). The multicasting of this kind of traffic which is subject to CAC is relatively simpler compared to traffic like ABR and UBR. ABR is based on rate based flow control with a Minimum Cell Rate (MCR) guarantee while with UBR there are no QoS guarantees. Research work is required to determine how to efficiently multicast ABR and UBR; for instance in a point-to-multipoint ABR connection what is the best
Chapter 7. Conclusions and future research areas

policy to deal with congestion on one or more paths of the multicast tree, what is its effect on the cell replication algorithm, how should the multicast cell be buffered.

Enhancing the performance of the FAB by using a Benes network topology

The Buffered Benes network has been used as the basis for the Washington University Gigabit switch [1]. This network suffers from QoS coupling within the fabric due to the internal buffers. However, the technique of using the first half of the Benes network as a randomizing network and subsequently doing the routing in the second half of the network can be effectively used to enhance the performance of the FAB switch. It would be very interesting to study the performance of this network with the second half replaced by a FAB network. The randomization in the first half will probably lead to a decrease in dilation requirement (hence cost) of the second half (FAB network) to achieve a specified cell loss probability.

Enhancing fabric performance to deal with multi-QoS traffic

The switch fabrics which have been considered in the literature are generally insensitive to the QoS requirement of the traffic traversing the fabric. In order to maintain the QoS of different streams through the switch there is a need to enhance the switch fabric design to be sensitive to the QoS of the traffic. For a memoryless fabric like the FAB switch this would require enhancing the arbitration mechanism in the switching elements. This feature will allow higher priority cells to be given preference in case of conflict.
## Appendix: A: List of standard acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAL</td>
<td>ATM adaption layer</td>
</tr>
<tr>
<td>ABR</td>
<td>Available bit rate</td>
</tr>
<tr>
<td>ATM</td>
<td>Asynchronous transfer mode</td>
</tr>
<tr>
<td>BISDN</td>
<td>Broadband integrated services digital network</td>
</tr>
<tr>
<td>CAC</td>
<td>Connection admission control</td>
</tr>
<tr>
<td>CBR</td>
<td>Constant bit rate</td>
</tr>
<tr>
<td>FIFO</td>
<td>First in first out</td>
</tr>
<tr>
<td>GFC</td>
<td>Generic flow control</td>
</tr>
<tr>
<td>HEC</td>
<td>Header error control</td>
</tr>
<tr>
<td>IP</td>
<td>Internet protocol</td>
</tr>
<tr>
<td>LAN</td>
<td>Local area network</td>
</tr>
<tr>
<td>MCR</td>
<td>Minimum cell rate</td>
</tr>
<tr>
<td>NNI</td>
<td>Network network interface</td>
</tr>
<tr>
<td>OAM</td>
<td>Operation and maintainance</td>
</tr>
<tr>
<td>OC-n</td>
<td>Optical carrier signal</td>
</tr>
<tr>
<td>PT</td>
<td>Payload type</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of service</td>
</tr>
<tr>
<td>RM</td>
<td>Resource management</td>
</tr>
<tr>
<td>TCP</td>
<td>Transport control protocol</td>
</tr>
<tr>
<td>UBR</td>
<td>Unspecified bit rate</td>
</tr>
<tr>
<td>UNI</td>
<td>User Network Interface</td>
</tr>
<tr>
<td>VBR</td>
<td>Variable bit rate</td>
</tr>
<tr>
<td>VCI</td>
<td>Virtual circuit identifier</td>
</tr>
<tr>
<td>VPI</td>
<td>Virtual path identifier</td>
</tr>
<tr>
<td>WAN</td>
<td>Wide Area Networking</td>
</tr>
</tbody>
</table>
Bibliography


1994.


[18] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, “Input versus output queueing on space-


Bibliography

1990.


Bibliography


Bibliography


