MASTER

Design of an ATM-based multimedia network

van Lierop, J.J.

Award date:
1995

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
Master's Thesis:

Design of an ATM-Based Multimedia Network

Jeroen J. van Lierop

Coaches: ir. G.M.J. Havermans (Philips)
         ing. F.P. Derks (Philips)
Supervisor: prof. ir. M.P.J. Stevens
Period: September 1994 - July 1995
Acknowledgements

First of all, I would like to thank my supervisor, Prof. Stevens, for giving me the opportunity to graduate under his supervision.

I gratefully acknowledge the support of the members of the Product Innovation department of Philips Business Communication Systems in Hilversum. Special thanks go to Gerard Havermans and Frank Derks who have been intensively engaged in this project and without whom this thesis would have been impossible. Gerard, thank you for being my tutor during the last months and for your guidance, patience and encouragement. Frank, I’m indebted to your everlasting support, comments and contributions to the project. I would also like to thank Oscar Rikkert de Koe for reading and commenting on this report.

Finally, my thanks go to my roommate Allard van der Heijden. Allard, thanks for your good company during our stay in KOC 3.06 and for not throwing me out during the last days of writing.
Abstract

The appearance of new (multimedia) networking applications, such as video conferencing, bring about requirements for networks that cannot be met by the current telecommunication and datacommunication infrastructures. Therefore, a new multimedia networking solution is required that supports these multimedia applications and can also replace the above mentioned infrastructures by supporting their services.

In the long run, a single Broadband Integrated Services Digital Network (B-ISDN) will exist that supports all services mentioned above. This network will make use of an efficient switching technique, called Asynchronous Transfer Mode (ATM), that combines efficient bandwidth utilization with quality of service guarantee and the support of multiple bit-rates. Until the B-ISDN is fully operational, multimedia networks will only exist in local and metropolitan areas. Since these networks have similar requirements and need to be interconnected with the B-ISDN in the long run, ATM is a very suitable technique for these networks as well.

A customer premises multimedia network typically comprises a set of ATM end-systems, a set of ATM switches and means to interconnect them (interfaces). It is believed that the following features increase the value of an ATM-based multimedia networking solution:

- Ethernet LANs are connected to the switches without interworking units. The switches are capable of acting like a LAN bridge thereby simultaneously supporting ATM and Ethernet.
- The switches internally contain a bus structure that allows easy broadcasting. This facilitates the interconnection of LANs and the implementation of LAN emulation, a service that supports LAN interconnection and the use of LAN applications by emulating virtual LANs (VLANs).
- The network contains an embedded clock distribution facility that allows it to support both isochronous and asynchronous services without expensive resynchronisation and delay compensation.
- The switches in the network are typically group-switches that do not need huge amounts of bandwidth. Therefore, the architecture of the switches may be relatively simple, which may significantly lower the cost.
- By reducing the bit-rate to the end-user, the cost of the interfaces may also be reduced.

The architecture that has been presented in this report, provides a scalable network built with switches that internally use a high speed TDM-bus, named "Autobahn". This bus provides a means to switch reasonably high bit-rates (currently up to a total of 1.6 Gbits/s per switch) with a relatively low complexity. Furthermore, it facilitates the implementation of multicast and broadcast, thereby making it an excellent solution for LAN interconnection. The switches internally use synchronous time-slot assignment, which facilitates the implementation of isochronous services.

VLANs are implemented by means of multipoint-to-multipoint Permanent Virtual Connections (PVCs), thereby creating virtual shared media. This allows the ATM network to function as a LAN bridge without having to set up a large number of ATM connections. Furthermore, the multicast server is distributed over the switches, so that separate multicast servers are
avoided. This implementation requires additional functionality in the switches. Since it is believed that the cost of a separate multicast server would exceed the additional cost of the proposed implementation, the additional cost of this implementation is not troublesome.

An example of a low-cost ATM interface is the ATM-on-top-of-Ethernet interface, which makes use of existing Ethernet hardware. If an end-system can implement the ATM functions in software, it can use existing Ethernet adapters. The ATM-on-top-of-Ethernet principle, as described in chapter 8, turns out to be useful for low-end multimedia applications.

Two approaches can be distinguished when upgrading a Microsoft Windows-based PCs to a multimedia end-systems, namely a software-based and a hardware-based. The software-based approach uses an Ethernet adapter and additional software to implement the ATM-on-top-of-Ethernet and multimedia functions. It is useful for "no-budget" support of near real-time services. The hardware-based approach supports true real-time services by means of a multimedia networking board, that includes audio and video functionality and an ATM(-on-top-of-Ethernet) network interface. Since this does not allow the use of the installed base of Ethernet adapters, much of the attractiveness of ATM-on-top-of-Ethernet is lost. A possible implementation of a multimedia board can be realised by using the Brooktree BtV MediaStream chipset.

During the graduation project, a lot of issues regarding the development of the customer premises multimedia network have been studied. The general aspects of such a network have been discussed and some of these aspects have been worked out in detail. However, there still remains a lot of work to be done.
# Table of Contents

1 Introduction ................................................................. 1  
  1.1 Context .............................................................. 1  
  1.2 Characteristics of a customer premises multimedia network ....... 2  
  1.3 Outline ............................................................. 2  

2 LAN techniques ............................................................. 4  
  2.1 IEEE standard 802.3 and Ethernet .................................. 4  
      2.1.1 Frame format ............................................... 4  
  2.2 IEEE standard 802.4: token bus .................................... 5  
  2.3 IEEE 802.5: token ring ............................................. 6  
  2.4 FDDI/CDDI .......................................................... 7  
  2.5 Ethernet derivatives ................................................ 7  
      2.5.1 Switched Ethernet ........................................... 7  
      2.5.2 Fast Ethernet ............................................... 7  
      2.5.3 100VG/AnyLAN .............................................. 8  
      2.5.4 Isochronous Ethernet ....................................... 8  
  2.6 Comparison of LANs ................................................ 8  

3 Asynchronous Transfer Mode (ATM) ...................................... 10  
  3.1 The B-ISDN model .................................................. 10  
  3.2 Physical layer ..................................................... 11  
  3.3 ATM layer .......................................................... 12  
  3.4 ATM adaptation layer .............................................. 14  
      3.4.1 AAL type 1 ................................................... 15  
      3.4.2 AAL type 2 ................................................... 16  
      3.4.3 AAL type 3/4 ................................................. 16  
      3.4.4 AAL type 5 ................................................... 17  
  3.5 ATM connections ..................................................... 17  
  3.6 ATM LAN emulation ................................................ 19  
  3.7 Summary ........................................................... 22  

4 ATM switching techniques ................................................ 24  
  4.1 Principle of an ATM switch ....................................... 24  
  4.2 Implementation aspects of switches ................................ 25  
      4.2.1 Transfer media ............................................. 25  
      4.2.2 Queuing methods ............................................ 26  
      4.2.3 Routing methods ............................................ 26  
  4.3 Summary and conclusions .......................................... 28  

5 Multimedia networking requirements .................................... 29  
  5.1 Multimedia networking ............................................. 29  
  5.2 General requirements for a multimedia networking solution ...... 30  
  5.3 Why ATM? .......................................................... 31  
  5.4 ATM network requirements ......................................... 32  
      5.4.1 Flexibility and migration .................................. 32
# TABLE OF CONTENTS

5.4.2 Scalability ................................................. 33  
5.4.3 Simplified network ...................................... 33  
5.4.4 Lower cost .............................................. 33  
5.4.5 More bandwidth ......................................... 34  
5.4.6 Multimedia ............................................. 34  
5.4.7 Other requirements ..................................... 34  
5.5 Summary and conclusions .................................. 35  

6 ATM networking functional description ...................... 37  
6.1 Network modelling ......................................... 37  
6.1.1 Definitions ............................................ 37  
6.1.2 Network model ......................................... 38  
6.1.3 Addressing and routing ................................ 39  
6.2 End-system interfaces ..................................... 40  
6.3 Network users ............................................ 42  
6.4 Summary and conclusions .................................. 43  

7 ATM networking design aspects ............................... 44  
7.1 Network structure .......................................... 44  
7.2 Number of connections ..................................... 45  
7.3 Queuing ................................................... 46  
7.4 LAN emulation implementation ............................. 47  
7.5 Network timing support for isochronous services ............ 48  
7.6 Summary and conclusions .................................. 48  

8 ATM-on-top-of-Ethernet ....................................... 50  
8.1 Guaranteeing quality of service ............................. 50  
8.2 Implementation aspects ..................................... 51  
8.2.1 The ATM-on-top-of-Ethernet model ..................... 51  
8.2.2 Clock synchronisation .................................. 51  
8.2.3 Bandwidth usage ....................................... 52  
8.2.4 Packing methods ....................................... 52  
8.3 Conclusions ................................................ 53  

9 Switch design aspects ......................................... 55  
9.1 Functions of a switch ...................................... 55  
9.2 Structure of the switch .................................... 56  
9.3 Multipoint-to-multipoint connections ...................... 59  
9.4 Routing .................................................. 60  
9.5 Backplane design .......................................... 61  
9.5.1 Frame format .......................................... 62  
9.5.2 Synchronisation ....................................... 62  
9.5.3 Dimensioning .......................................... 63  
9.6 Board design .............................................. 63  
9.6.1 Line Interface Module (LIM) ........................... 66  
9.6.1.1 Description of the MACE ......................... 67  
9.6.2 Autobahn Interface Module (AIM) ..................... 68
9.6.3 Switch Control Module (SCM) ........................................ 70
9.6.4 Board Control Module (BCM) ...................................... 71
9.6.5 LIM-SCM interface ................................................... 71
9.6.6 SCM-AIM interface .................................................. 72
9.6.7 SCM-BCM interface .................................................. 72
9.6.8 Queue dimensioning .................................................. 72
9.7 Summary and conclusions ................................................ 72

10 Multimedia end-system design aspects ................................. 74
10.1 Introduction ............................................................. 74
10.2 Multimedia capabilities of Microsoft Windows ......................... 74
10.3 Software-based approach ............................................. 75
10.4 Hardware-based Multimedia PC ....................................... 76
   10.4.1 The BtV MediaStream family .................................. 77
   10.4.2 Recommendations on board implementation ..................... 78
10.5 Conclusions .................................................................. 78

11 Conclusions ..................................................................... 80

12 Recommendations ................................................................ 82

References ............................................................................ 83

Appendix A Functional description of the network ....................... 87
   A.1 Network model .......................................................... 87
      A.1.1 Application - network service interactions .................. 87
      A.1.2 Subnetwork services ............................................. 88
      A.1.3 Subnetwork elements interactions ............................ 89
   A.2 Interworkings ........................................................... 89

Appendix B ATM-on-top-of-Ethernet calculations ....................... 90
   B.1 Proposed methods ...................................................... 90
   B.2 Maximum bit-rate calculations ..................................... 90

Appendix C Windows architectures ........................................... 97
   C.1 Windows architectures .............................................. 97
      C.1.1 Windows 3.x architecture .................................... 97
      C.1.2 Windows NT architecture ..................................... 99
      C.1.3 Windows 95 architecture ..................................... 102
   C.2 Windows networking .................................................. 102
      C.2.1 Windows 3.x networking ...................................... 103
      C.2.2 Windows NT networking ...................................... 103
      C.2.3 Windows 95 networking ...................................... 104
   C.3 Windows multimedia architectures .................................. 105
      C.3.1 Windows audio architecture ................................... 105
      C.3.2 Video for Windows architecture .............................. 105
   C.4 Windows timing ........................................................ 107
      C.4.1 Windows timers ................................................... 107
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.4.2</td>
<td>Windows Multimedia timers</td>
<td>108</td>
</tr>
<tr>
<td>C.4.3</td>
<td>Virtual timer services</td>
<td>108</td>
</tr>
<tr>
<td>C.5</td>
<td>Windows and multimedia networking</td>
<td>109</td>
</tr>
<tr>
<td>C.5.1</td>
<td>Windows multimedia platforms</td>
<td>110</td>
</tr>
<tr>
<td>C.6</td>
<td>Literature</td>
<td>110</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1</td>
<td>Ethernet/802.3 frame format</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>LLC-PDU</td>
<td>6</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>ATM network</td>
<td>10</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>B-ISDN Protocol Reference Model</td>
<td>11</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>ATM cell structure</td>
<td>13</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>SAR-PDU format for AAL type 1</td>
<td>15</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>SAR-PDU of AAL type 3/4</td>
<td>16</td>
</tr>
<tr>
<td>Figure 3.6</td>
<td>Coding of AAL Type 5</td>
<td>17</td>
</tr>
<tr>
<td>Figure 3.7</td>
<td>ATM connection relationships</td>
<td>18</td>
</tr>
<tr>
<td>Figure 3.8</td>
<td>Virtual paths and virtual channels</td>
<td>19</td>
</tr>
<tr>
<td>Figure 3.9</td>
<td>LE connections</td>
<td>21</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>ATM switching principle</td>
<td>24</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Transfer media</td>
<td>25</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Switching element with input, output and central queues</td>
<td>27</td>
</tr>
<tr>
<td>Figure 6.1</td>
<td>End-systems connected to a subnetwork</td>
<td>38</td>
</tr>
<tr>
<td>Figure 6.2</td>
<td>Practical example of subnetworks</td>
<td>39</td>
</tr>
<tr>
<td>Figure 6.3</td>
<td>Protocol model of example</td>
<td>40</td>
</tr>
<tr>
<td>Figure 6.4</td>
<td>ATM protocol stack</td>
<td>41</td>
</tr>
<tr>
<td>Figure 6.5</td>
<td>LAN Emulation protocol architecture</td>
<td>42</td>
</tr>
<tr>
<td>Figure 7.1</td>
<td>Cross Connects</td>
<td>45</td>
</tr>
<tr>
<td>Figure 8.1</td>
<td>ATM-on-top-of-Ethernet concept</td>
<td>51</td>
</tr>
<tr>
<td>Figure 9.1</td>
<td>Switch block diagram</td>
<td>56</td>
</tr>
<tr>
<td>Figure 9.2</td>
<td>Board layout</td>
<td>57</td>
</tr>
<tr>
<td>Figure 9.3</td>
<td>Network model applied to switch</td>
<td>58</td>
</tr>
<tr>
<td>Figure 9.4</td>
<td>Network model with LAN emulation</td>
<td>59</td>
</tr>
<tr>
<td>Figure 9.5</td>
<td>Routing tag structure</td>
<td>61</td>
</tr>
<tr>
<td>Figure 9.6</td>
<td>Autobahn frame</td>
<td>62</td>
</tr>
<tr>
<td>Figure 9.7</td>
<td>Autobahn slot contents</td>
<td>62</td>
</tr>
<tr>
<td>Figure 9.8</td>
<td>Functional block diagram of a board</td>
<td>64</td>
</tr>
<tr>
<td>Figure 9.9</td>
<td>Board block diagram</td>
<td>66</td>
</tr>
<tr>
<td>Figure 9.10</td>
<td>Line Interface Module block diagram</td>
<td>66</td>
</tr>
<tr>
<td>Figure 9.11</td>
<td>MACE block diagram</td>
<td>67</td>
</tr>
<tr>
<td>Figure 9.12</td>
<td>Spaceceiver block diagram</td>
<td>69</td>
</tr>
<tr>
<td>Figure 9.13</td>
<td>Switch Control Module block diagram</td>
<td>71</td>
</tr>
<tr>
<td>Figure 10.1</td>
<td>Software based solution</td>
<td>76</td>
</tr>
<tr>
<td>Figure 10.2</td>
<td>BtV chipset for multimedia</td>
<td>77</td>
</tr>
<tr>
<td>Figure 10.3</td>
<td>Multimedia board</td>
<td>78</td>
</tr>
<tr>
<td>Figure A.1</td>
<td>Network model</td>
<td>87</td>
</tr>
<tr>
<td>Figure B.1</td>
<td>Maximum bit-rates at the cell-payload level</td>
<td>92</td>
</tr>
<tr>
<td>Figure B.2</td>
<td>ATM on top of Ethernet timing diagram</td>
<td>93</td>
</tr>
<tr>
<td>Figure B.3</td>
<td>One-way bit-rates</td>
<td>94</td>
</tr>
<tr>
<td>Figure C.1</td>
<td>Windows 3.1 architecture</td>
<td>97</td>
</tr>
<tr>
<td>Figure C.2</td>
<td>Windows NT architecture</td>
<td>99</td>
</tr>
<tr>
<td>Figure C.3</td>
<td>Windows NT video display driver model</td>
<td>102</td>
</tr>
<tr>
<td>Figure C.4</td>
<td>Windows NT networking architecture</td>
<td>104</td>
</tr>
<tr>
<td>Figure C.5</td>
<td>Windows audio architecture</td>
<td>105</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------------------------------------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>Figure C.6</td>
<td>Overview of the Video for Windows modules</td>
<td>106</td>
</tr>
</tbody>
</table>
List of Tables

Table 3.1  Functions of the B-ISDN Layers .................................................. 12
Table 3.2  Service Classes for AAL ................................................................. 14
Table 5.1  Characteristics of audio and video services ................................. 30
Table 5.2  Characteristics of data services ..................................................... 31
Table 8.1  Characteristics of packing methods ............................................. 53
Table 8.2  Comparison of packing methods ................................................... 54
Table B.1  Characteristics of packing methods .............................................. 90
Table B.2  Possible frame sizes ................................................................. 95
Table B.3  Comparison modified method 4 to method 1 and 2 ...................... 96
List of Acronyms

AAL  ATM Adaptation Layer
AB   AutoBahn
ABR  Available Bit-Rate
ADPCM Adaptive Delta Pulse Code Modulation
AFB  Autobahn Filter Block
API  Application Programming Interface
ASCII American Standard Code for Information Interchange
ATB  Address Translation Block
ATM  Asynchronous Transfer Mode
AUU  ATM User to User
B-ICI Broadband Inter-Carrier Interface
B-ISDN Broadband Integrated Services Digital Network
BCM  Board Control Module
BIB  Board Interface Block
BOM  Beginning Of Message
BUS  Broadcast and Unknown Server
CBR  Constant Bit-Rate
CCITT Commite Consultatif International Télegraphique et Téléphonique
CDDI Copper Distributed Data Interface
CLP  Cell Loss Priority
CMC  Common Mezzanine Card
COM  Continuation Of Message
CPCS Common Part Convergence Sublayer
CPI  Common Part Indicator
CRC  Cyclic Redundancy Check
CS  Convergence Sublayer
CSMA/CD Carrier Sense Multiple Access with Collision Detect
DA  Destination Address
DIX  Digital Intel Xerox
DLPI Data-Link Provider Interface
DMA  Direct Memory Access
DSAP Destination Service Access Point
ECMA European Computer Manufacturers Association
EOM  End Of Message
FCS  Frame Check Sequence
FDDI Fiber Distributed Data Interface
FEC  Forward Error Control
FIFO First In First Out
FM  Frequency Modulation
GFC  Generic Flow Control
HDTV High Definition TeleVision
HEC  Header Error Control
IEEE Institute of Electrical an Electronics Engineers
IP  Internet Protocol
ISDN  Integrated Services Digital Network
ISO  International Standardization Organization
ITU  International Telecommunication Union
IWU  InterWorking Unit
LAN  Local area Network
LANC  Controller for Ethernet
LAPD  Link Access Protocol D-channel
LAPPEN  Look-Ahead Packet Processing Enable
LD  Low Delay
LD-CELP  Low Delay-Code Excited Linear Predictive Coding
LE  LAN Emulation
LE-ARP  LAN Emulation Address Resolution Protocol
LEC  LAN Emulation Client
LECS  LAN Emulation Configuration Server
LES  LAN Emulation Server
LI  Length Indicator
LIM  Line Interface Module
LL  Low Loss
LLC  Logical Link Control
LSB  Least Significant Bit
MAC  Media Access Control
MCI  Media Control Interface
MID  Multiplexing Identifier
MPEG  Motion Picture Experts Group
NDIS  Network Driver Interface Specification
NIC  Network Interface Card
NNI  Network-Node Interface
NTSC  National Television System Committee
ODI  Open Data-link Interface
OSI  Open Systems Interconnect
PACDAC  Packetized Digital to Analog Convertor
PAL  Phase Alternating Line
PC  Personal Computer
PCI  Peripheral Component Interface
PCM  Pulse Code Modulation
PDU  Protocol Data Unit
PLL  Phase Locked Loop
PM  Physical Medium
PMC  PCI Mezzanine Card
PT  Payload Type
QCB  Queue Control Block
QID  Queue Identifier
QoS  Quality of Service
RES  REServed
RGB  Red Green Blue
RTAG  Routing TAG
SA  Source Address
SAP  Service Access Point
SAR  Segmentation And Reassembly
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS</td>
<td>System Assurance Software</td>
</tr>
<tr>
<td>SCM</td>
<td>Switch Control Module</td>
</tr>
<tr>
<td>SDU</td>
<td>Service Data Unit</td>
</tr>
<tr>
<td>SEAL</td>
<td>Simple and Efficient Adaptation Layer</td>
</tr>
<tr>
<td>SFD</td>
<td>Start of Frame Delimiter</td>
</tr>
<tr>
<td>SN</td>
<td>Sequence Number</td>
</tr>
<tr>
<td>SNMP</td>
<td>Simple Network Management Protocol</td>
</tr>
<tr>
<td>SNP</td>
<td>Sequence Number Protection</td>
</tr>
<tr>
<td>SOPHO</td>
<td>Synergetic Open PHilips Office automation</td>
</tr>
<tr>
<td>SRTS</td>
<td>Synchronous Residual Time Stamp</td>
</tr>
<tr>
<td>SSAP</td>
<td>Source Service Access Point</td>
</tr>
<tr>
<td>SSCS</td>
<td>Service Specific Convergence Sublayer</td>
</tr>
<tr>
<td>SSM</td>
<td>Single Segment Message</td>
</tr>
<tr>
<td>ST</td>
<td>Segment Type</td>
</tr>
<tr>
<td>SVGA</td>
<td>Super Video Graphics Array</td>
</tr>
<tr>
<td>TC</td>
<td>Transmission Convergence</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TDM</td>
<td>Time Division Multiplexing</td>
</tr>
<tr>
<td>UNI</td>
<td>User-Network Interface</td>
</tr>
<tr>
<td>UNSPM</td>
<td>Unconstrained Network Service Provider Model</td>
</tr>
<tr>
<td>UTP</td>
<td>Unshielded Twisted Pair</td>
</tr>
<tr>
<td>UU</td>
<td>User-to-User</td>
</tr>
<tr>
<td>VBR</td>
<td>Variable Bit-Rate</td>
</tr>
<tr>
<td>VC</td>
<td>Virtual Channel</td>
</tr>
<tr>
<td>VCC</td>
<td>Virtual Channel Connection</td>
</tr>
<tr>
<td>VCI</td>
<td>Virtual Channel Identifier</td>
</tr>
<tr>
<td>VLAN</td>
<td>Virtual LAN</td>
</tr>
<tr>
<td>VP</td>
<td>Virtual Path</td>
</tr>
<tr>
<td>VPC</td>
<td>Virtual Path Connection</td>
</tr>
<tr>
<td>VPI</td>
<td>Virtual Path Identifier</td>
</tr>
</tbody>
</table>
List of symbols

$H_f$: Size of Ethernet frame header + trailer
$K$: Subnetwork layer
$N$: Number of cells in an Ethernet frame
$r_b$: bit-rate
$S_c$: Cell size
$S_f$: Size of an Ethernet frame
$S_{f_{\text{max}}}$: Maximum size of an Ethernet frame
$S_s$: Slot size
$T$: Length of time-frame
$T_{\text{H}}$: Delay caused by end-system
$T_f$: Time to transmit an Ethernet frame
$T_i$: Total idle time
$T_p$: Propagation delay
$T_s$: Delay caused by switch
1 Introduction

1.1 Context

Currently, two communication infrastructures exist. The first infrastructure is referred to as the telecommunication infrastructure. It evolved from telephony networks and now includes a worldwide network indicated as ISDN (Integrated Services Digital Network). Typical applications for this infrastructure are telephony (voice), facsimile and modem. The second infrastructure is referred to as the datacommunication infrastructure and comprises the interconnection of computers for the purpose of exchanging data. This infrastructure, which in fact consists of many local computer networks that are interconnected to each other, is based on many different standards. Typical applications for this infrastructure are file transfer, electronic mail and remote computing. Although many different techniques are used, eighty percent of the datacommunication networks is based on Ethernet [DAT93]. Other techniques are Token-Ring, FDDI/CDDI and variations on these technologies.

The characteristics of datacommunication networks are entirely different from those of telecommunication networks. Datacommunication networks are typically asynchronous networks: there is no explicit timing relationship between the sending and receiving of data. The delay data encounters when traversing the network is of minor importance. What is important, is the fact that the data traverses the network without being mutilated. Telecommunication networks, on the other hand, must provide an explicit timing relationship between the sending and receiving of data. Especially for voice services, it is very important that the delay of the data is short and constant and the jitter (variation in delay) is small. Avoiding data mutilation is of minor importance.

Future communication solutions will integrate datacommunication and telecommunication networks into a single infrastructure that will offer every service that is provided by current networks, and more. New multimedia networking applications that are not supported by datacommunication and telecommunication networks are real-time video conferencing and distribution applications, and document sharing. These applications require synchronisation of source and destination and large amounts of bandwidth that current networking solutions cannot provide. ATM is a new switching technique that is suited for multimedia networking. Obviously, a multimedia network must also support the above mentioned datacommunication and telecommunication applications.

In the long run, a single Broadband ISDN (B-ISDN) wide area network will exist that supports all multimedia services mentioned above. In the meantime, multimedia networks will only exist in local and metropolitan areas. The requirements of such a multimedia network depend on the application the network is used for. Public networks, for example, need to provide extremely reliable services, while customer premises networks may be simpler. This thesis describes the development of a customer premises multimedia network.
1.2 Characteristics of a customer premises multimedia network

When developing a multimedia network, the first thing to do is recognise the demands of users of current networks towards multimedia networks. Subsequently, a switching technique needs to be selected that is capable of fulfilling the needs of multimedia applications. It appears that the Asynchronous Transfer Mode (ATM) is the promising technology that can fulfil these needs. It has to be decided if ATM is the most suitable technology. Information is needed on basic ATM functionality, how ATM guarantees support of all the services described in the previous section, how ATM supports existing networking applications and how ATM provides interworking with existing networks.

A customer premises ATM network typically comprises a set of ATM end-systems, a set of ATM switches, and means to interconnect them. Each of these have to be specified as precise as possible with the objective to clarify the associated pro’s and con’s and to distinguish them from other solutions. Special attention needs to be paid to cost reduction, manageability, how to provide migration from current networks to the multimedia network and how to provide interworking with other networks (e.g. LANs, ISDN).

Finally, a closer look must be taken at multimedia end-systems. A multimedia network is of no use without multimedia end-systems. The most important question is how to upgrade today’s systems (in particular Microsoft Windows-based PCs) to full-blown multimedia end-systems. Again, the cost of such an upgrade is a major issue.

1.3 Outline

This thesis is organised in ten chapters and two appendices. It begins with a description of present Local Area Network (LAN) techniques and their advantages and disadvantages in chapter 2. This chapter discusses Ethernet, Token bus, Token ring, FDDI/CDDI, Fast Ethernet, Switched Ethernet, VG/AnyLAN and Isochronous Ethernet are discussed.

Chapter 3 introduces the Asynchronous Transfer Mode (ATM) technique. It starts with a presentation of the layered B-ISDN model and a description of the functions of each layer. Subsequently, the ATM paradigm of virtual channels and virtual paths is explained. The chapter concludes with a description of ATM LAN emulation, a service defined by the ATM Forum, which enables LAN applications to run on top of ATM.

Chapter 4 describes general aspects on ATM switching. It describes the main tasks of a switch and several ways to implement these tasks. The issues discussed are: transfer media, queuing and routing.

The above described chapters provide background information that is needed by the rest of this document. Chapter 5 is the first chapter that discusses the design of the multimedia network. It describes the requirements of a multimedia network and why ATM is a suitable switching technique for such a network.
Chapter 6 presents a functional description of a customer premises ATM network and identifies the users of this network. The network is modelled and several general issues are discussed.

Chapter 7 describes the decisions that have been made during the design of the network. These decisions exert influence on the performance of the network and the applicability for potential customers.

Chapter 8 describes a low-cost ATM interface, named "ATM-on-top-of-Ethernet". This interface supports true ATM services over an Ethernet physical medium. The chapter describes the functioning of the ATM-on-top-of-Ethernet protocol and how it guarantees quality of service.

The ATM network consists of ATM switches. These ATM switches are designed in chapter 9. This chapter first describes the functions of a switch. Subsequently, the structure of the switch is described and how it performs its functions.

Another important part of the multimedia network is the multimedia end-system. Chapter 10 describes how a Microsoft Windows-based personal computer can be upgraded to a multimedia end-system. Two approaches are described, namely a software-based approach and a hardware-based approach.

Chapter 11 and chapter 12 describe the conclusions that have been drawn and the recommendations for further study.

Appendix A presents a starting-point for the functional description of the network to be designed. As a matter of fact, this description pertains to chapter 6, but so far too it contains too little information to incorporate it in this chapter.

Appendix B presents calculations that are used in chapter 8.

Appendix C gives an overview of the Microsoft Windows operating systems. This information is needed for the design of a multimedia end-system in chapter 10.
2 LAN techniques

2.1 IEEE standard 802.3 and Ethernet

The IEEE 802.3 standard [ISO90a] is for a Carrier Sense Multiple Access with Collision Detection (CSMA/CD) LAN. The topology is that of a shared (broadcast) medium to which all end-systems are connected. CSMA/CD is a technique to detect and recover from conflicting use of the broadcast medium. If an end-system has data to transmit, it listens to the medium to see if anyone else is transmitting. If the medium is busy (because another end-system is currently transmitting), it waits until the medium is idle. If the medium is idle, the end-system starts transmitting. If two end-systems start transmitting at the same time, a collision occurs. In this case, all colliding end-systems abort transmission and wait a random amount of time. After this, the described procedure is started again.

The IEEE 802.3 standard evolved from an experimental 2.94 Mbits/s CSMA/CD network, developed by Xerox in the 1970s. This system, called Ethernet, was so successful that Xerox, DEC and Intel developed a standard for a 10 Mbits/s Ethernet (DIX-Ethernet, named after the three developers). This standard formed the basis for 802.3 which differs from the Ethernet standard in that it describes a whole family of 1-persistent CSMA/CD systems, running at speeds from 1 to 10 Mbits/s on various media. Nowadays, the term Ethernet has become widely accepted to refer to both DIX-Ethernet and 802.3, although strictly speaking this is incorrect.

The Ethernet protocol provides the physical layer functions and the functions of the MAC sublayer of the datalink layer. Chapter 8 will describe a means to put ATM-on-top-of-Ethernet for which information is needed about the frame structure of Ethernet. Therefore, the Ethernet frame structure is described in this section. The unit of transmission is a variable-length frame. The minimum length of a frame is 64 octets, the maximum length is 1518 octets and the gap between two frames must be at least 9.6 μs.

2.1.1 Frame format

The frame formats of Ethernet and 802.3 are depicted in Figure 2.1. An 802.3 frame starts with a Preamble field which contains seven times the bit-pattern 10101010 and is used by the receiver for synchronisation purposes. The Start of Frame Delimiter (SFD) field contains the bit-pattern 10101011 and indicates the start of a frame. The Destination Address and Source Address fields contain the MAC addresses of the system(s) for which the frame is intended and the system sending the frame. A MAC address is a 48-bit address of which the first bit (LSB) indicates whether the address is a group address (1) or an individual address (0), and the second bit is used to indicate whether the address has global (1) or local (0) significance. The Length field contains the number of user data octets in the data field. The Data field contains the actual data. For this field, a minimal length of 46 octets is required to facilitate the discrimination between aborted frame fragments and valid frames. If the length of the data is shorter than 46 octets, the data is extended by appending extra octets (padding). The actual length of the data is determined from the Length field. The Frame Check Sequence (FCS) field contains a 32-bits Cyclic Redundancy Check (CRC) which is computed as a function of the contents of all fields except the preamble, SFD, and FCS.
CHAPTER 2: LAN TECHNIQUES

Ethernet frames are slightly different from 802.3 frames. Ethernet does not have a length field, but a Type field which indicates which higher layer protocol is used. Because the maximum length of the data field is 1500 octets and the coding of none of the assigned protocol types corresponds with a value smaller than 1500, these two frame types can safely coexist on the same LAN.

To allow multiple protocols to use the same network, the Logical Link Control (LLC) layer, which resides on top of the Ethernet MAC layer, provides a way to identify which protocol is used on top of it. The Logical Link Control Protocol Data Unit (LLC-PDU), which is transported in the data field of the 802.3 frame, is shown in Figure 2.2. The LLC header contains fields to identify the Service Access Points (SAPs) for the services of the next higher layer. The Destination SAP (DSAP) identifies the SAP at the receiving system; the Source SAP (SSAP) identifies the SAP at the transmitting system. However, only a limited number of services is identified a SAP value. To provide organizations a method to define their own proprietary SAPs, the value OxAA in the DSAP and SSAP fields indicates the presence of SubNetwork Access Protocol (SNAP) fields. The SNAP Organization field identifies the organization that offers the service. The SNAP Protocol field identifies the service that is offered. The Control field is used by the LLC protocol to identify the type of the frame (information transfer, supervisory, etc.).

2.2 IEEE standard 802.4: token bus

Although 802.3 is widely used, it has a couple of serious limitations. For one thing, end-systems may have to wait arbitrarily long to send a frame. For another, 802.3 frames do not have priorities which makes them unsuitable for real time information transfer.

A simple medium with a known worst case concerning how long an end-system needs to wait to send a frame is a ring in which the end-systems take turns sending frames. A ring topology has some drawbacks. In the first place, a break in the ring would bring the whole network down. Furthermore, a ring is a poor fit in linear topologies, like assembly lines (token bus
was developed by people interested in factory automation. The 802.4 standard \cite{ISO904} combines the robustness of the 802.3 broadcast cable and the known worst-case behaviour of a ring.

Token bus uses a linear shared medium but the end-systems are logically organised into a ring. Each end-system knows the addresses of the preceding and succeeding system. The permission to send a frame is passed around the logical ring by sending a special control frame called a token. Since only one end-system holds the token at a time, collisions do not occur. The 802.4 MAC protocol is very complex. It has to deal with end-system joining and leaving the ring and reckon with end-systems breaking down and the accidental appearance of multiple tokens.

The token bus protocol is defined to work at bit-rates of 1, 5 and 10 Mbits/s.

2.3 IEEE 802.5: token ring

Ring networks have many attractive features. On the first place, a ring is not really a broadcast medium, but a collection of individual point-to-point links that form a circle. Point-to-point links involve a well-understood and field-proven technology. A ring is also almost entirely digital, whereas 802.3, for example, has a substantial analog component for collision detection. A ring is also fair and has a known upper boundary on medium access.

In a token ring, a special bit pattern, called the token, circulates around the ring whenever all end-systems are idle. When a system wants to transmit, it seizes the token and removes it from the ring. An end-system may hold the token for the token-holding time. After the end-system has transmitted its frame(s) or the token-holding time has expired, the end-system regenerates the token frame and puts it on the ring. A major drawback of ring networks is their vulnerability: if the cable breaks somewhere, the ring dies. This problem can be solved by the use of a wire centre which contains bypass relays that are released when the ring
breaks or an end-system goes down. An advantage of the 802.5 token ring protocol over the 820.3 Ethernet protocol is that it has an elaborate scheme for handling multiple priority frames.

The standard, which is described in [ISO91a], defines 1 Mbits/s and 4 Mbits/s token ring networks, but a 16 Mbits/s version is currently being used by IBM.

2.4 FDDI/CDDI

Fiber Distributed Data Interface (FDDI), as it is defined by the ISO [ISO89], is a high performance fiber optic token ring LAN running at 100 Mbits/s. It can be used in the same way as any of the 802 LANs, but with its high bandwidth, another common use is as a backbone to connect copper LANs.

The FDDI cabling consists of two fiber rings, one transmitting clockwise and the other transmitting counterclockwise. If one of these rings breaks, the other can be used as a backup. If both break at the same point, the two rings can be joined into a single ring. Each end-system contains relays that can be used to join the two rings or bypass the system in the event of system problems.

The FDDI protocols are similar to the 802.5 protocols. One difference is that FDDI allows a system to generate a token as soon as its frame has left, which results in the possibility of several frames being on the ring at the same time in large rings.

Copper Distributed Data Interface (CDDI), a variation of FDDI, uses the same protocol modified to run over less-expensive copper cabling.

2.5 Ethernet derivatives

Several solutions, derived from Ethernet, that have been developed are conquering their share of the market. Examples are: Fast Ethernet, Switched Ethernet and Isochronous Ethernet. This section shortly describes the features of these solutions.

2.5.1 Switched Ethernet

Technically spoken, Switched Ethernet does not differ from Ethernet. The only difference between the two is that Switched Ethernet uses the Ethernet 802.3 protocol as an access to a switch (hub), thereby providing each end-system with 10 Mbits/s.

2.5.2 Fast Ethernet

Fast Ethernet is an extension of the existing Ethernet IEEE 802.3 standard which simply increases the shared bandwidth from 10 Mbits/s to 100 Mbits/s. Like 10 Mbits/s Ethernet, Fast Ethernet can be configured in a switched environment, thereby providing each end-system with 100 Mbits/s.
2.5.3 100VG/AnyLAN

The 100 Mbits/s 100VG/AnyLAN standard is described in IEEE 802.12 [IEE95]. It supports both 802.3 and 802.5 frame formats. In the 100VG solution, end-systems are connected to intelligent hubs via four cable pairs. By using a demand priority access scheme instead of the CSMA/CD scheme, packet collisions are eliminated and the network bandwidth is used more efficiently. This scheme also permits rudimentary prioritization of time-sensitive traffic, such as real-time voice and video. However, since 100VG assumes it is used between end-systems and intelligent hubs, it is a rather expensive networking solution.

2.5.4 Isochronous Ethernet

Isochronous Ethernet, which is currently being specified as IEEE standard 802.9a [IEE94c], uses fixed cycle Time Division Multiplexing (TDM) to allocate multiple channels on one physical link. It uses one 10 Mbits/s P channel supporting Ethernet, one 6.144 Mbits/s C channel supporting 96 ISDN B channels at 64 kbits/s each, one ISDN D channel and an M (maintenance) channel for signalling, maintenance and timing.

Isochronous Ethernet hubs will support standard Ethernet, so only end-systems that require isochronous capabilities have to be up-graded with isochronous Ethernet Network Interface Cards (NICs).

2.6 Comparison of LANs

The differences between Ethernet, Token bus and Token ring are small. They use roughly similar technologies and get roughly similar performance. The most important advantage of Ethernet is that it is by far the most widely used type at present, with a huge installed base and considerable operational experience. Moreover, it uses a simple algorithm and systems can be installed on the fly, without taking the network down. On the other hand, Ethernet has a substantial analog component and is non-deterministic, which is inappropriate for real-time networking. Also, at high loads, the throughput can be seriously affected by the occurrence of collisions.

Token bus is more deterministic than Ethernet and supports priorities. It also has excellent throughput and efficiency at high load. However, the protocol is extremely complex and has substantial delay at low load since the systems must always wait for the token.

Token ring uses point-to-point connections which makes engineering easy and fully digital. It allows priorities. The major drawback is the presence of a centralized monitor function and the vulnerability of the ring.

FDDI delivers significantly more bandwidth (100 Mbits/s), but does not efficiently handle isochronous traffic. For desktop connections, it is expensive relative to Fast Ethernet.

Fast Ethernet also delivers 100 Mbits/s. For the rest it is identical to Ethernet and, therefore, has the same advantages and drawbacks.
Switched Ethernet provides every end-system with 10 Mbits/s, but uses the same protocol as Ethernet with the same advantages and drawbacks.

Isochronous Ethernet provides both Ethernet and isochronous services. It is, however, still under development. The Asynchronous Transfer Mode (ATM), which will be discussed in chapter 3, provides the same services but can do this at significantly higher bandwidth. ATM development, however, is very slow, which allows Isochronous Ethernet to gain a market share. Isochronous Ethernet is seen as an intermediate solution that will lose its attraction as soon as ATM is fully deployed.
Asynchronous Transfer Mode (ATM)

The Asynchronous Transfer Mode (ATM) is the switching and multiplexing technique chosen for the Broadband Integrated Services Digital Network (B-ISDN). ATM uses an asynchronous time division multiplexing technique to transport short fixed length packets (called cells) over a single physical channel. Cells are switched in the network based on the routing information contained in their headers. Since ATM uses small fixed length cells, the cells can be easily switched by hardware and the queuing and processing delays are relatively small.

Figure 3.1 shows the various interfaces between ATM network elements. The private User-Network Interface (UNI) and the public UNI specifications define how users (end-systems) establish connections to private and public ATM networks. The private Network-Node Interface (NNI) and the public NNI interface define the interfaces between private ATM switches and between public ATM switches. The Broadband Inter-Carrier Interface (B-ICI), finally, defines the framework for networks to share traffic, management and billing tasks.

Figure 3.1 ATM network

3.1 The B-ISDN model

For the development of communication protocol standards, the Open Systems Interconnect (OSI) reference model for communication systems (ISO84) of the International Standardization Organization (ISO) is generally used as a framework (for an explanation of this model, see [TAN89]). The same logical hierarchical architecture as used in OSI is used for the ATM B-ISDN network. However, only the lower layers are explained.
The B-ISDN protocol model for ATM is depicted in Figure 3.2 \[1.321\]. It contains three planes:

- The user plane is used to transport user information.
- The control plane is mainly composed of signalling information.
- The management plane is used for operational and maintenance functions.

A layered approach as in OSI is used for each plane with a high degree of independence between layers. The **physical layer** mainly performs functions on the cell and bit level. Strictly speaking, the ATM physical layer does not conform to the OSI definition, as it deals with cells rather than bits. The **ATM layer** is in charge of transporting cells to their destinations. The **ATM Adaptation layer (AAL)** takes care of the adaptation of the information of the higher layer to the ATM cells. These layers can further be divided into sublayers, each of which performs a number of functions. A description of the functions of the B-ISDN layers, which are contained in \[1.321\] and \[1.413\], is given in Table 3.1.

### 3.2 Physical layer

The Physical Layer consists of two sublayers: the Physical Medium (PM) sublayer and the Transmission Convergence (TC) sublayer.

The **Physical Medium Sublayer** includes only physical medium-dependent functions. Its specification will therefore depend on the used medium. One function common to all medium types is bit timing. This sublayer is responsible for transmitting/receiving a continuous flow of bits with associated timing information to synchronise transmission and reception.
Table 3.1 Functions of the B-ISDN Layers

<table>
<thead>
<tr>
<th>Layer Management</th>
<th>Higher-layer Functions</th>
<th>Higher Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Convergence</td>
<td>CS</td>
</tr>
<tr>
<td></td>
<td>Segmentation and reassembly</td>
<td>SAR</td>
</tr>
<tr>
<td></td>
<td>Generic flow control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cell header generation/extraction</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cell VPI/VCI translation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cell multiplex and demultiplex</td>
<td>ATM</td>
</tr>
<tr>
<td></td>
<td>Cell rate decoupling</td>
<td>TC</td>
</tr>
<tr>
<td></td>
<td>HEC generation/verification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cell delineation</td>
<td>Physical Layer</td>
</tr>
<tr>
<td></td>
<td>Transmission frame adaption</td>
<td>PM</td>
</tr>
<tr>
<td></td>
<td>Transmission frame generation/recovery</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physical medium</td>
<td></td>
</tr>
</tbody>
</table>

The Transmission Convergence Sublayer is responsible for the following functions:

- Transmission frame generation and recovery.
- Transmission frame adaptation. This sublayer is responsible for all actions which are necessary to adapt the cell flow according to the used payload structure of the transmission system in the sending direction. In the opposite direction, it extracts the cell flow from the transmission frame.
- Cell delineation. This mechanism enables the receiver to recover the cell boundaries.
- HEC sequence generation and cell header verification to protect against header errors. The HEC sequence is inserted at the transmitting side. At the receiving side, the HEC value is recalculated and compared with the received value. If possible, header errors are corrected, otherwise the cell is discarded.
- Cell rate decoupling. This mechanism inserts and removes idle cells in order to adapt the rate of ATM cells to the payload capacity of the transmission system.

3.3 ATM layer

The ATM layer, which mainly performs switching/routing and multiplexing, is fully independent of the physical layer. Its principal functions are:

- Transmission, i.e. multiplexing and demultiplexing of cells of different connections onto a single cell stream.
Switching, including header extraction and addition.

Generic Flow Control (GFC).

The unit of transport in ATM is a 53-octet cell, the structure of which is depicted in (Figure 3.3). There are several advantages of using small, fixed-size cells. Firstly, the use of small cells reduces queuing delay for a high-priority cell since its waiting time is less if it arrives slightly behind a lower-priority cell. Secondly, fixed-size cells can be switched more efficiently, which is important for the very high data rates of ATM.

As the figure shows, an ATM cell contains a 5-octet header and a 48-octet information field. The header contains a 4-bit GFC field, an 8-bit VPI field, a 16-bit VCI field, a 3-bit PT field, a 1-bit CLP field and an 8-bit HEC field.

Figure 3.3 ATM cell structure

The Generic Flow Control (GFC) field is used for end-to-end flow control and exists for the User Network Interface (UNI) only. Between nodes in the network, i.e. on a Network-Node Interface (NNI), the GFC is unused. The Virtual Path Identifier (VPI) indicates a user-to-user or user-to-network virtual path. The Virtual Channel Identifier (VCI) indicates a user-to-user or user-to-network virtual channel. These identifiers have local significance only and may change as the cell traverses the network. Since the GFC field is not used on the NNI, the 4 bits of this field are added to the VPI field. So on the NNI, the VPI field is 12 bits long. The Payload Type (PT) field indicates the nature of the data in the payload that follows. User data, for example, would have a different payload type than supervisory information. The Cell Loss Priority (CLP) field is used to provide guidance to the network in the event of congestion. A value of 0 indicates a cell of higher priority, which should not be discarded unless no other alternative is available. A value of 1 indicates that, if necessary, the cell may be discarded by the network.
The ATM service relies on reliable digital circuits for error free communications and therefore provides no error detection or correction for the payload data. In the rare event of an error, it is left to the higher layer protocols to detect the error and retransmit the data. The correct functioning of the ATM layer, however, needs to be guaranteed and therefore the ATM service has to guarantee the validity of the cell header. The Header Error Control (HEC) field is an 8-bit error code that can be used to correct single-bit errors and to detect multiple-bit errors in the header. If a multiple-bit error occurs, the whole cell is discarded.

3.4 ATM adaptation layer

The ATM Adaptation Layer (AAL) is located between the ATM layer and the higher layer. Its basic function is the adaptation of services provided by the ATM layer to the requirements of the higher layer. To minimize the number of AAL protocols, the ITU-T has defined four classes of service, which are described in [I.362] and [I.363]. Table 3.2 depicts the AAL classes. The classification is based on whether a timing relationship must be maintained between source and destination, whether the application requires a constant bit-rate, and whether the transfer is connection-oriented or connection-less. Class A services correspond to circuit emulation (e.g. transport of a 2 Mbits/s or 45 Mbits/s signal) or constant bit-rate video. Class B services correspond to variable bit-rate video and audio. Classes C and D correspond to data transfer applications, which may be either connection-oriented (Class C) or connection-less (Class D).

Table 3.2 Service Classes for AAL

<table>
<thead>
<tr>
<th>Timing relation between source and destination</th>
<th>Class A</th>
<th>Class B</th>
<th>Class C</th>
<th>Class D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bit rate</td>
<td>Constant</td>
<td>Variable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection mode</td>
<td>Connection-oriented</td>
<td>Connection-less</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The ATM Adaptation Layer consists of two sublayers: the Segmentation And Reassembly (SAR) sublayer and the Convergence Sublayer (CS). The Convergence Sublayer is service-dependent and defines the services that AAL provides to higher layers. Each application attaches to AAL through a Service Access Point (SAP).

The Segmentation and Reassembly sublayer is responsible for the segmentation of higher layer information into a size suitable for the information field of an ATM cell and the reassembly of the contents of a sequence of ATM cell information fields into higher layer information on reception. At the ATM layer, each cell consists of a 5-octet header and a 48-octet information field. Thus, the SAR layer must pack any SAR headers and trailers plus CS information into 48-octet blocks.
At first, the International Telecommunication Union (ITU), formerly CCITT, defined four AAL protocols (AAL1-AAL4), one protocol for each class of service [I.363]. Later, the ITU defined a fifth protocol in addition to these four protocols. This fifth protocol (AAL5) is a highly efficient data transfer protocol and can be used for class C and class D services. Study group 13 of the ITU is currently working on a new AAL1 and AAL2 specification named I.363.X [I.363X]. The objective of this specification, of which the current status is "preliminary draft", is to fill in the "for further study" gaps of recommendation I.363. The next five sections explain the five protocols that have been defined.

3.4.1 AAL type 1

AAL type 1 is used to transfer constant bit-rate data between source and destination when a timing relation between source and destination is required (service class A).

At the transmitting end, the SAR sublayer accepts a 47 octet block of data from the CS and then prepends a one octet SAR-PDU header to this block. At the receiving end, it receives a 48 octet block from the ATM layer and then separates the SAR-PDU header. The 47-octet block of SAR-PDU payload is passed to the CS. The SAR-PDU header consists of two 4-bit fields. The first field contains a Sequence Number (SN), allowing to detect lost or mis-inserted cells. This SN-field is protected by a Sequence Number Protection (SNP) field. The AAL1 SAR-PDU is depicted in Figure 3.4.

![SAR-PDU format for AAL type 1](image)

**Figure 3.4** SAR-PDU format for AAL type 1

The functions of the Convergence Sublayer (CS) of AAL1 strongly depend on the service to be supported. Some functions are:

- Handling of cell delay variation.
- Further handling of lost and mis-inserted cells if the 4-bit SN field in the SAR sublayer is inadequate.
- Source clock frequency recovery using the Synchronous Residual Time Stamp Method (SRTS). This method is described in [I.363] and [I.363X].
- Transfer of structure information between source and destination, for example, to support 8 kHz-based frame formats in circuit-mode services. This method is described in [I.363] and [I.363X].
- Forward error correction. This method may be used to ensure high quality for some video and audio applications. An example of this method, using a Reed-Solomon code, is described in [I.363] and [I.363X].
3.4.2 AAL type 2

AAL type 2 is proposed for variable bit-rate services with a timing relation between source and destination. Since the source is generating a variable bit-rate, it is possible that cells are not completely filled. Therefore, additional functions are required in the SAR. The definition of AAL type 2 is still in progress, so this section only presents a list of functions as mentioned in 1.363X. The following functions may be performed by AAL type 2:

- Segmentation and reassembly of user information.
- Handling of cell delay variation.
- Handling of lost and misinserted cells.
- Source clock frequency recovery at the receiver.
- Recovery of the source data structure at the receiver.
- Monitoring and handling of bit errors.

3.4.3 AAL type 3/4

Initially, separate AALs were defined for class C and class D services (connection-oriented and connection-less services). In the meantime, both types have merged, thereby supporting both service classes.

<table>
<thead>
<tr>
<th>ST</th>
<th>SN</th>
<th>MID</th>
<th>SAR-PDU payload</th>
<th>LI</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 bits</td>
<td>4 bits</td>
<td>10 bits</td>
<td>44 octets</td>
<td>6 bits</td>
<td>10 bits</td>
</tr>
</tbody>
</table>

*Figure 3.5 SAR-PDU of AAL type 3/4*

The SAR sublayer provides segmentation and reassembly of variable length CS-PDUs, by providing an indication of the cell type and an indication of the number of useful octets (Figure 3.5). The cell type is indicated by the Segment Type (ST) field. This 2-bit field indicates the Beginning Of Message (BOM), Continuation Of Message (COM), End Of Message (EOM), or Single Segment Message (SSM). SSM indicates that the SAR-PDU contains an entire CS-PDU. If a CS-PDU needs to be distributed over multiple SAR-PDUs, the ST field of the first SAR-PDU indicates the BOM, the ST field of the last SAR-PDU indicates the EOM and the ST fields of all the intermediate SAR-PDUs indicate the COM.

The LI field indicates the length of the message in the SAR-PDU. PDU sequence and PDU bit error protection is provided by respectively the SN field and the CRC field. The Multiplexing IDentifier (MID) allows the multiplexing of several AAL connections in a single ATM connection.
The CS is split into a common part (CPCS) and a service-specific part (SSCS). The SSCS is application dependent and may be null. Its function are for further study. The CPCS transfers user data frames with any length from 1 to 65535 octets.

3.4.4 AAL type 5

AAL type 5 is also known as the Simple and Efficient Adaptation Layer (SEAL). It is a fairly new, but well defined AAL that offers improved efficiency over AAL type 3. It serves the same purpose, but by assuming that the current higher layer processes will provide error recovery, the SEAL simplifies the adaptation process to pack all 48 octets of the information field with data.

Segment-type information is carried in the PT field in the ATM header and is called ATM-User-to-User information (AUU). With this coding scheme the SAR sublayer needs no header or trailer. Two types of SAR-PDUs are defined (Figure 3.6). AUU=0 indicates any AAL5 cell that is not an EOM or SSM cell. All 48 octets of the information field are filled with data (and padding if the data is less than 48 octets, but more than 40 octets). AUU=1 indicates an EOM or SSM cell. The first field in the cell is 40 octets long and contains data and possibly padding. The UU (User to User) field is used to transparently transfer CPCS user to user information. The use of the Common Part Indicator (CPI) field is for further study, but may be used to align the CPCS-PDU trailer to 64 bits. The cell also contains a 32-bit CRC field and a Length field. AAL5 does not support a multiplexing function because it does not provide a MID.

![Figure 3.6 Coding of AAL Type 5.](image)

3.5 ATM connections

ATM cell transport requires a connection. This connection can be set up statically at subscription time or dynamically by using a signalling protocol. A major advantage of the connection-oriented approach of ATM is the reduced functionality of the header. Due to this reduced
CHAPTER 3: ASYNCHRONOUS TRANSFER MODE

functionality, the implementation of the header processing in the ATM nodes is simple and can be done at very high speeds. This results in very low processing and queuing delays. A major disadvantage, however, is the need for connection setup, which complicates the implementation of management and signalling. ATM connections always guarantee delivery of cells in sequence and can be point-to-point or point-to-multipoint.

The ATM layer has two hierarchical levels, namely a virtual path level and a virtual channel level. Both are defined in [1.113].

A Virtual Channel (VC) is "a concept, used to describe unidirectional transport of ATM cells associated by a common unique identifier value". This identifier is called the Virtual Channel Identifier (VCI) and is part of the cell header. A Virtual Path (VP) is "a concept used to describe unidirectional transport of cells belonging to virtual channels that are associated by a common unique identifier value". This identifier is called the Virtual Path Identifier (VPI) and is also part of the cell header.

Figure 3.7 shows the relationship between virtual channels, virtual paths and transmission paths. A transmission path may contain several virtual paths, each of which may contain several virtual channels.

Concerning virtual channels and virtual paths, a distinction is made between links and connections [1.113]. A Virtual channel link is "a means of unidirectional transport of ATM cells between a point where a VCI value is assigned and the point where that value is translated or removed". Similarly, a virtual path link is terminated by the points where a VPI value is assigned and translated or removed.

A concatenation of VC links is called a Virtual Channel Connection (VCC), and a concatenation of VP links is called a Virtual Path Connection (VPC). Although VCCs and VPCs are defined to be unidirectional, they are always set up in pairs of two. Moreover, these two connections must have identical identifiers so in practice a bi-directional connection is created.

The virtual path concept allows the switching of bundles of virtual channels as one unit. A virtual path connection (cross-connect) may be set up between two switches via management
or signalling. After this VPC has been set up between two switches, intermediate switches are no longer involved in connection control.

![Figure 3.8 Virtual paths and virtual channels](image)

The interworking of VP switches and VC switches is illustrated in Figure 3.8. In this figure, virtual path connections exist between End-System (ES) A and the VC-switch (T), and between the VC-switch and ES B. ES A wants to setup a VC connection to ES B using those two VP connections. The network has to provide a VCI value (A1) for the A to T link, and a VCI value (A2) for the T to B link. The VC connection from A to B is thus made of two VCI links only. At switching points S1 through S4, only the VPI field is changed. At the switching point T, both VPI and VCI fields are changed. As illustrated in this example, the VCI field is transported unchanged through the entire virtual path connection.

### 3.6 ATM LAN emulation

ATM LAN emulation, as defined by the ATM Forum [ATM95a], is an entity in the ATM network that allows today’s LANs like Ethernet and Token-Ring to interwork with the ATM network, and allows existing LAN application software to run on an ATM network. To achieve this, the LAN emulation entity emulates services of existing LANs across an ATM network.

End-systems can connect to the ATM network while the software applications interact as if they are attached to a traditional LAN. Also, existing LANs can be interconnected with ATM networks by means of today’s bridging methods, which allows interoperability between
software applications residing on ATM-attached end-systems and on traditional LAN end-systems. The goal of ATM LAN Emulation (LE) is to eliminate the need for changes in existing higher layer protocols.

The LAN-specific characteristics to be emulated are:

• Connection-less services. LANs transport data without setting up connections, while ATM needs to set up a connection before data can be transported. The LAN Emulation entity sets up the necessary connections and provides a connection-less service to the upper layer protocols.

• Broadcast and multicast services. These facilities are easily accomplished through the shared medium of a LAN. ATM, on the other hand, does not have such a shared medium. Therefore, support of broadcast and multicast facilities represent an additional demand on the hardware and software architecture. This does not necessarily mean that all messages addressed to a multicast MAC address must be distributed to every station. A service could be established to intercept these messages and forward them directly to their destinations instead of broadcasting them to every station.

• MAC driver interfaces. These interfaces enable existing applications to access an ATM network via protocol stacks like NetBIOS as if they were running over traditional LANs. Since these protocol stacks communicate with a MAC driver, the LAN emulation has to offer the same MAC driver service primitives. This means providing interfaces and services that comply with common interface specifications like Open Data-link Interface (ODI), Network Driver Interface Specification (NDIS) and Data-Link Provider Interface (DLPI).

• Virtual LANs. The LAN Emulation entity must allow the logical definition of groups of devices called Virtual LANs (VLANS). A VLAN is a collection of end-systems (PCs, workstations, servers) intended to mirror organisational groupings without regard to network topology. Unlike real LANs, VLANs allow the membership to be independent of the systems physical location. Moreover, an end-system can subscribe to multiple emulated LANs. End-systems within a VLAN communicate freely (using MAC-level bridging), while the communication between users of separate VLANs requires a router to translate addresses and enforce security. Broadcast frames are distributed only to the members of that emulated LAN.

• Interconnection with existing LANs, which includes connections between ATM stations (that use LAN protocols on top of ATM protocols) and LAN stations as well as connections between LAN stations across an ATM network.

Emulated LANs are composed of a set of LAN emulation clients and a single LAN Emulation service which comprises three logical servers, namely the LAN emulation server, the broadcast and unknown server and the LAN emulation configuration server. Each client represents a set of users, identified by their MAC address. The LAN emulation service may
be part of an end-station or a switch; it may be centralized or distributed over a number of stations.

The **LAN Emulation Client (LEC)** is the entity in end-systems that performs data forwarding, address resolution and other control functions. It provides a MAC level interface to higher level software.

The **LAN Emulation Server (LES)** implements the control coordination function for the emulated LAN. It provides a facility for registering and resolving MAC addresses and/or route descriptors to ATM addresses.

The **Broadcast and Unknown Server (BUS)** handles data sent by a LEC to the broadcast MAC address, all multicast traffic and initial unicast frames of which the MAC address has not been resolved to a direct ATM connection.

The **LAN Emulation Configuration Server (LECS)** implements the assignment of individual LECs to different VLANs. Based upon its own policies, configuration database and information provided by the LEC, it assigns any client which requests configuration information to a particular emulated LAN service by giving the client the LES’s ATM address.

When a LEC wants to join a VLAN, it approaches the LECS. The LECS then gives the LEC the ATM address of the LES of the VLAN it wants to join, after which the LEC registers itself with this LES. The LEC is now a member of the VLAN and can send data to other

![LE connections](image-url)
LECs. When a LEC has a packet to send to another LEC, it sets up a point-to-point VCC to that LEC. If the LEC does not know the ATM address of the destination, it sends an LE-ARP (LAN Emulation Address Resolution Protocol) request to the LES. While the ATM address of the destination is being resolved and the connection is being set up, the LEC sends the packets destined for the destination to the BUS which broadcasts it to all LECs of the VLAN. This is done to prevent the packets from having to wait until the ATM address of the destination has been resolved and the point-to-point VCC has been set up.

To communicate with the servers, LECs use two types of connections (Figure 3.9). Control connections are used for housekeeping tasks such as finding the ATM address of another client; data connections are used for everything else. The bi-directional Configuration Direct VCC is set up by the LEC as part of the configuration phase and is used to obtain the ATM address of the LES from the LECS. The LEC does not have to maintain this VCC after completing this phase. The Control Direct VCC, which is set up by a client when it joins an emulated LAN, runs between the client and the LES. It is a bi-directional point-to-point link. Optionally, a unidirectional Control Distribute VCC (point-to-point or point-to-multipoint) may be set up by the LES to the client.

Data connections are used to connect one client to another and to link clients to the BUS. The former is a bi-directional link known as a Data Direct VCC, the latter is a pair of unidirectional links carrying data from the client to the BUS (Multicast Send VCC) and from the BUS to the client (Multicast Forward VCC). The Multicast Forward VCC can be either a point-to-multipoint VCC or a unidirectional point-to-point VCC.

3.7 Summary

ATM is the switching and multiplexing technique chosen for the B-ISDN. It uses an asynchronous time division multiplexing technique to transport short fixed length packets called cells over a single physical channel.

The ATM protocol model identifies three planes: the user plane, the control plane and the management plane. Each plane is modelled on the basis of the OSI reference model and consists of three layers: the physical layer which mainly performs functions on the bit level, the ATM layer which transports cells to their destination, and the ATM Adaptation Layer (AAL) which adapts the information of the higher layer to the ATM cells.

Five AAL protocols have been defined. AAL1 is used for constant bit-rate video or circuit emulation, while AAL2 supports variable bit-rate video and audio. AAL3 and AAL4 were designed to support data transfer applications, but have become outdated by the development of AAL5. AAL5 offers improved efficiency over AAL3 and AAL4.

ATM is a connection oriented technology. Two hierarchical levels exist for these connections. Virtual Channel Connections (VCCs) are the elementary connections between two ATM end-systems. Virtual Path Connections (VPCs) are used transport bundles of VCCs as one unit.
To allow today's LANs like Ethernet and Token-Ring to interwork with an ATM network, and to allow existing LAN application software to run on an ATM network, a LAN Emulation (LE) entity has been defined by the ATM Forum. This entity emulates services of existing LANs across an ATM network.
4 ATM switching techniques

This chapter describes issues regarding ATM switching techniques. A switch can be divided into two parts: a control part and a transport part. The transport part is defined as all physical means responsible for the correct transportation of the information (i.e. ATM cells) from an inlet to an outlet, within the Quality of Service (QoS) specifications of ATM. This part mainly performs functions located in the user plane of the protocol reference model described in section 3.1. The control part of the switch decides which inlet to connect with which outlet on the basis of signalling or set by an operator on a semi-permanent basis. This part mainly performs functions in the control plane of the protocol reference model described in section 3.1. In this chapter only attention will be paid to the transport part of the switch.

4.1 Principle of an ATM switch

The basic principle of an ATM switch is shown in Figure 4.1 [PRY91]. Incoming cells are physically switched from an inlet $I_i$ to an outlet $O_j$. At the same time the header value is translated from an incoming value $\alpha$ to an outgoing value $\beta$.

![ATM switching principle](image)

Multiplexing techniques can be used to concentrate traffic from a certain number of inlets to a smaller number of outlets; demultiplexing techniques are used to perform the reverse operation. Because cells from different inlets that are destined for the same outlet may arrive simultaneously at the switch, a queue has to be provided to store cells which cannot be forwarded immediately.
So a switch needs to perform three tasks: label translation, routing cells from an inlet to an outlet and queuing. The distinction between different switches is made by the specific implementation of these tasks and where they reside in the switch.

4.2 Implementation aspects of switches

4.2.1 Transfer media

Every ATM switch contains a buffer space. The part of the switch that transfers cells from an inlet to the buffer and from the buffer to the outlet is called the switching transfer medium. Transfer media can be categorised into three sorts [AUG93].

The first transfer medium is the matrix of slotted buses. Every inlet is connected to its own physical bus, which is connected to all outlets. This medium, which is depicted in Figure 4.2a, has the advantage of operating at the same speed as an inlet, but requires a large hardware complexity.

Figure 4.2b shows the second transfer medium: the Time Division Multiplexed bus (TDM-bus). All inlets are connected to all outlets (or to a buffer space) via a single bus operating at N times the speed of an inlet/outlet (N being the number of inlets/outlets). The advantage of using the TDM-bus is the decrease in hardware complexity. A disadvantage is that the bit-rate on the TDM-bus needs to be extremely high when this medium is used to switch high-speed traffic. The reflections on the bus caused by this high bandwidth needs to be suppressed.

The third transfer medium is the multistage interconnection network (Figure 4.2c). If the switch is composed of multiple switching elements, the network that interconnects these switching elements is the transfer medium.
4.2.2 Queuing methods

There are mainly three ways of queuing data in a switch. The distinction is based on the physical location of the queues: input queuing, output queuing and central queuing [PRY91].

The input queuing solution solves the possible contention problem at the input (Figure 4.3a). Each cell is stored in a queue dedicated to an inlet until the arbitration logic decides that the queue may be served. The switching transfer medium then transfers the ATM cells from the input queue to the outlet(s) without contention. The arbitration logic can range from very simple, like round-robin, up to quite complex, e.g. taking into account the filling levels of the input queues.

An important disadvantage of input queuing switching elements is the so-called head of line blocking. If cells from two queues are destined for the same outlet, one of the queues is blocked. Blocking this queue causes every cell in the queue to be delayed, even if it is destined for an outlet that is currently free. This significantly decreases the performance.

The output queuing approach, allows cells of different inlets destined for the same outlet to be transferred during one cell time. The possible output contention is solved by queues which are located at each outlet of the switching element (Figure 4.3b). To ensure that no cells are lost inside the transfer medium, the cell transfer must take place at N times the speed of the inlets (N being the number of inlets). No arbitration logic is required since all cells can go to their respective output queue.

The central queuing solution shares one memory space between all inlets and outlets (Figure 4.3c). Cells are directly stored in the central queue and each outlet selects the cells which are destined for it from the central memory. This type of queuing requires a complex memory management system that allows random access to the central memory, but preserves the cell sequence for each outlet. The memory space can be partitioned statically into multiple queues or dynamically. Dynamically partitioning the space allows for smaller queues, since they can be enlarged if necessary. Therefore, this method is the most efficient but also the most complex.

A switch composed of multiple stages may combine more than one of these three methods.

4.2.3 Routing methods

Usually, a switch is composed of multiple smaller switching elements. Each time a cell reaches a switching element, a routing decision has to be made. Two methods may be used when routing cells in a switch [AUG93]. The first method uses a table every time a routing decision is made. This kind of routing is flexible and multicast and broadcast functions can be implemented easily, but a lot of memory space is needed for the tables. In the alternative method, the first switching element appends extra information to every arriving cell so that the remaining switching elements can make routing decisions on the basis of this information. This extra information is called a routing tag and is determined on the basis of a table at the entrance of the switch. This method has the advantage that only the first switching element needs a table and the routing control in the remaining switching elements can be very simple.
CHAPTER 4: SWITCHING STRUCTURES

(a) Input queueing

(b) Output queueing

(c) Central queueing

Figure 4.3 Switching element with input, output and central queues
Another issue regarding the routing mechanism is whether the selection of the route that cells take through the switch is determined for the duration of a connection (connection-oriented) or every time a cell arrives (connection-less). Connection-oriented routing has the advantage that the cell sequence is preserved, but traffic is not distributed evenly over the switch, so the resources are not optimally utilized. Connection-less routing has the advantage of more evenly spread traffic (and therefore optimal utilization of resources) but needs cell sequence recovery at the outlets because cells may experience different delays.

These two fundamental choices (i.e. whether or not to use a routing tag and whether or not to select a route for the duration of a connection) lead to four possible routing methods. It is also possible to use combinations of these methods in one switch.

4.3 Summary and conclusions

An ATM switch needs to perform three tasks: label translation, routing and queuing. The distinction between the different switches is made by the specific implementation of these tasks and where they reside in the switch.

The part of the switch that transfers ATM cells to the queues or to the outlets is called the transfer medium. Three examples of transfer media are the matrix of slotted busses, the TDM-bus and the multistage interconnection network. If the reflection problems caused by the high bit-rate on the bus can be solved, the TDM-bus is an ideal transfer medium due to its low hardware complexity and non-blocking property.

The queuing of cells inside a switch can be performed in three manners, namely input queuing, output queuing and central queuing. The output queuing method is preferred over the input queuing method because it gives better performance. The central queuing method uses one memory space to implement the queues and therefore requires less memory but introduces a high amount of complexity. The memory space can be partitioned statically or dynamically. The latter method is the most efficient but also the most complex.

Four methods may be used when routing cells in a switch. The distinction between these methods is based on two fundamental choices. The first choice is whether to attach a routing tag to every cell at the entrance of the switch so that no extra routing control is needed inside the switch or to use tables every time a routing decision is made. The other choice is whether cell routing is determined for the duration of a connection or every time a cell arrives. The latter possibility spreads traffic more evenly over the switch, but needs cell sequence recovery at the outlets.
5 Multimedia networking requirements

5.1 Multimedia networking

A multimedia infrastructure must be capable of replacing the two current communication infrastructures. For the support of telecommunication applications, like telephony, it must provide an explicit timing relationship between the sending and receiving of data. Moreover, the end-to-end delay of the data must be small and constant and the jitter must be small. To support datacommunication applications, like file transfer, the delay is of minor importance, but the data must traverse the network without being mutilated. Apart from the above mentioned applications, a multimedia network must also support multimedia services like real-time video conferencing. Multimedia applications typically require both real-time services and large amounts of bandwidth.

These different applications can be supported by the network by providing services characterised by several attributes. These attributes are:

- User bit-rate.
- Traffic flow (Constant Bit-Rate (CBR), Variable Bit-Rate (VBR), burstiness, ...).
- Communication configuration (point-to-point, point-to-multipoint, multipoint-to-multipoint).
- Symmetry (uni-directional, bi-directional asymmetric, bi-directional symmetric).
- Connection mode (connection-oriented, connection-less).
- Quality of Service (QoS) parameters.
- Residual error rate.

The Quality of Service (QoS) is a concept for specifying how "good" network services are. An application that needs a certain guarantee of service from the network, negotiates a traffic contract with this network. The best set of QoS parameters in this contract still needs to be determined, but commonly used parameters are end-to-end delay, jitter in the end-to-end delay, peak bit-rate, average bit-rate and burstiness.

Table 5.1 [ROY94] shows an overview of several audio and video applications and some of their attributes. Table 5.2 [ROY94] shows a similar overview for data applications. These lists are by no means complete, but serve to give an impression of QoS requirements.
Table 5.1  Characteristics of audio and video services

<table>
<thead>
<tr>
<th>Service</th>
<th>Uncomp. bit-rates in kbits/s</th>
<th>Transm. mode</th>
<th>Expected bit-rates in kbits/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD audio</td>
<td>1411.4-1536</td>
<td>CBR</td>
<td>Peak: 192</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VBR</td>
<td></td>
</tr>
<tr>
<td>FM stereo audio</td>
<td>1024-1536</td>
<td>CBR</td>
<td>Peak: 128</td>
</tr>
<tr>
<td>PCM audio</td>
<td>64</td>
<td>CBR</td>
<td>Peak: 64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VBR</td>
<td></td>
</tr>
<tr>
<td>ADPCM audio</td>
<td>64</td>
<td>CBR</td>
<td>Peak: 32</td>
</tr>
<tr>
<td>LD-CELP audio</td>
<td>64</td>
<td>CBR</td>
<td>Peak: 16</td>
</tr>
<tr>
<td>MPEG-1 video</td>
<td>24,883</td>
<td>CBR</td>
<td>Peak: 768-1500</td>
</tr>
<tr>
<td>360x288 30 Hz (8 bits/sample)</td>
<td></td>
<td>VBR</td>
<td>239.6</td>
</tr>
<tr>
<td>MPEG-1 video</td>
<td>82,944</td>
<td>CBR</td>
<td>Peak: 4000-9000</td>
</tr>
<tr>
<td>720x480 30 Hz (8 bits/sample)</td>
<td></td>
<td>VBR</td>
<td>2000-4000</td>
</tr>
<tr>
<td>MPEG-1 video</td>
<td>41,472</td>
<td>CBR</td>
<td>Peak: 2000-4000</td>
</tr>
<tr>
<td>360x480 30 Hz (8 bits/sample)</td>
<td></td>
<td>VBR</td>
<td></td>
</tr>
</tbody>
</table>

5.2 General requirements for a multimedia networking solution

The requirements regarding a customer premises multimedia networking solution are reflected by the following terms:

- **Flexibility**: A network preferably supports interconnection with existing networks and provides all services supported by these networks.

- **Migration**: A network must provide for a method to migrate from existing network architectures to the new architecture.

- **Scalability**: A network must be suitable for small and large environments and be scalable.

- **Simplified network**: The network preferably has a simple architecture and must be easy to manage.

- **Lower cost**: The network must be competitive to other networking solutions.

- **More bandwidth**: Current networking solutions supply limited bandwidth. Most of the time, this bandwidth is not guaranteed and shared with other users. Because more and more bandwidth demanding applications like video conferencing and
Table 5.2 Characteristics of data services

<table>
<thead>
<tr>
<th>Data</th>
<th>Uncompr. size in Mbits/s</th>
<th>Typical compr. ratio</th>
<th>Peak bandwidth requirements in Mbits/s</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Retrieval and transfer (2 seconds response time)</td>
</tr>
<tr>
<td>ASCII text (A4)</td>
<td>0.029</td>
<td>2-4</td>
<td>0.015</td>
</tr>
<tr>
<td>A4 color page (200 pixels/inch, 24 bits/pixel)</td>
<td>90</td>
<td>10-20</td>
<td>45</td>
</tr>
<tr>
<td>A4 color page (400 pixels/inch, 24 bits/pixel)</td>
<td>359</td>
<td>10-20</td>
<td>180</td>
</tr>
<tr>
<td>A4 color page (800 pixels/inch, 24 bits/pixel)</td>
<td>1436</td>
<td>10-20</td>
<td>718</td>
</tr>
<tr>
<td>A4 color page (1600 pixels/inch, 24 bits/pixel)</td>
<td>5744</td>
<td>10-20</td>
<td>2872</td>
</tr>
</tbody>
</table>

CAD/CAM arise, end-users need more bandwidth than before. However, not all users will need more bandwidth. Therefore, the network ideally provides every user with the bandwidth it needs. For delay-sensitive applications like voice and video, bandwidth needs to be guaranteed.

- **Multimedia:** Currently, users are starting to demand the integration of voice, video and data into a single computer environment and networking solution. The integration of different services in a single network requires the network to support multiple guaranteed classes of service.

The first step in the development of a customer premises multimedia networking solution is the selection of a network technique that is capable of meeting all these requirements. The subsequent section will demonstrate why ATM is the right solution.

### 5.3 Why ATM?

A major issue in the development of a multimedia network is the choice of networking technique. A large variety of networking techniques currently exist, ranging from 10 Mbits/s shared Ethernet to 100 Mbits/s FDDI. However, the problem is that all these techniques, which have been described in chapter 2, fail as multimedia networking solution.
Ethernet, Token bus and Token ring and switched Ethernet simply provide insufficient bandwidth for multimedia applications. Fast Ethernet, 100VG-AnyLAN, FDDI and CDDI provide sufficient bandwidth (for the time being), but are incapable of supporting real-time services. Isochronous Ethernet is capable of supporting real-time services, but provides insufficient bandwidth for true multimedia applications like video distributing. Another major drawback of the above mentioned techniques is that they are only suitable for local areas. Wide area networks simply require a new switching technology like ATM to support multimedia networking. For this reason, a lot of effort is being put into the development of ATM which will eventually lead to ATM becoming the standard for wide area multimedia networking. This will significantly lower the price of ATM hardware, thereby making it an attractive solution for local area multimedia networking too.

The main advantage of ATM is that it uses small fixed-length cells, which significantly simplifies switching and therefore reduces queuing and processing delays. Other advantages are that ATM supports multiple bit-rates, uses the available bandwidth very efficiently, is capable of transporting video, voice and data, and can provide a lot of bandwidth, if needed. For local area networks, ATM has also two important disadvantages, namely that it requires a whole new infrastructure and that it is still under development. However, no alternative solution exists for supporting real-time services which is compatible with the current developments in wide area networks. Therefore, ATM is still seen as the most appropriate networking technique to achieve the requirements described above.

5.4 ATM network requirements

Since ATM is the main networking technique, it must meet the general requirements of section 5.2. This section describes how each of these requirements can be met and how they are translated into more specific requirements for the ATM network to be designed.

5.4.1 Flexibility and migration

Apart from ATM end-systems, the network must also provide interworking with traditional LANs like Ethernet and Token-Ring. Since Ethernet forms the main part of existing LANs, this is the most important technology to interwork with. Interworking with LANs is provided through ATM LAN Emulation (LE), as described in section 3.6.

The support of connection-less services is required. Since ATM is connection-oriented, these services have to be realized on top of ATM. The network must also support multicasting. In shared-medium LAN technologies multicasting is essentially free. In switch-based networks, it represents an additional demand on the switch hardware and software architecture.

At the user interface, the ATM network needs to support existing protocols. The ATM network must provide MAC driver interfaces to enable existing applications to access the network via protocol stacks like NetBIOS as if they were running over traditional LANs. Since these protocol stacks communicate with a MAC driver, the LAN emulation entity, which provides the support of MAC driver interfaces, has to offer the same MAC driver service primitives. When using Microsoft Windows-based Personal Computer (PC), this means providing NDIS interfaces and services.
The flexibility of a network is also increased if standardised interfaces are used between the components of the switching units. This allows the use of third-party hardware and software to implement certain elements of the switch (e.g. line interface circuitry).

5.4.2 Scalability

The architecture of a switch must be distributed and modular to facilitate scalability and to allow easy expansion of the switch and replacement of components.

5.4.3 Simplified network

Current networks consist of many routers and bridges. A more structured network can be created by replacing these networking elements with a distributed switch.

The use of a single management framework also simplifies the network. Simple Network Management Protocol (SNMP) is the de-facto standard for network management, so SNMP agents should be provided for all ATM devices.

5.4.4 Lower cost

There are several opportunities to reduce the cost of an ATM networking solution. In the first place, cost reduction in the end-system could be achieved by implementing the ATM protocol by ATM specific software running over currently available low-cost hardware. In particular, ATM software could be used on top of Ethernet physical interfaces which are not only inexpensive but also have a huge installed base. The major penalty to accept is the bandwidth reduction to 10 Mbits/s. However, according to [ARM95], most end-users do not need more bandwidth than 10 Mbits/s for at least the next decade. Moreover, the use of low bit-rates permits the use of UTP-3 cabling, which may avoid expensive rewiring of buildings.

The second cost reducing opportunity lies in the implementation of ATM multipoint connections in the ATM switching units. ATM switches that internally use the shared-bus principle facilitate the implementation of multipoint connections. Another advantage of the shared-bus principle is that this solution requires relatively few components, which also lowers the cost. Also, the implementation of virtual LANs may differ from the solution proposed by the ATM-Forum [ATM95a], as long as interaction with ATM networks outside the corporate network is provided. The data-direct connections between LECs may be replaced by a single multipoint-to-multipoint connection. To prevent the network from bothering end-systems with data not destined for them, an embedded bridging mechanism is required in every switch that selectively broadcasts the data to other members of the same VLAN.

The third opportunity lies in handling of synchronous traffic, especially PCM coded voice. Switches that internally operate isochronously facilitate QoS guarantee and avoid the necessity of retiming circuitry for isochronous services (AAL 1).

Finally, the support of third-party interface hardware as described in section 5.4.1, also reduces the cost because it is likely that companies specialised in the production of this
hardware are able to produce at lower cost. If these interfaces could be integrated in the switches, the cost of these switches would decrease.

5.4.5 More bandwidth

ATM supports multiple physical interfaces and bit-rates. This allows a network to provide extremely high bit-rates to users that need it and low bit-rates to other users. Furthermore, ATM can use the available bandwidth very efficiently.

5.4.6 Multimedia

A multimedia network must support multiple guaranteed classes of service. The three types of traffic, i.e. voice, video and data, all have their own specific transport requirements. The network must minimise the delay of voice and video and, even more important, the variation in the delay (jitter). If this requirement cannot be met for a particular voice-fragment or video-fragment, it must be discarded. The requirements of the transport of data are more or less opposite to those of the transport of voice and video. The delay may vary, but data must not be mutilated or lost.

An ATM network transports the three types of data in different manners. Voice is transported through the ATM AAL1 service for constant bit-rate traffic. Video is transported through the ATM AAL1 service or through the ATM AAL2 service for variable bit-rate traffic. Data is transported through the ATM AAL5 service for asynchronous traffic. In the future, additional AALs may be introduced (for example for MPEG data transport).

Along with a multimedia network go multimedia end-systems. These end-systems need at least a native ATM application programming interface (API) for access to ATM’s unique capabilities, such as bandwidth reservation and AAL selection. This API must be provided for popular hardware platforms and operating systems. In particular, an API must be provided for Microsoft Windows-based PCs.

5.4.7 Other requirements

The network must also meet several requirements regarding performance, availability and reliability. For ATM networks, these requirements have not yet been expressed in figures. Therefore, this section only presents a number of issues.

The performance requirements of an ATM network may be expressed in terms of the percentage of inadequately handled connection attempts (ca. $10^{-2}$), the end-to-end delay (for voice: mean approximately 300 ms, 95% $\leq$ 400 ms) and the performance during overload (no figures yet).

The availability of a network is defined as the percentage of the time that the network is operating normally. Possible requirements are that the mean accumulated down-time decreases when the number of systems increases and that the mean accumulated down-time per system $\leq$ 30 minutes per year.
Reliability is defined as the number of failures per year. Possible requirements are that the number of failures need to be less than 2 per 100 extensions per year and that a total failure of the network may occur only once every 20 years (source: Dutch PTT).

To meet these requirements, the following architectural requirements are defined:

- Redundancy & sectionalization to minimize impact of failure. This improves the reliability of the network.

- System Assurance Software (SAS): fault detection, fault isolation, system restart, periodical recovery of resources, fault reporting. The SAS keeps the network up as long as possible.

- Design rules (safety, dissipation, fan-in/fan-out, crosstalk, layout etc.) to improve reliability.

- Easy reconfiguration. No administrative intervention should be required for connecting or disconnecting an ATM host. Network elements should be added and removed without taking down the network.

- Network management (fault management, performance management, configuration management).

- No-reset software updating and downloading.

5.5 Summary and conclusions

A multimedia networking solution must meet a lot of requirements. Since ATM is the switching technique that can meet these requirements best and the future B-ISDN is based on it, it is the most suitable technique for the customer premises network. Naturally, interworking with existing networks is also required. Since Ethernet forms the main part of the current data networks, this is the most important data network type to interwork with.

The requirements of an ATM-based customer premises network are:

- Support of all sorts of traffic (voice, video, data) through multiple guaranteed classes of service, represented by the AAL classes AAL1, AAL2 and AAL5.

- Support of popular hardware platforms and operating systems by providing an ATM API for access to ATM's unique capabilities, such as bandwidth reservation and AAL selection.

- Support of interworking with other networks, especially Ethernet. Ethernet interworking needs to be provided by implementing LAN emulation, as defined by the ATM Forum.
• Support of multicasting and connection-less services.

• Simple management and modular architecture of the network elements to facilitate scalability.

• Cost reduction by low-cost interfaces, efficient multipoint implementation and internal synchronous operation. Also compliance to interface standards to facilitate integration of third-party products.

• Performance, availability and reliability requirements as described in section 5.4.7.
6 ATM networking functional description

This chapter presents an analyses of the function to be implemented by the network. It discusses three aspects, namely the functioning of the networking elements, the services that are offered to users and the interworking of the network with its users.

A network typically comprises a set of end-systems, a set of switches, and means to interconnect them (interfaces). This chapter starts with the description of a network layering model, which will be used for the modelling of the network and the design of the switching units in chapter 9. In addition, the protocol stacks that apply to the various interfaces are presented.

Subsequently, a summary of potential network service users and how they are supported by the network is given.

In conclusion, the interworking with the network is described. Humans make use of network services by means of applications running on end-systems. The interaction between user applications and the network need to be defined and translated into interactions between the elements of the network (e.g. end-system or switching unit). The definition of these interactions has not yet been achieved. However, a starting-point is given in Appendix A.

6.1 Network modelling

6.1.1 Definitions

The following set of definitions applies for terms used throughout this thesis. These definitions correspond with the definitions used in ISO 8648 which describes the internal organisation of the network layer (ISO91b), and with ECMA document TR/44 which describes an architectural framework for private networks (ECM89).

An End-System (ES) is equipment intended for running user applications, e.g. a PC or a workstation.

An Intermediate System (IS) is a system which does not contain application processes and is used only to enable interconnection of other systems through relaying mechanisms at or below the Network Service boundary. Examples of intermediate systems are: switches, bridges and routers (or collections of switches, bridges and routers).

A SubNetwork (SN) is a physical medium or a collection of both equipment and physical media, which form(s) an autonomous whole and which can be used to interconnect systems for the purpose of communications. If the subnetwork consists of a collection of both equipment and physical media, then it may be represented as an IS.

A Subnetwork access protocol is a protocol which has to be executed by a system that wishes to access that subnetwork, irrespective of conformance of that protocol to OSI standards.
A **Subnetwork service** is the service provided by those elements of the subnetwork access protocol on which the subnetwork actively operates in combination with the routing and relaying capabilities of the subnetwork.

### 6.1.2 Network model

This section describes a network model that permits any real network to exist as a subnetwork. The model corresponds with the Unconstrained Network Service Provider Model (UNSPM) as described by the European Computer Manufacturers Association (ECMA) in [ECM89].

Subnetworks can be interconnected by an Intermediate System (IS). The result can be seen as a larger composite subnetwork which in turn can be interconnected with other subnetworks. The objective of subnetwork interconnection is to increase the number of end-systems. If the service supported by the resulting subnetwork differs from that of the smaller subnetworks, a new subnetwork service boundary and a new protocol layer is generated. Each subnetwork has its own addressing space. End-systems that want to communicate with each other, need to share a common subnetwork service so that they can both be identified in the same addressing space. An example of a widely used subnetwork protocol is the Internet Protocol (IP) which allows end-systems all over the world to communicate with each other.

This model allows the presence of multiple hierarchical subnetwork layers. A subnetwork may consist of a physical medium only, in which case it is difficult to indicate the boundaries of a subnet. However, networks can be described in terms of subnetwork (access) service boundaries. This boundary separates access protocols from protocols over the top. In case of a physical medium only, it corresponds with the MAC service boundary. The boundary of a subnetwork service is located inside an ES or an IS. A subnetwork service boundary hides everything below that boundary to its users.

**Figure 6.1**  End-systems connected to a subnetwork

As an example, Figure 6.1 shows two end-systems, ES A and ES B, that share a common subnetwork service of layer K. The figure shows the subnetwork service boundary of this subnetwork and the protocol model. The (K)-layer subnetwork service provides a Service Access Point (SAP) to its users. How the subnetwork realises these services is not relevant to the users and therefore not shown in the figure.
Figure 6.2 Practical example of subnetworks

The subnetwork service of Figure 6.1 may be decomposed into (K-1)-layer subnetwork services and (K-1)-layer entities inside intermediate systems. An example of such a decomposition is shown in Figure 6.2. Apart from ES A and ES B, five other end-systems share the (K)-layer subnetwork service. If the (K-1)-layer subnetwork services of Figure 6.2 contain both equipment and physical media, they may be further decomposed into (K-2)-layer subnetwork services and entities inside intermediate systems. This process can be repeated until none of the subnet services contain such entities.

The protocol model of the decomposition example is shown in Figure 6.3. The figure shows that the shortest path between end-system A and end-system B in fact comprises three (K-1)-layer subnetwork services which are interconnected by (K-1)-layer entities in two intermediate systems. In the protocol model, the height of a subnetwork represents the intervention level of its service. In the figure, the various (K-1)-layer subnetwork services have different heights to indicate that the intervention level of these subnet services may (but do not need to) differ.

6.1.3 Addressing and routing

The objective of this section is to uniformly solve the addressing and routing problems on every subnetwork layer K, thereby making the network scalable. The problem is discussed on the basis of the example of Figure 6.2 and Figure 6.3. ES A wants to send a packet to ES B whose (K)-layer subnetwork address it knows. The addressing process iteratively binds (K)-layer addresses to (K-1)-layer addresses and selects a route to the destination. Since the destination (ES B) must be able to respond, the address of ES A of each layer are sent along.
Figure 6.3 Protocol model of example

The first step is to bind the (K)-layer subnetwork address of ES A to the (K-1)-layer subnetwork address of ES A. After this, a route is selected to ES B. How this route is selected (i.e. via a hierarchical addressing scheme, by an address resolution protocol or by using tables) or whether the route is selected for the duration of a connection or for a single packet is beyond the scope of this document. The result of the route selection process is the (K-1)-layer subnetwork address of IS I. Now, the packet is ready to leave ES A. Along with the packet, ES A sends the (K)-layer subnetwork addresses of ES A and ES B. The (K-1)-layer subnetwork addresses of ES A and IS I are presented to SN 1 as parameters. These parameters are needed to successfully transport the packet to IS I.

In IS I, the (K)-layer address of ES B is used to select a route. The result of this process is the (K-1)-layer subnetwork address of IS II and the output port of SN 2. After this, the packet is passed on to SN 2.

IS II also selects a route to ES B on the basis of its (K)-layer subnetwork address. This process yields the (K-1)-layer address of ES B and the output port of SN 3. The packet is then passed on to SN 3, which presents it to ES B.

This example only describes the procedures for two hierarchical layers, but the above-mentioned procedure can be repeated for multiple layers.

6.2 End-system interfaces

The network supports at least three interfaces to end-systems: ATM for true multimedia support, Ethernet for support of latency LANs, and ATM-on-top-of-Ethernet for support of low-cost ATM.

End-systems that are equipped with ATM hardware use the native ATM protocol stack and are offered all services that are provided by the ATM network. The protocol architecture is depicted in Figure 6.4. The service boundary of the ATM subnetwork is located between the AAL and the ATM layer. The ATM subnetwork provides true multimedia services.
Applications may request a guaranteed QoS on which they can rely when granted. Interworking with latency LANs is provided by the LAN emulation entity of which the LAN Emulation Client (LEC) is located in the end-system. The remaining LAN emulation entities (i.e. LES, LECS and BUS) are located inside the network.

End-systems that are connected to the network via Ethernet equipment and a shared medium make use of the Ethernet protocol stack. These end-systems are provided with a traditional connection-less best effort (emulated) LAN service (e.g. shared Ethernet). The protocol architecture is shown in Figure 6.5. The right end-system is connected to the ATM network via an ATM-LAN InterWorking Unit (IWU). Since the network needs to support Ethernet without a separate IWU, it must be accommodated inside the switch. All the LAN Emulation entities are located in the network rather than in the end-system. Figure 6.5 also shows how an ATM end-system (the end-system on the left side) communicates with an Ethernet end-systems through LAN emulation. As the figure shows, the MAC service boundary is located on top of the LE layer in the ATM end-system and on top of the MAC layer in the Ethernet end-system.

End-systems that are connected to the network via a point-to-point link and are equipped with Ethernet adapter hardware, make use of a protocol architecture similar to that of Figure 6.4. The only difference is that the physical layer is constituted of Ethernet hardware. This architecture enables end-systems to use ATM-on-top-of-Ethernet which allows them to make use of the features of ATM without ATM hardware. End-systems are offered the same services as true ATM systems but with a lower bandwidth. Since this architecture uses Ethernet adapters, it does also support switched Ethernet. If and end-system does not need QoS assurance, it may choose to use switched Ethernet instead of ATM-on-top-of-Ethernet for efficiency reasons. As soon as an assured QoS is needed, the used protocol stack changes back to ATM-on-top-of-Ethernet. The changeovers may be carried out via signalling. The concept of "ATM-on-top-of-Ethernet" is worked out in chapter 8.
6.3 Network users

Applications rely on different network interfaces. Since the ATM network needs to support all applications, each and every of these interfaces needs to be provided. Examples of network environments that applications use are: ATM, LAN and Internet (TCP/IP).

The ATM network environment provides a native ATM API that provides all services supported by the network. Future multimedia applications will use this API to communicate. This API is currently being defined by the ATM Forum \[ATM95b\].

Most existing applications are designed to interface to a LAN. ATM LAN Emulation (LE), which has been described in section 3.6, provides a LAN interface on top of the ATM network. LE supports both Ethernet and Token ring LANs. If LE is used to run LAN applications on an ATM end-system, the LAN emulation client must be implemented in the form of a NDIS/ODI software driver in the end-system. If LE is used to connect a LAN to the ATM network, the LEC resides in the interworking unit.

Many networking applications make use of the TCP/IP protocol stack. This stack can be supported in two manners. The first manner is by putting TCP/IP on top of the LEC entity. Currently, TCP/IP protocols are placed on top of LAN drivers. Since LE provides this driver, this manner needs no additional effort. The second approach omits the LE layer by directly putting TCP/IP on top of ATM \[LAU94\]. The ATM Forum is currently working on a specification to put multiple protocols, including TCP/IP, on top of ATM \[ATM95c\].
6.4 Summary and conclusions

The purpose of this chapter is to give a functional description of the network to be designed.

A network model has been described that uses multiple layers to indicate different levels of subnetworks. The addressing and routing has also been discussed. This model will be applied to the design of the ATM switches in chapter 9.

End-system can be attached to the network via three different interfaces, namely Ethernet, ATM and ATM-on-top-of-Ethernet. The protocol stacks of these three interfaces (with the identification of the boundaries of the subnetwork levels as defined by the network model) have been described.

There are basically three environments that applications rely on: ATM, LAN and Internet (TCP/IP). All three environments must be supported by the network. The ATM environment is supported by an ATM API, whereas the LAN environment is supported by LAN emulation. TCP/IP can be supported in two different ways: through LAN emulation, or directly. The direct support of TCP/IP is currently being developed.

The interaction between the users and the network has not yet been described, but a starting point is given in Appendix A.
7 ATM networking design aspects

The previous chapter has presented a networking architecture, based on OSI terminology, of an ATM-based multimedia network. Prior to the design of the three building blocks of the network (i.e. end-systems, switches and interfaces) in the subsequent chapters, several design decisions are made that influence the design of these blocks. This chapter describes these decisions. The decisions are based on the fact that the target market is identified as the customer premises networking market.

7.1 Network structure

This section defines certain restrictions to the network structure with the objective to simplify network operation. In principle, the networking equipment within a single building is concentrated in a single location (e.g. the basement). Switching elements are only distributed if this is inevitable because of cable length limitations. Switching units typically serve a group of users and are interconnected via high speed links. Since ATM supports various bit-rates, the bandwidth an end-user is provided with is scaled to his needs: the CAD/CAM division gets high-speed interfaces, while other divisions may get lower speed interfaces. This way, every user is provided with sufficient bandwidth without superfluous hardware cost.

Reliability of the network is improved by providing redundant links. Switches are preferably interconnected via multiple links. The connection setup entity selects one of these links when a virtual connection is set up. If the link fails, the connection is re-established via another link. The link selection mechanism may be implemented in several ways. The most straightforward way is to maintain a list of available links and select the first link on this list. Another, more complicated, way is to select a link on the basis of availability of a link and traffic density on a link. This allows traffic to be spread more evenly over several links and therefore reduces congestion risks.

The Virtual Path concept is used to drastically decrease the amount of signalling between switches. If possible, cross-connects (permanent Virtual Path Connections) are set up between switches to prevent intermediate switches from having to participate in connection control procedures. VPCs are also set up between switches and network servers. An example of this principle is shown in Figure 7.1. The figure shows four end-systems connected to three different switches. These switches are interconnected via VPCs so that every switch that contains links to end-systems is connected to every other switch that contains links to end-systems. Now, if end-system A wants to establish a connection to end-system B, only IS A and IS B are involved in connection setup. Likewise, if end-system D wants to set up a connection with the Server, only IS C is involved in connection setup.

Switches are interconnected by subnets which may be classified by two types: proprietary and foreign. The difference between these two types is that a foreign subnet is controlled by a public network provider while a proprietary subnet is owned by the owner of the switches. On proprietary subnets, the number of open connections is not of great importance, but on foreign subnets, connections are charged. ATM charging is expected to be based on a basic rate reflecting the cost for providing the customer access and traffic dependent charges.
comprising charging for call establishment and charging according to the duration and bit-rate of a call. If the latter component is relatively expensive, cross-connects may turn out to be highly cost-ineffective. For this reason, special attention is paid to the use of VPCs and VCCs between switches.

If a proprietary subnet allows the use of cross-connects, a VPC is set up between two switches when needed. After this VPC has been set up, intermediate systems are no longer involved in connection control between those two switches. Moreover, the size of the header translation table of an intermediate switch remains small because the VCCs are all part of the same VPC and thus covered by the same entry in the translation table. Cross-connects between switches that are interconnected via foreign subnets are only used if needed or if this turns out to be the most cost-effective method.

The usability of permanent VPCs highly depends on the complexity of the network. Managing permanent virtual connections in a complex network can be extremely unpleasant. Therefore, the permanent VPC concept is only feasible for networks with a relatively small number of switches. For larger networks, the VPCs need to be set up via signalling.

7.2 Number of connections

An important parameter for the dimensioning of an ATM switch is the number of connections that it needs to switch simultaneously and thus the maximum number of entries in its translation table. The method a switch uses to look up addresses in the translation table is essential for the switching speed. The looking up is considerably simplified (and therefore accelerated) if the table entries reside on fixed locations. This implies that the size of the translation table for every inlet is fixed. Since this size limits the maximum number of
connections an attached end-system can set up simultaneously, its determination must be well thought-out.

End-systems are divided into two categories: normal end-systems and servers. Normal end-systems typically represent a single user, who only needs a limited number of connections simultaneously, say 32. Furthermore, normal end-systems do not need cross-connects so the use of multiple VPCs is unnecessary. Therefore, normal end-systems are provided with a limited number of VCCs (32) within one single VPC.

Servers need to maintain connections to a large number of end-systems. The previous section has demonstrated how VPCs may be set up from the server to switches, thereby allowing end-systems to set up connections without bothering intermediate switches. For this reason, a server needs to be able to set up multiple VPCs and a large number of VCCs. The maximum number of VPCs a server is offered determines the maximum number of end-systems in the network. If, for example, the maximum number of VPCs is equal to 32, a switch can set up VPCs to 32 other switches or servers. The maximum number of VCCs a switch needs to support within a single VPC is equal to the number of end-systems that may be attached to a switch. In section 9.5.3, it will be demonstrated that an "ATM-on-top-of-Ethernet" switch supports a maximum of 160 end-systems. Therefore, it is assumed that 160 VCCs in a single VPC will suffice.

Under the above described assumptions, the translation tables of switching units may be divided into two parts. The upper part represents the normal end-systems and the lower part represents the servers. Every entry in the table has a fixed size so that the location of each end-system in the translation table can be easily determined.

7.3 Queuing

There are two fundamental classes of traffic: guaranteed traffic and best-effort traffic [NEW94]. Guaranteed traffic is traffic for which an explicit guarantee of service in terms of QoS parameters as described in section 5.1 has been agreed upon by the network and the application. Applications not capable of predicting their bandwidth requirements use a best-effort service, also known as Available Bit-Rate (ABR) service.

The ATM network needs to ensure that the QoS of the guaranteed traffic will never be affected by best-effort traffic. Therefore, the cell queuing in ATM switches must be separated into at least two traffic classes implemented in separate FIFO queues. The two queues approach places guaranteed traffic in one queue and best-effort traffic in the other. The guaranteed traffic is always served prior to the best-effort traffic. More complex queuing structures may be used to offer greater efficiency or isolation within different sub-classes of traffic.

There are at least two mechanisms to share the available bandwidth between the contending users: a delay mechanism and a loss mechanism. The delay mechanism puts cells in a queue until the bandwidth is available. The loss mechanism discards cells when the queue is full. If the mechanism simply discards arriving cells, each discarded cell is likely to belong to a different higher-layer protocol packet. An improvement is to implement a loss mechanism in
the best-effort queue of the ATM switch that discards the remainder of a packet if it discards on cell from the packet.

To limit the required number of queues and the complexity of the algorithm, the two queues approach is used. Since one queue is always served first, it causes little delay. This queue is thus typically meant for low-delay traffic and may therefore be relatively small. The other queue is always served last so it may cause considerable delay. However, if the queue is very large, the loss probability is small. therefore, this queue is typically meant for low-loss traffic. Since LAN traffic is typically low-loss traffic, it is stored in the low-loss queue.

7.4 LAN emulation implementation

One of the services to be provided is LAN Emulation (LE). The ATM network provides multiple Virtual LANs (VLANs) to which end-systems can subscribe. The LE specification defines that across the User-Network Interface (UNI), a VLAN comprises a broadcast channel and several point-to-point channels. If a LEC wants to send a frame to another LEC, it immediately sets up a point-to-point connection. However, this requires a considerable amount of signalling and switched VCCs are expensive in terms of memory. Therefore, the implementation of LAN emulation will differ from the proposals of the ATM Forum. Although the network supports the UNI by allowing LECs to execute the procedures for setting up point-to-point connections (data direct VCCs) to other members of the same LAN, these connections are not really set up by the network. Instead, only a single broadcast channel (a multipoint-to-multipoint connection) exists for every VLAN, thereby creating a virtual shared medium. This broadcast channel is used for all traffic between the LECs. A LEC that is aware of this implementation does not bother to set up a data-direct connection, but simply forwards all cells to the broadcast channel. This significantly simplifies the implementation of the LEC.

To prevent unicast data from being sent to every member of the VLAN, a selective bridging function is implemented in every switch. This bridging function maintains a list of known LECs and the ports behind which they reside. Incoming traffic is passed on solely to the LEC(s) it is destined for. Because a large number of LECs may be connected to a single port, the list only contains LECs that have recently been sending or receiving traffic. An aging mechanism removes the other LECs. If a LEC does not appear in the list, it is located through Address Resolution. In the meantime, the traffic destined for this LEC is forwarded to all ports.

The implementation of LE by means of a single broadcast VC has a number of advantages:

- Less connections are needed, because only a single connection is used for every VLAN. Since switched virtual connections are expensive in terms of memory and setup time, this is a major advantage.

- The (ATM) switching units can easily be used for LAN interconnection by setting up a permanent virtual multipoint-to-multipoint channel (via management) that interconnects all these LANs. This way, no signalling is needed.
• The multicast function is distributed over the switches.
• The characteristics of the hardware are optimally utilized.

There are also a couple of disadvantages:
• A proprietary implementation complicates interworking with the outside ATM world. If, for example, a VLAN needs to be defined that includes end-systems of foreign corporations, an interworking function needs to be implemented that multiplexes traffic of all data-direct VCCs between LECs into a single broadcast connection and vice versa.
• Multipoint-to-multipoint connections impose additional requirements to the architecture of switches. Traffic originating from different sources must not get intermingled. In a multipoint-to-multipoint connection, the switch must discriminate between cells from different sources and therefore the cells must be accompanied by additional information.

7.5 Network timing support for isochronous services

ATM networks must be able to support isochronous applications like PCM voice. Therefore, the clock of the sender must be provided to the receiver in order to avoid slips. This implies requirements on the network in terms of support of synchronisation of the access lines and tolerable slips in the case of a synchronisation failure.

One way of providing the necessary clock information is to make use of the existing clock distribution network for the 64 kbits/s ISDN. If the corporate ATM network itself is capable of distributing this clock to all end-system, transport of isochronous traffic is significantly simplified.

A manner to distribute the clock is by sending synchronisation signals over the interfaces between switches and from switches to end-systems. This method of approach is used in the "ATM-on-top-of-Ethernet" concept which is described in chapter 8.

7.6 Summary and conclusions

This chapter has described several design decisions regarding the multimedia network, with the objective to simplify the functioning of the network, increase its performance and lower the cost of implementation.

In the first place, permanent VPCs are set up between switches to facilitate connection setup procedures. However, if in a specific situation permanent VPCs turn out to be cost-ineffective, for example when a foreign net is used, switched VPCs are used.
To simplify the header translation in the switches, the number of VPCs and VCCs an end-system may set up are limited. A normal end-system is provided a single VPC with a maximum of 32 VCCs. A server is provided 32 VPCs, each with a maximum of 160 VCCs.

The switches make use of two queues: one for low-loss traffic and one for low-delay traffic. Since low-delay traffic is always served first, the low-delay queue may be small. The low-loss queue, on the other hand must be fairly large.

The implementation of LAN emulation differs from the ATM Forum’s proposals. The communication between LECs does not take place over data-direct VCCs, but over a single virtual shared medium implemented by a multipoint-to-multipoint connection. This considerably decreases the amount of signalling.

For the support of synchronous applications, the network must provided the clock of the sender to the receiver. The network distributes a single clock to all end-systems by sending synchronisation signals to them. This way, the transport of isochronous traffic is considerably simplified.
8 ATM-on-top-of-Ethernet

This chapter describes a concept, named ATM-on-top-of-Ethernet, which lowers the cost of ATM by implementing it on top of existing Ethernet equipment. Since the installed base of Ethernet equipment is huge and the price low, it is extremely suitable for low-cost ATM. Furthermore, it forms an attractive means to migrate from Ethernet to ATM. Issues that need to be discussed are how to pack ATM cells in Ethernet frames and how to provide QoS guarantee.

Since the maximum bit-rate of Ethernet is 10 Mbits/s, the ATM-on-top-of-Ethernet approach yields a relatively low maximum bit-rate. The maximum bit-rate can be increased to 100 Mbits/s by implementing ATM on top of Fast Ethernet. However, since the installed base of Fast Ethernet is small, this chapter focuses on ATM on top of 10 Mbits/s Ethernet.

8.1 Guaranteeing quality of service

The main obstacle for the implementation of ATM-on-top-of-Ethernet is that Ethernet has no means to guarantee Quality of Service (QoS). Concerning the influence of Ethernet on the QoS, the following observations can be made:

- Ethernet uses frames with variable lengths, which causes traffic to experience variable packing delays.

- Because Ethernet uses the CSMA/CD protocol, the delay caused by collisions may be arbitrary long.

- Ethernet does not have a priority scheme.

End-to-end QoS guarantee can only be provided if the ATM-on-top-of-Ethernet implementation meets the following requirements:

- It must be used in a switched environment so that every end-system is provided with 10 Mbits/s and the probability of collisions is minimised.

- The switch and the end-system must use fixed-length frames so that delay is small and constant.

- The switch must send synchronisation frames at fixed intervals so that the end-system can synchronise its clock to these frames.

- The time between two consecutive frames must be an integer multiple of 125 μs to facilitate the transport of PCM-coded voice.
8.2 Implementation aspects

8.2.1 The ATM-on-top-of-Ethernet model

The entity that performs QoS guarantee can be implemented by enhancing the upper part of the Ethernet MAC layer (that is not implemented in the Ethernet hardware, but in the software drivers) by adding a synchronisation and bandwidth distribution mechanism. Since this mechanism is implemented in software, PCs equipped with an Ethernet adapter only need a new set of software drivers to upgrade to a full blown ATM end-system. The only drawback is the limited amount of available bandwidth and the fact that ATM-on-top-of-Ethernet requires a switched networking environment. The protocol may also provide a means to dynamically switch between legacy Ethernet and ATM-on-top-of-Ethernet. If an end-system does not need QoS guarantee, it may choose to switch to legacy Ethernet. This way, the extra burden of the ATM protocol stack in the end-system is avoided. The protocol stack of the ATM-on-top-of-Ethernet model is depicted in Figure 8.1.

![Figure 8.1 ATM-on-top-of-Ethernet concept](image)

8.2.2 Clock synchronisation

The end-system must be provided with clock synchronisation information. This can be done by letting the switches send synchronisation frames, which are used by the end-system to synchronise its clock. These frames then have to be sent at fixed intervals and must not be subject to collisions. Therefore, a bandwidth distribution scheme needs to be provided that allows the switch and the end-system to send frames without collisions.

In theory, an end-system can synchronise its clock to the reception of the beginning or the ending of the frame. However, Ethernet hardware has no means to indicate the reception of the beginning of a frame to its driver. Therefore, synchronisation can only take place on the reception of the entire Ethernet frame. This has the additional advantage that it assures the
end-system of being allowed to start transmitting immediately after the reception of the synchronisation interrupt. The interrupt handling routine can then initiate sending before processing the received data. Synchronisation on the ending of a frame does require that the length of a frame is constant so that the packing delay is also constant.

8.2.3 Bandwidth usage

Since Ethernet is half-duplex, the available bandwidth needs to be shared by the switch and the end-system. This can be done in two ways. One possibility is to equally distribute the available bandwidth over the end-systems and the switch, and allowing both to alternately send a single Ethernet frame. This way, both have 5 Mbits/s at their disposal (theoretically). Equally distributing the available bandwidth has one important disadvantage. Many applications, like video distribution and file transfer applications, transfer large amounts of data in one direction while almost nothing is send in the opposite direction. In these situations, the reserved bandwidth in one direction is hardly being used while the reserved bandwidth in the other direction is fully loaded. This situation may be improved by dynamically allocating more bandwidth to the system that sends the data and less to the system that receives it. This dynamic bandwidth allocation mechanism must also ensure that the switch can send its synchronisation frames without collisions.

8.2.4 Packing methods

Several methods can be used to pack ATM cells in Ethernet frames. The first and most straightforward method packs as many complete ATM cells in the data field of the Ethernet frame as possible. As an alternative packing method, the ATM cell headers may be replaced by smaller proprietary cell headers. It is anticipated that the number of connections an end-system needs will be relatively small (see also section 7.2). Under the assumption that an ATM end-system does not need to use more than 32 connections simultaneously, the VPI and VCI fields in the ATM cell header may be replaced by a single virtual connection identifier of 5 bits. Moreover, since the Ethernet frame is protected by a CRC field, the HEC field of the ATM cell is superfluous. If the CLP field is also omitted by assigning all cells the same priority, the ATM cell header may be replaced by a 1-octet proprietary header which consists of a 5-bit connection identifier and a 3-bit payload type identifier. This way, the 53-octet ATM cells are replaced by 49-octet cells so that more cells can be stored in a single Ethernet frame.

The efficiency of the two above described packing methods may be increased by using a part of the Ethernet frame header to store payload. Since the Ethernet hardware is only used as a physical medium in a point-to-point link, the Source Address and Destination Address fields in the Ethernet header do not provide essential information. Ethernet hardware may be programmed to process every incoming Ethernet frame (so without looking at the Destination Address field) and the Source Address field is supplied by the driver. Therefore, the 12 octets occupied by these two fields may be used to carry payload. This does, however, complicate the processing of the Ethernet frame since part of the payload is separated from the rest.

Table 8.1 shows an overview of the four proposed packing methods and their characteristics. To evaluate these methods, it is helpful to calculate the maximum possible bit-rate on the ATM payload level for each method. Appendix B presents these calculations under the
Table 8.1 Characteristics of packing methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Cell size</th>
<th>Cell header size</th>
<th>Ethernet header+trailer size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method 1</td>
<td>53</td>
<td>5</td>
<td>26</td>
</tr>
<tr>
<td>Method 2</td>
<td>49</td>
<td>1</td>
<td>26</td>
</tr>
<tr>
<td>Method 3</td>
<td>53</td>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>Method 4</td>
<td>49</td>
<td>1</td>
<td>14</td>
</tr>
</tbody>
</table>

assumption that the bandwidth is statically distributed over the end-system and the switch. This Appendix shows that method 4 is the most efficient, but does not always lead to the highest bit-rate on the ATM cell payload level, since this also depends on the maximal allowable packing delay, the time an end-system needs to send its frame and the complexity of a packing method. Therefore, determination of the optimal packing method cannot be done until additional information is obtained. Moreover, since the results in Appendix B only apply to a situation in which the bandwidth is statically distributed, additional calculations may be required when a dynamic bandwidth allocation mechanism is used. ? summarizes the advantages and disadvantages of each of the four packing methods.

So far, the best results are obtained by using method 4 to pack 9 cells in a frame. This leads to a maximum one-way bit-rate at the ATM cell payload level of 4.61 Mbits/s and a padding delay of 750 µs.

8.3 Conclusions

This chapter has described a low-cost ATM interface using Ethernet hardware, named ATM-on-top-of-Ethernet. Two bandwidth distribution scheme's have been proposed. The first statically divides the available bandwidth into two equal parts; the second uses a dynamic bandwidth allocation scheme to optimally utilise the available bandwidth. In both situations, the switch sends frames at fixed times so that the end-system can use these frames to synchronise.

Four methods to pack cells in Ethernet frames have been discussed. The most efficient method replaces ATM cell header with a 1-octet proprietary header and uses the source and destination address fields of the Ethernet frame header to store data. Whether this method is preferred over the other three methods depends on parameters such as the maximal allowable packing delay, the time an end-system needs to send a frame and the complexity of the implementation of each method. Since this information is not yet available, the selection of the most suitable packing method is for further study.
Table 8.2 Comparison of packing methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lowest complexity</td>
<td>Lowest bit-rate</td>
</tr>
<tr>
<td>2</td>
<td>More efficient than 1,3</td>
<td>Limitations caused by proprietary cell headers</td>
</tr>
<tr>
<td></td>
<td>Low complexity</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>More efficient than 1</td>
<td>High complexity caused by payload separation</td>
</tr>
<tr>
<td>4</td>
<td>Highest bit-rate</td>
<td>Highest complexity caused by payload separation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Limitations caused by proprietary cell headers</td>
</tr>
</tbody>
</table>
9 Switch design aspects

Networks are formed by end-systems, switches and interfaces. This chapter describes the functional design of a switching unit (switch). First, a summary is given of the functions a switch has to perform. Subsequently, the modular structure of the switch is described on the basis of the network model of section 6.1. Finally, the components of the switch are described and designed.

9.1 Functions of a switch

In the previous chapters, several functions that an ATM switch needs to perform have been discussed. These functions are listed below. For each of these functions, the decision has to be made if it is implemented in software or in hardware. The functions are:

- Translation of ATM cell headers. This is one of the three primary functions of the switch (see section 4.1). To minimise the delay caused by the header translation process, it must take place "on the fly" and therefore this function must be implemented in hardware.

- Routing traffic from a certain inlet to a certain outlet. Just like header translation, this function is carried out on the fly to minimise the delay that traffic experiences. Therefore, this function is implemented in hardware.

- Queuing data that cannot be forwarded immediately. To distinguish between low delay traffic and low loss traffic, separate queues are needed for both types (see section 7.3). This is also one of the primary functions of the switch and is implemented in hardware.

- Connection control and management. These functions are typically implemented in software because of their complexity and since they are less time critical. Moreover, software implementation facilitates the maintainability and adaptation to changing standards of these functions.

- Address learning and address resolution. These function are implemented in software.

- Implementation of the LAN emulation entities, including the LEC. The LEC has to set up and maintain ATM connections for ethernet frames that are translated to ATM cells. LAN emulation is also implemented in software.

- Selective bridging function (see section 7.4). Because VLANs are implemented by a single multipoint-to-multipoint virtual connection, a selective bridging function must be provided, that prevents frames from being forwarded to LECs they are not destined for. The filtering part must be carried out on the fly and thus in hardware. The function uses an address table to determine whether or not to forward the data. This table is updated by software.
• Interworking. The switch must be able to interwork with Ethernet. This requires an implementation of AAL5. Furthermore, the switch must convert the proprietary cell headers of 'ATM-on-top-of-Ethernet' to ATM cell headers. Both functions must be implemented in hardware.

9.2 Structure of the switch

Currently, Philips use a TDM-bus mechanism for its SOPHO (which stands for Synergetic Open PHilips Office automation) telephony system. An important advantage of the TDM-bus principle is that it is very suitable for scalable switches. Furthermore, using the TDM-bus in switches reduces the hardware complexity and cost of the switch, and simplifies the implementation of multipoint traffic. Since Philips focuses on the customer premises market, these are very important requirements. An ATM switch using a TDM-bus would utilize these advantages and, moreover, facilitate the integration with the current SOPHO telephony system. If the reflection problems caused by the high bit-rate on the bus can be solved, the TDM-bus would be the ideal transfer medium.

The structure of a switch that uses a TDM-bus is shown in Figure 9.1. A switch is composed of a several boards, which are accommodated in a shelf and interconnected by a high-speed TDM-bus: the backplane. Each board contains several ports, to which the lines are connected, and is divided into a Transmit section that sends data onto the lines and a Receive section that receives data from the lines. The support of different physical interfaces and interworking units may be implemented on different boards. When the number of end-systems exceeds the total number of ports on all boards in a single shelf, boards of separate shelves can be interconnected, thereby creating a larger (distributed) switch. If the shelves are located in the same room, they may be accommodated in a single cabinet. If data is sent from board 1 over the backplane to board 2, board 1 is called the source board, whereas board 2 is referred to as the destination board.

![Switch block diagram](image)

Figure 9.1 Switch block diagram

Regarding the routing method, section 4.2.3 describes two fundamental choices. The first fundamental choice is whether a routing tag is used to automatically route the cell to the destination port. A routing tag facilitates the routing process but complicates multicast functions. To utilize the advantage of a routing tag without complicating the implementation
of multicasting, it will not be used to individually indicate one outlet on a destination board, but to indicate a group identifier, which is forwarded to multipoint distribution entities on the destination boards (why this is required will be explained in section 9.4). These entities then use a table to translate this into a set of individual outlets belonging to that group. The second choice is whether the route to the destination is determined for the duration of a connection or each time a cell arrives. The switch of Figure 9.1 has only one route to every destination so the route selection procedure always yields the same result. Since in this situation selecting the route each time a cell arrives does not lead to a more optimal use of resources but only increases the complexity, it yields no profit. Therefore, the route is determined and stored in a table for the duration of a connection.

To avoid unnecessary congestions and deadlocks, the queuing of cells must take place on the destination board (output queuing). A single block of memory is used for the implementation of the queues of all outlets (central queuing). The decision whether the memory space is partitioned statically or dynamically depends on how much memory is needed for each method and how complex the control will be. This decision can be made later on.

![Figure 9.2](image)

**Figure 9.2** Board layout

Figure 9.2 shows the layout of a board. The three main tasks of a switch (header translation, routing and queuing) are performed as follows. Cells entering the switch arrive on the source board at the header translation entity. This entity interprets the VPI/VCI values of the cell and translates them to the values that apply when the cell leaves the switch from the destination board. Furthermore, a routing tag is attached to the cell that contains sufficient information to route the cell via the queues to the multipoint distribution entity on the destination board and convey the group identifier to this entity. Two routing decisions need to be made. The first selects the destination board(s) and the second selects the destination line(s).

Applying the model of section 6.1 to this routing mechanism results in the situation shown in Figure 9.3. The figure shows an ATM network (layer (K+1)) that consists of ATM end-systems and ATM switches (shelves). This ATM network uses VPI/VCI fields to identify
connections. The links between the switches, and between the end-systems and the switches, are formed by (K)-layer subnetworks, which in most cases will be point-to-point links, but may also be implemented by a foreign network (e.g. ISDN). Looking inside a shelf reveals that it is in fact composed of boards that are interconnected by a (K)-layer subnet, namely the backplane.

Let us now consider how the routing function described above fits in the model. The following observations can be made:

- The routing from a board to another board is the responsibility of the (K)-layer backplane network.

- The generation and selection of all addresses (including the queue identifier) is the responsibility of the (K+1)-layer. This includes:
  - manipulation of VCI/VPI values,
  - generation and manipulation of destination group identifiers and queue identifiers,
  - generation of board identifiers (the corresponding routing is delegated to the backplane subnet).

In the example of Figure 9.3, the function of board A is to route cells to board B. The function of board B is to route every cell to the correct outlet and give it the correct VCI/VPI values, which can then be interpreted by board C to further route the cell. However, the function of board B is significantly simplified by the fact that board A supplies board B with the correct VPI/VCI values and a routing tag that contains the sufficient information to let board B select the correct outlet. The procedure for board C and D is analogous to that for board A and B.
The requirement that a switch must support interconnection with Ethernet LANs, necessitates the switch to contain an embedded interworking unit. This situation is depicted in Figure 9.4. In this figure, the end-system on the left is a LAN end-system, whereas the end-system on the right is an ATM end-system. The (K+1)-layer subnetwork is the ATM network, dominated by ATM connection identifiers (VPIs/VCIs), whereas the (K+2)-layer subnetwork is a (virtual) LAN, dominated by MAC addresses. The (K+1)-layer subnetwork service boundary is now located inside the left switch and the left end-system is connected to the shelf via a (K+1)-layer subnet. The left shelf now also contains an interworking function that maps (K+2)-layer LAN addresses to (K+1)-layer ATM connections and vice versa. The two end-systems can only communicate if they are both part of a (K+2)-layer subnetwork. This is the case if the ATM end-system on the right contains a LAN emulation entity.

9.3 Multipoint-to-multipoint connections

As described in section 7.4, the proposed implementation of virtual LANs is by means of a single "broadcast" multipoint-to-multipoint VCC. The implementation of such a multipoint-to-multipoint connection imposes additional requirements to the switch. Inside a multipoint-to-multipoint connection, cells originating from different sources simultaneously arriving at the same destination board get intermingled because the address of the source of each cell is not revealed to the destination. Since these cells are all part of the same connection, it is impossible to distinguish them from one another. To assure that the AAL-SDUs (which is a LAN frame in the case of LAN emulation) can be regenerated, all cells belonging to the same AAL-SDU must remain together. This is guaranteed by introducing the following two restrictions when switching cells of multipoint-to-multipoint connections:
(1) A board collects all cells that are part of the same AAL-SDU (which is a LAN frame in the case of a VLAN connection). When all these cells are collected (regrouped), the cells are passed on to the destination board. This restriction prevents AAL-SDU segments from different ports on the same board to get intermingled.

(2) Cells coming from the backplane and transmitted by different boards, are stored in separate queues on the destination board. This prevents AAL-SDU segments from different source boards to get intermingled at the destination board. An alternative solution would be to assign each board sufficient time on the backplane for the transmission of an entire AAL-SDU. However, the transmission of a packet of 64 Kbytes (the maximum length of an AAL-SDU) would introduce an unacceptably long delay. A bit-rate of 1 Gbits/s on the backplane shared by 10 boards, for example, would lead to a delay caused by the backplane of more than 5 ms.

The regrouping of ATM cells belonging to the same AAL-SDU before they are passed on to the destination board causes an additional delay which may be unacceptable for certain services. Multipoint-to-multipoint connections are primarily used for the implementation of VLANs and typically have a low loss QoS requirement.

9.4 Routing

The routing of cells inside a switch comprises three decisions: selection of the destination board(s), selection of the destination port(s) and selection of the queue (low delay or low loss). A routing tag ideally provides sufficient information to automatically route an ATM cell to the destination port or ports it is destined for. This routing tag is determined at connection establishment. Cells that are part of a multipoint connection may need to be distributed to any set of destination ports. However, it is impossible to let the routing tag select each and every of these ports individually because this would require a huge routing tag. If, for example, a shelf contains 10 boards each of which supports 16 ports, the routing tag would need to be 160 bits long if every port is identified by a single bit. Therefore, the routing of cells to their destination port is split up into two stages.

The first stage is the selection of the destination board(s) and a logical group on these boards. Since it must be possible to select multiple boards, all boards are individually identified (a single bit is used for every board to select whether or not it belongs to the set of destinations). The logical group identifies the entry in a translation table that is used to distribute the cell to the destination ports. If the cell belongs to a point-to-point connection, the destination is a single port and the destination group identifier identifies this port. If the cell belongs to a multipoint connection, the cell must be distributed to a subset of the destination ports. In this case, the destination group identifier corresponds to a table entry that contains the corresponding destination ports (every port is selected by a single bit). The second stage is the copying of the ATM cell to the output ports. This copying is performed by the multipoint distribution entity on the basis of the destination group identifier and a table.

Throughout the whole switch, the two types of traffic (low delay and low loss) are separated (two queuing method, as described in section 7.3). The routing tag indicates in which queue
each cell needs to be stored. The routing tag structure is depicted in Figure 9.5. It comprises
a destination board(s) identifier, a destination group identifier, and a (1-bit) queue identifier
(QID).

| DEST. BOARD | DEST. GROUP | QID |

Figure 9.5  Routing tag structure

The relation to the model is now that the destination board identifier corresponds to the
address related to the backplane subnetwork. This field is used to find board B. The
destination group identifier and the queue identifier correspond to addresses that are used to
simplify the routing on board B (i.e. task of B is to find the outlets to C).

The following observations are made to match the model to the example:

- The destination board identifier is a (K)-layer identifier, but it is generated by a
  (K+1)-layer entity.
- The destination group identifier and the queue identifier are (K+1)-layer identifiers.

9.5 Backplane design

Boards are interconnected by a backplane: a high speed serial TDM-bus which is used by the
boards for mutual data exchange. Since the characteristics of the switch highly depend on the
possibilities and limitations of the TDM-bus, the bus has to meet certain requirements. These
requirements are:

- High bit-rate.
- Low hardware cost (components, backplane).
- Little overhead.

A possible solution for the TDM-bus backplane is Motorola's "Autobahn" [MOT93]. The
Autobahn consists of multiple Serial-to-parallel, parallel-to-serial transceivers (Spaceivers)
which can be used to implement a high-speed, half duplex, bi-directional serial data link with
a maximum effective data transfer rate of 200 Mbytes/s. Future versions will support data
transfer rates of 400 Mbytes/s. The serial link can be used to establish point-to-point or
multipoint connections.

Data is transmitted in bursts without preamble bits, which allows instantaneous data
acquisition without PLL clock recovery. Every byte of data is preceded by one
synchronization bit, so the data transfer is nearly overhead free. The Autobahn Spaceivers
contain components to eliminate the reflections caused by the high bit-rates.
Due to its limited functionality, the Autobahn seems to be a very suitable solution for the implementation of the TDM-bus. However, since it provides no framing, additional hardware is required.

9.5.1 Frame format

The most straightforward implementation of the TDM-bus is to statically divide the available bandwidth into equal parts, called time slots, and assign a slot to each attached board. One iteration for all time slots is referred to as a frame (Figure 9.6).

A board uses its assigned time slot to send data to other boards. Each slot is filled with Autobahn cells (AB-cells) each of which comprises an ATM cell and a routing tag. The routing tag is used to route the cell to its destination. Figure 9.7 shows the contents of an Autobahn slot. The gap between the slots is needed by the Autobahn Spaceivers to switch from transmitting mode to receiving mode and vice versa.

Since the Autobahn device operates with 32-bit words, processing is facilitated by guaranteeing that the size of an Autobahn cell is also an integer multiple of 32 bits. This implies that the size of the routing tag is also an integer multiple of 32 bits.

9.5.2 Synchronisation

The Autobahn does not provide any means to mutually synchronise the boards. Therefore, the synchronization has to be performed by external logic. How this can be realised has not yet been investigated. So this topic is for further study.
9.5.3 Dimensioning

The bit-rate on the backplane is 1.6 Gbits/s (200 Mbyte/s). This yields a bit-time of 0.625 ns or an octet-time of 5 ns. According to the Autobahn datasheets [MOT93], the gap between two slots needs to be at least 100 ns long, which corresponds to 20 octets (160 bits). In the future, the bit-rate can be doubled when Motorola releases the 400 Mbyte/s version of the Autobahn.

If \( S_s \) is the size of a slot in octets, the effective bit-rate is equal to

\[
\begin{align*}
    r_s &= 1.6 \cdot 10^9 \cdot 8 \cdot \frac{S_s}{S_s + 20}. \\
    \text{(9.1)}
\end{align*}
\]

The delay introduced by the backplane is equal to the frame-time \( T_f \). If \( N \) is the number of boards, then \( T_f \) is equal to

\[
\begin{align*}
    T_f &= 1.6 \cdot 10^9 \cdot 8N \cdot (S_s + 20). \\
    \text{(9.2)}
\end{align*}
\]

For the dimensioning of \( T_f \) and the routing tag, the maximum number of boards a shelf may contain needs to be determined. The Autobahn is capable of transporting 1600 Mbits/s. Frequently used ATM interfaces use bit-rates of 155.52 Mbits/s, 51.84 Mbits/s and 10 Mbits/s. A maximum of 10 boards in a shelf is a favourable number. A board then supports a maximum accumulated bit-rate of 160 Mbits/s without having to queue data before it is passed onto the Autobahn. This corresponds to one 155.52 Mbit/s interface, three 51.84 Mbit/s interfaces or sixteen 10 Mbits interface on a single board.

This yields the following dimensions for the routing tag:

- Destination board field: 10 bits (one bit per board).
- Source board field: 4 bits (0-15, but only the values 0-9 are used).
- Queue identifier field: 1 bit (low delay or low loss).
- Destination group field: 17 bits (this makes the length of the routing tag 32 bits and allows for 131,072 destination groups, which is more than sufficient).

The determination of the optimal values of \( T_f \) and \( S_s \) depend on how synchronisation is implemented and can therefore not yet be determined.

9.6 Board design

ATM is designed to support many different bit-rates and physical interfaces. For each of these interfaces, a board may be designed. It is also possible to accommodate support of multiple physical interfaces on a single board or add embedded interworking units to a board for the interworking with protocols other than ATM (e.g. Ethernet). This report focuses on the design
of a board that supports ATM-on-top-of-Ethernet because of the low-cost migration potential of this principle. However, most of what will be described also applies to the design of ATM boards using other physical interfaces.

The switch must also provide interworking with existing LANs, like Ethernet. For this purpose, the interworking unit needs to contain Ethernet physical interfaces. Since an ATM-on-top-of-Ethernet board also uses Ethernet physical interfaces, it uses the same physical layer hardware as an Ethernet interworking unit. Therefore, these two functions can be combined onto a single board, making use of the same hardware. For every port, it must be possible to configure whether it is currently used for Ethernet LAN interworking or for ATM-on-top-of-Ethernet. This configuration is performed by the network manager, or dynamically via signalling.

Figure 9.8 shows the logical structure of the two sections a board comprises (only the transport functions are shown). The receive section transfers data from the lines to the Autobahn. First, the data is processed. The processing starts with the conversion of an Ethernet frame to ATM cells (if that particular port is configured for Ethernet interworking) or unpacking ATM cells from an Ethernet frame (if the port is configured for ATM-on-top-of-Ethernet). Subsequently, ATM cell headers are translated into the headers that apply when the cells leave the switch, and a routing tag is attached. The cells are then divided into two groups: low delay (LD) cells and low loss (LL) cells. LL cells belonging to a multipoint-to-multipoint connection that are part of the same AAL-SDU are regrouped so that they can all
be put in the FIFO at the same moment. At the same time, a selective bridging function interprets the encapsulated destination MAC address and adjusts the routing tag. After the processing, the cells of all the lines are multiplexed into two FIFOs: one for LD cells and one for LL cells. These FIFOs may be very small because congestion does not take place. Finally, the board fills its time slot on the TDM-bus with the LD cells and as many LL cells as it can contain.

The transmit section transfers data from the Autobahn to the lines. First, cells originating from different boards are separated and cells belonging to a multipoint-to-multipoint connection that are part of the same AAL-SDU are regrouped. This is only possible if a source board identifier is provided with the cell. Since every board has its own time slot on the backplane, the identification of the current time slot reveals the identity of the source board. Subsequently, the cells are passed on to the multipoint distribution entity that puts the cells in the queue(s) of the destination port(s). This entity also contains a selective bridging function that, in the case of a VLAN connection, interprets the destination MAC address of the encapsulated LAN frame and only copies the data to the ports it is destined for. Since this is the place where congestion may arise, the queues have to be quite large (how large exactly is determined later on). Finally the data is converted to (Ethernet interworking) or packed into (ATM-on-top-of-Ethernet) Ethernet frames and then passed onto the lines.

So far, only the transport functions of a board have been described. A board also needs to perform three other tasks, namely interfacing to the lines, interfacing to the backplane and higher layer tasks such as connection control and management. The four functions of a board are performed by four different modules. Figure 9.9 shows the block diagram of a board. The four modules perform the following functions:

- The Line Interface Module (LIM) provides the interfacing to the lines. It performs some lower layer functions and exchanges data between the lines and the SCM. The LIM for ATM-on-top-of-Ethernet consists of several Ethernet MAC controllers.

- The Autobahn Interface Module (AIM) provides the interfacing to the backplane. As described in section 9.5, the backplane is implemented by the "Autobahn" serial bus. The AIM comprises an Autobahn Spaceiver and additional logic for timing and synchronisation purposes.

- The Switch Control Module (SCM) performs the switching functions of the board. It queues data, translates headers and forwards data to the appropriate interface module (LIM or AIM) or to the BCM. In the case of the ATM-on-top-of-Ethernet board, it also performs the packing of ATM cells into Ethernet frames for ATM-on-top-of-Ethernet and interworking with Ethernet.

- The functions of the Board Control Module (BCM) comprise the handling of the upper layer protocols that are not implemented in the SCM, such as connection control, management and LAN emulation. The BCM also initialises the other modules.
The interfaces between the modules contain three types of signals: data signals, control signals and indication signals. The LIM and the AIM are controlled by the SCM which in turn is controlled by the BCM.

### 9.6.1 Line Interface Module (LIM)

The structure of the LIM highly depends on the physical interface that is used. For ATM-on-top-of-Ethernet the physical interface is Ethernet. Therefore, the LIM comprises 16 Ethernet MAC controllers (one for every port). The data bus to the SCM is shared by all the Ethernet MAC controllers, so a centralised data transfer mechanism is required to optimally use the bandwidth of the bus. This is why passive Ethernet controllers are needed. An example of a passive Ethernet controller is AMD's Media Access Controller for Ethernet (MACE). Figure 9.10 shows the functional block diagram of the LIM.
The control block handles the control and select signals from the SCM (for data transfer) and the BCM (for initialisation). It also handles the arbitration between the BCM and the SCM. Since the BCM only accesses the MACEs for initialisation purposes it always has the highest priority. The interfaces from the control block to the MACEs comprise control/select signals, indication signals and data signals. How these signals look exactly is dictated by the MACEs. The data signals from the MACEs to the control block and from the control block to the SCM and the BCM are in reality implemented by a single bus (see also section 9.6.5).

### 9.6.1.1 Description of the MACE

The Am79C940 Media Access Controller for Ethernet (MACE) \[AMD94b\] is a slave register based peripheral that contains a high-speed, 16-bit synchronous system interface that is optimized for an external DMA or I/O processor system. All transfers to and from the system are performed using simple memory and I/O read and write commands. The MACE contains two FIFOs: a transmit FIFO and a receive FIFO. Figure 9.11 shows the block diagram of the MACE.

**Figure 9.11** MACE block diagram

The host system interface of the MACE comprises the following signals (control signals preceded by an asterisk (*) are active low):

- **DBUS15-0 (Data bus).** The data bus is used to read and write data to and from internal registers and the Transmit and Receive FIFOs.
- **ADD4-0 (Address Bus).** The address bus is used to access the internal registers and FIFOs to be read or written.
- **R/*W (Read/Write).** This input signal indicates the direction of data flow during accesses to one of the registers of the MACE, the Transmit FIFO or the Receive FIFO.
• *RDTREQ (Receive Data Transfer Request). *RDTREQ is used by the MACE to indicate that there is data to be read in the Receive FIFO.

• *TDTREQ (Transmit Data Transfer Request). The MACE asserts *TDTREQ when there is room in the Transmit FIFO for more data.

• *FDS (FIFO Data Select). FIFO Data Select allows direct access to the Transmit or Receive FIFO without use of the ADD address bus.

• *DTV (Data Transfer Valid). The MACE uses *DTV to indicate that the read or write operation has completed successfully.

• *EOF (End Of Frame). End Of Frame will be asserted by the MACE when the last byte/word of frame data is read from the Receive FIFO.

• *BE1-0 (Byte Enable). This input signal indicates the active portion of the data transfer to or from the FIFOs. Activation of both *BE0 and *BE1 indicates 16-bit word transfers.

• *CS (Chip Select). Chip Select is used to access the FIFOs and internal registers using the ADD address bus or the *FDS and R/*W signals.

• *INTR (Interrupt). The MACE uses this signal to indicate the occurrence of certain events.

The Receive and Transmit FIFOs can be read or written by using the address bus, *CS and R/*W signals or by using the *FDS and the R/*W signals. In receive operation, the MACE asserts *RDTREQ when the FIFO contains adequate data. It can be programmed to activate when there are 16, 32 or 64 bytes of data available. In transmit operation, the MACE asserts *TDTREQ if the filling level of the Transmit FIFO comes below the Transmit FIFO watermark (which can be configured via a control register).

9.6.2 Autobahn Interface Module (AIM)

The AIM is responsible for the interfacing of the board to the backplane. Motorola has developed the MC100SX1451 Autobahn Spanceiver [MOT93] for this purpose. The Autobahn Spanceiver is a high-speed serial-to-parallel, parallel-to-serial transceiver. The chip is used to implement a high-speed TDM-bus with a maximum effective data transfer rate of 1.6 Gbits/s (future: 3.2 Gbits/s). The block diagram of the Spanceiver is depicted in Figure 9.12.

The interface of the Autobahn Spanceiver to the host system (i.e. the SCM) contains the following signals (control signals preceded by an asterisk (*)) are active low):

• D31-D00 (Data bus). The data bus is used to read and write data to and from internal registers including the Transmit and Receive registers.

• R/*W (Read/Write). This input signal indicates the direction of data flow during accesses to one of the registers of the Autobahn device.
**Figure 9.12** Spaceiver block diagram

- **STRB (Strobe).** This input signal indicates that the data is valid on the parallel bus during a write and that the Autobahn can now place data on the parallel bus during a read.
- **ERROR (Error).** This output signal indicates that the Autobahn has identified a fault condition. The error condition can then be read out from the Error register.
- **FULL (Full).** This output signal indicates that the transmitter or receiver presently contains data.
- **BUSY (Busy).** This output signal indicates that the Autobahn bus is presently in use.
- **REGSEL (Register Select).** This input signal is used to select between the Parallel data register and the control and error register(s).

The Autobahn has three primary operating modes: idle, transmit and receive. After the device has been reset, the default operating mode is idle. The function of this mode is to detect serial bus activity and assert a *BUSY signal. To begin a transfer, data is written into the parallel data register. This event starts an internal timeout timer. The Autobahn transfers the data to the serial transmit register, inserts timing information, which adds one additional bit into the data stream for every octet of data, and shifts the data out on the serial bus. If a new word is loaded into the parallel data register, the next transfer will begin. Otherwise, the bus is held in the state of the last transmitted bit until new data is loaded into the parallel data register or the timeout time expires. The timeout timer runs for a period of four 32-bit transfer times. When the timeout time expires, the device returns to the idle state.
When the Autobahn is operating in receive mode, it strips off the timing information and clocks the data into the serial register. When the register is full, it transfers the data into the parallel data register and asserts the FULL signal pin to indicate the presence of data. The host detects the presence of new data and reads out the content of the data register. In the receive mode, a timeout timer is also started. This timer functions in the same manner as the transmit timeout timer.

How synchronisation of the boards and timing of the slots are realised has yet to be determined. It might turn out that the logic that performs these functions resides between the SCM and the Spaceiver. Therefore, the interface between the AIM and the SCM may differ from the signals described above.

9.6.3 Switch Control Module (SCM)

The Switch Control Module (SCM) is the central part of the board. This is where the real switching functions take place. The block diagram of the SCM is depicted in Figure 9.13. The following blocks can be distinguished:

- The Autobahn Filter Block (AFB) exchanges data with the AIM and stores data in FIFOs. Two FIFOs are present: a transmit FIFO for data to the Autobahn and a receive FIFO for data from the Autobahn. Data received from the AIM is filtered "on the fly" so that only data destined for the board is stored. This is required because otherwise a boards would need to store 1.6 Gbits/s in its FIFOs.

- The Queue Control Block (QCB) contains a number of logical FIFOs, physically implemented in one single memory block. Two memory pools are present: one for low delay traffic and one for low loss traffic. Each pool is shared by several queues each of which belongs to a destination group. The memory also contains the tables for the multipoint distribution entity and the selective bridging function. These tables are updated by the BCM.

- The BCM Interface Block (BIB) is used to transfer data to and from the BCM. It also allows the BCM to update the translation table and to initialize the blocks (not shown in the figure).

- The Address Translation Block (ATB) assembles/disassembles data to and from the lines and performs address translation on the basis of the Address Translation Table. It also adds a routing tag to cells coming from the LIM.

- The Address Translation Table (ATT) is used to translate VPIs/VCIs to add routing tags. Actually, two tables are present. While the BCM is updating one of the tables, for example if it is setting up a new connection, the other table is used by the ATB for translation. After the table has been updated the two tables are switched.

As Figure 9.13 shows, the ATB and BIB control the QCB, whereas the QCB controls the AFB.
9.6.4 Board Control Module (BCM)

The BCM is responsible for performing higher layer functions, like connection control, management and address resolution. For this purpose, it contains a Communications Controller (e.g. Motorola's MC68302 or MC68360).

Another task of the BCM is the initialisation of the other components on the board. For this purpose, the BCM uses a data bus, an address bus and initialisation signals.

Other functions of the BCM are for further study.

9.6.5 LIM-SCM interface

The interface between the LIM and the SCM allows the SCM to communicate with every individual port and transfer data to or from it. To accomplish this, a data bus is needed as well as an address bus (for port selection) and several control and indication signals (e.g. R/*W, *CS).

If the interface complies with a standard, third-party hardware could replace the LIM. This allows the switch to support a wide range of physical interfaces via third-party products. A potential candidate for this standard interface is PMC (PCI Mezzanine Card) as defined by IEEE in draft standard P1386.1 [IEE94b]. This standard provides the specifications for implementing the Peripheral Component Interface (PCI) bus between a host and mezzanine card based on the Common Mezzanine Card (CMC) Standard P1386-1994 [IEE94a]. How the
interface between the LIM and the SCM can be made compliant to the PMC standard is for further study.

9.6.6 SCM-AIM interface

The SCM-AIM interface is the interface that connects the SCM to the Autobahn device. Therefore, it needs to comply with the Autobahn device specifications, of which the signals are described in section 9.6.2. This section also describes that the implementation of the synchronization logic may change the SCM-AIM interface.

9.6.7 SCM-BCM interface

The interface between the SCM and the BCM is used to transfer cells, to initialise the SCM and to arbitrate control over the other modules (the BCM occasionally needs to initialise these modules). How the interface looks exactly is for further study.

9.6.8 Queue dimensioning

Section 9.2 has stated that there is no need to queue cells on the source board. Indeed, the maximum one-way load at the sending board of 80 Mbits/s (the ATM-on-top-of-Ethernet board contains 16 ports each of which support a maximum one-way bit-rate less than 5 Mbits/s) can be easily handled by the Autobahn. If the board is used to connect Ethernet networks, these networks may theoretically produce $10^{16} = 160$ Mbits/s per board which corresponds to 1.6 Gbits/s for 10 boards. Since this does not include the overhead of packing Ethernet frames in ATM cells and routing tag attachment, the total bit-rate may exceed the capacity on the Autobahn. In the rare occasion that all end-systems are Ethernet systems and they all produce 10 Mbits/s, the data is discarded.

So congestion takes place at the receive part of a board only. The dimensioning of the queues is essential for the performance characteristics of the switch. However, since the design of the switch is still in a conceptual phase, the dimensioning of the queues is for further study.

9.7 Summary and conclusions

This chapter has described the design of an ATM switch. The basic switching unit is a shelf which comprises 10 boards. These boards are interconnected by a TDM-bus, implemented by Motorola's Autobahn. Different boards can be designed to implement different functions. The ATM-on-top-of-Ethernet board comprises 16 line interfaces, each of which can be used to connect an ATM end-system or an Ethernet LAN. A routing tag is used to route cells from an inlet on the source board to the multipoint distribution entity on the destination board. This entity sends the cells to the outlet(s). This is also where the cells are queued.

The complexity of the above described ATM switching architecture is relatively low. Furthermore, the system is easily scalable and is capable of supporting sufficient bandwidth for multimedia end-systems. Therefore, it is believed that the architecture of the switch is very suitable for customer premises multimedia networking. The implementation of LAN emulation by a single multipoint-to-multipoint connection and a selective bridging functions allows the
switch to be used for LAN interconnection, which is an extremely useful feature. Although this implementation increases the complexity of the switch, a separate multicast server would probably be even more complex. Therefore, the proposed implementation is worthwhile further study.

There are still several aspect of the switch that have to be worked out. The mutual synchronisation of the boards is one of them. Another important aspect is the determination of the queue sizes.
10 Multimedia end-system design aspects

10.1 Introduction

The network needs to support two types of end-systems: "normal" end-systems that are connected to the network via a LAN, and multimedia end-systems that are connected to the network via ATM network adapters. This chapter describes a low-cost solution for upgrading a Microsoft Windows-based PC to an ATM multimedia end-system. For this purpose, the PC needs to be equipped with an ATM(-on-top-of-Ethernet) network interface, an audio input/output system and a video input/output system.

There are two approaches to turn a Microsoft Windows-based PC into a multimedia end-system. The first approach is software based. It relies on common PC peripherals like the Soundblaster audio system, Ethernet (for ATM-on-top-of-Ethernet) or ATM network adapters and Diamond video boards for multimedia functions. This approach is especially useful when used in combination with ATM-on-top-of-Ethernet, since it allows the upgrading of an existing PC with Ethernet adapter only by installing new software. This software needs to provide ATM functions, synchronisation with the network for isochronous services, and interfacing to Ethernet adapter hardware.

The second approach is hardware based and consists of the design of a multimedia board that comprises audio, video and network systems. This solution is more expensive, but is likely to give better performance since all functions can be implemented in hardware. Since this board is incapable of using the installed base of Ethernet adapter boards, the ATM-on-top-of-Ethernet concept may be less attractive for this board.

For both approaches, knowledge is needed of the Microsoft Windows operating systems. Therefore, this chapter starts with a description of the multimedia capabilities of the various Microsoft Windows families.

10.2 Multimedia capabilities of Microsoft Windows

Currently, three Microsoft Windows families exist: Windows 3.x, Windows NT and Windows 95. Windows 3.x is the most used operating system but is limited and somewhat outdated. Windows NT is a completely different operating system with a user-interface identical with that of Windows 3.x. It is a 32-bit, preemptive multitasking environment with a modular architecture. Windows 95 is the 32-bit successor of Windows 3.x that includes many, but not all, features of Windows NT. Despite the differences between these three families, they also have a lot in common. The Application Programming Interface (API) of Windows 3.x and Windows 95 are subsets of the Windows NT API. Moreover, the three families use similar interfaces enabling the use of drivers and applications on different environments with little (or no) modification. Appendix C gives an overview of the architecture of each family and discusses how multimedia is supported. The most important conclusions are mentioned below.

The Windows operating systems are very complex environments that are not easy to master. Therefore, when developing a multimedia end-system, it is advisable not to deviate from
existing solutions unless this is absolutely necessary. Moreover, the use of existing solutions allows interoperability with other Windows multimedia products.

Windows uses driver stacks to communicate with hardware components. These drivers provide standardised interfaces to the hardware. The interface specification for network drivers is called NDIS (Network Driver Interface Specification) and is specified in [MIC93]. To give Windows access to the ATM network, a new NDIS compliant driver needs to be developed for the network interface.

One of the main problems of multimedia applications on Windows-based PCs is that Windows is not a real-time operating system and the PC architecture is not suited to multimedia applications. This makes the support of true multimedia networking services controlled by Windows nearly impossible. Especially the transfer of large data blocks takes place very inefficiently and heavily loads the processor. Furthermore, it is impossible to synchronise audio and video end-to-end and to deliver audio and/or video in real-time.

To gather knowledge on how real-time services can be supported, several video conferencing systems have been examined. All these systems use the ISDN for transport and integrate an ISDN interface, audio device, and sometimes even a video display device on a single board (or two boards connected via a bandcable). This enables full control and synchronisation of the incoming and outgoing signals without eating up processor resources.

One of the largest problems is the mutual synchronisation of the video and audio streams and the synchronisation of these signals with the network. All examined systems incorporate both the ISDN port and the audio port on the same board, or connect the ISDN board and the audio board via a special cable. This allows the synchronisation of the outgoing audio signal with the ISDN. The video stream, which is sent to a standard video adapter is more or less synchronised to these streams. Corrections are made by omitting or inserting video frames. This technique may be suitable for video streams, but hardly is for audio streams because it will inevitably cause clicks. One system includes a board that mixes video and computer graphics and outputs SVGA signals to the monitor. Since the data is sent directly to the screen, Windows is unburdened from transporting large amounts of video data.

The conclusion can be made that implementing real-time multimedia services on a Windows-based PC can only be done by designing dedicated hardware. Using Windows will always lead to end-to-end QoS degradation.

### 10.3 Software-based approach

The software based approach to a multimedia end-system is depicted in Figure 10.1. The hardware components (e.g. video adapter, audio adapter and network interface) all interface to the Windows operating system via drivers. The network comprises two parts: a multimedia part that also synchronises the clock of the PC to the network, and a data part. The interface of the data part complies with the NDIS standard. All multimedia functions are performed by the Windows operating system.
CHAPTER 10: END-SYSTEM DESIGN

This approach has the following drawbacks:

- The synchronisation of the end-system to the network is performed in software. Such a solution considerably loads the processor. Furthermore, since the operating system has no means to synchronise audio and video hardware, synchronisation can only be realized partially.

- The synchronisation of audio and video streams is performed by the operating system. This also implies a load on the processor. Furthermore, Windows timers are inaccurate and Windows, not being a real-time operating system, is incapable of delivering data in real-time.

- The transfer of data from one Windows sub-system to another causes the data to be significantly delayed.

Due to the drawbacks mentioned above, the software-based solution is not feasible for real-time multimedia networking services. Therefore, this approach can only be used to implement near real-time services. This may be an appropriate solution for "no-budget" multimedia networking.

10.4 Hardware-based Multimedia PC

The previous section has demonstrated that a true multimedia end-system based on Microsoft Windows can only be realized by providing a board that performs the multimedia functions. This board needs to include an ATM (on top of Ethernet) network interface, video display and record functionality and audio display and record functionality. The board also needs to contain a PCI-bus interface for read/write operations to CD-ROM, harddisk and RAM. Along with the board, a set of drivers for Microsoft Windows operating systems that comply with the multimedia PC standards need to be provided. These drivers need to support common APIs and allow application controlled data transfers.
This section proposes a solution for a multimedia board. The board uses Brooktree’s BtV MediaStream chipset, which is described first.

10.4.1 The BtV MediaStream family

Brooktree Corp. created the BtV MediaStream family that brings together graphics, video and digital signal processing to form a single unified architecture. Because many multimedia functions can be handled with just 1 Mbyte of memory, it also forms a cost-effective solution.

![Figure 10.2 BtV chipset for multimedia](image)

Figure 10.2 shows the conceptual diagram of the MediaStream family. The central controller is implemented by the BtV2115 MediaStream controller, the central buffer by Volatile Random Access Memory (VRAM), the graphics/video PACDAC (PAcketized data Digital-to-Analog Controller) by the BtV 2487 PACDAC, the audio subsystem by the BtV2300 AudioStream interface and the video subsystem by the BtV2811 VideoStream decoder. The MediaStream controller supplies a central point of control for all audio, video and graphics. It works with a buffer, called the MediaBuffer, that stores the audio, video and graphics signals each in its own native format. The MediaBuffer connects to the PACDAC, a RAMDAC that accepts packetized data. The AudioStream Interface provides high quality analog audio I/O and the VideoStream Interface provides Video input. The chipset can be expanded with extra functionality, like an MPEG decoder.

Brooktree provides an extensive suite of software support products including display drivers, Windows MCI drivers, diagnostics utilities and applications that exercise the multimedia capabilities of the BtV MediaStream family hardware.
10.4.2 Recommendations on board implementation

An example of the multimedia board that needs to be developed, is depicted in Figure 10.3. The board contains the Brooktree chipset to perform the audio/video functions and an ATM-on-top-of-Ethernet network interface. The clock for the entire board is derived from the network via a PLL (Phase Locked Loop) system that is part of the network interface. This allows end-to-end synchronisation. The network interface splits the information into three parts. Audio and video information is passed on to the MediaStream chipset, whereas data is sent via the PCI bus to the computer’s RAM.

![Multimedia board diagram](image)

Figure 10.3 Multimedia board

10.5 Conclusions

The upgrade of Windows-based PC to a multimedia end-system can be approached in two manners: software-based and hardware-based. Due to several shortcomings of the PC architecture and the Windows operating system, the software-based method is not feasible for real-time multimedia networking. Therefore, a true multimedia end-system can only be
realized by developing an expansion board that comprises both audio/video and network interface hardware.

An example of an integrated multimedia audio/video solution is the BtV MediaStream family. This chipset can be expanded with an ATM network interface to form a true multimedia networking solution.

The software-based approach may still be used on combination with ATM-on-top-of-Ethernet to implement a "no-budget" multimedia solution.
11 Conclusions

This thesis presented the design of a customer premises multimedia network that is based on ATM. It is clear that the Asynchronous Transfer Mode (ATM) is the promising technology that will be used for multimedia networking in the long term. It is the only technology that combines efficient bandwidth utilisation with QoS guarantee and the support of interfaces with different bit-rates. In addition, interfacing to public B-ISDN is optimal. Therefore, ATM is the right technology for a customer premises multimedia network. However, since ATM is still under development, a lot of implementation aspects are under the judgement of the designer.

A multimedia networking solution must distinguish itself from other solutions. It is believed that this can be accomplished by:

- implementing Ethernet interworking functionality so that the ATM switches can also be used as LAN bridges and migration from LANs to ATM is facilitated,
- supporting both synchronous and asynchronous services and embedded clock distribution so that resynchronisation and delay compensation is avoided,
- supporting low-cost interfaces and reducing the cost of the switches by reducing the bit-rate to the end-user.

The architecture that has been presented in this report, provides a scalable network built of switches that internally use a high speed TDM-bus, named "Autobahn". This bus provides a means to switch reasonably high bit-rates (currently up to a total of 1.6 Gbits/s per switch and in the future up to 3.2 Gbits/s) with a relatively low complexity. Furthermore, it facilitates the implementation of multicast and broadcast traffic, thereby making it an excellent solution for LAN interconnection. The switches internally use synchronous time-slot assignment which facilitates the support of isochronous services. The implementation of network clock distribution, which is also needed for isochronous services, has not yet been worked out.

The implementation of VLANs by a single multipoint-to-multipoint virtual channel bring about several consequences to the implementation of ATM switches. These consequences cause an increase in complexity. However, this solution is essential for LAN interconnection and, moreover, provides an embedded distributed multicast server, which is easily scalable. Since it is believed that the cost of a separate multicast server would exceed the additional cost of the proposed implementation, the advantages of this implementation are greater than the disadvantages.

An example of a low-cost ATM interface is the ATM-on-top-of-Ethernet interface, which makes use of existing Ethernet hardware. If an end-system can implement the ATM functions in software, it can use existing Ethernet adapters. The ATM-on-top-of-Ethernet principle, as described in chapter 8, turns out to be useful for low-end multimedia applications.

Examination of Microsoft Windows based PCs yields that it is impossible to implement ATM and real-time multimedia support in software on these PCs. Therefore, a multimedia
networking board needs to be developed, that includes audio and video functionality and an ATM(on-top-of-Ethernet) network interface. Since this does not allow the use of the installed base of Ethernet adapters, much of the attractiveness of ATM-on-top-of-Ethernet is lost. The ATM-on-top-of-Ethernet concept is still useful for "no-budget" ATM interfaces in situations where near real-time services suffice.

During the graduation project, a lot of issues regarding the development of the customer premises multimedia network have been studied. The general aspects of such a network have been discussed and some of these aspects have been worked out in detail. However, there still remains a lot of work to be done.
12 Recommendations

There are still several aspects that need to be studied. The most important are:

• It has to be decided whether or not the software-based near real-time multimedia end-system is worthwhile being developed, since this decision also exerts influence on the development of the ATM-on-top-of-Ethernet concept.

• Further development of ATM-on-top-of-Ethernet. Especially how exactly the end-system synchronises itself with the network and whether the bandwidth is distributed statically or dynamically.

• Further development of the switch. Especially the implementation of the timing and synchronisation on the backplane and the dimensioning of the queues. Subsequently, the building blocks of the ATM-on-top-of-Ethernet boards can be further developed.

• Further development of the end-system. A possible next step is to contact Brooktree to obtain more information about the BtV chipset. Perhaps the multimedia board can be developed in association with Brooktree.

In conclusion, a recommendation on ATM development. The international community is working hard on ATM standardisation. However, there is still a lot of work to do and for some time to come there will be a lot of uncertainties. Therefore, during the further development of the products mentioned above, the standardisation developments need to be watched closely and, if necessary, the design must be adjusted to these developments.
References

[AMD94a] Advanced Micro Devices
Am79C90 CMOS Local Area Network Controller for Ethernet (C-LANCE).
Ethernet/IEEE 802.3 Family World Network Data Book/Handbook.

[AMD94b] Advanced Micro Devices
Am79C940 Media Access Controller for Ethernet (MACE).
Ethernet/IEEE 802.3 Family World Network Data Book/Handbook.

[ARM95] Armbrüster, H.
The Flexibility of ATM: Supporting Future Multimedia and Mobile
Communications.
IEEE Communications Magazine, February 1995, p76-84.


[ATM95a] ATM Forum, LAN emulation SWG drafting group.
LAN Emulation Over ATM Specification - Version 1.0.

[ATM95b] ATM Forum.
Native ATM Services, Semantic Description.

Multiprotocol over ATM.

[AUG93] Augustin, R.J.M.
Implementation aspects of ATM switches
Eindhoven: Technische Universiteit Eindhoven, graduation report nr. 7062.

[DAT93] Dataquest.
Local Area Networks - Europe, Market Trends 1994
Dataquest, 1993

[ECM89] ECMA: TR/44.
An Architectural Framework for Private Networks
REFERENCES


REFERENCES

Information Processing Systems - Open System Interconnect - Basic Reference Model.
American National Standards Association.

Fibre Distributed Data Interface (FDDI).
American National Standards Association.

American National Standards Association.

Token-passing Bus Access Method and Physical Layer Specifications
American National Standards Association.

Token Ring Access Method.
American National Standards Association.

Internal Organisation of the Network Layer (IONL).

[LAU94] Laubach, M.
Classical IP and ARP over ATM.

[MIC93] Microsoft.
Microsoft LAN Manager Network Driver Interface Specification (NDIS) Version 3.0.

MC100SX1451 Autobahn Spaceiver product preview. rev. 0.5.
Phoenix: Motorola, 1993.
REFERENCES

[NEW94]  NEWMAN, P.
Traffic Management for ATM Local Area Networks

[PRY91]  PRYCKER, M. DE.
Asynchronous Transfer Mode, solution for broadband ISDN.

[ROY94]  Roy, R.R.
Networking Constraints in Multimedia Conferencing and the Role of ATM Networks.
AT&T Technical Journal: July/August 1994, p 97-108

[TAN89]  TANENBAUM, A.S.
Computer Networks.
Appendix A Functional description of the network

A.1 Network model

The usage of the network by an end-user application on behalf of a human user can be modelled according to Figure A.1. The figure shows the interaction between the applications and the network services. Two types of applications exist, namely end-user applications and network management applications.

Figure A.1  Network model

A.1.1  Application - network service interactions

The end-user application makes use of the following services:

- Voice calls / Voice conferencing calls.
- Video calls / Video conferencing calls.
- Multimedia calls / Multimedia conferencing calls.
- Data calls / Data conferencing calls.
- Connectionless service (LAN, LAN emulation).

To be able to use these services, the end-user application needs primitives like establish call, release call, join conference, leave conference, send data and receive data.

Regarding network management, five functional areas have been defined by OSI. These areas are:

- Fault management. Fault management is the detection of a problem, fault isolation and correction to normal operation.
- Configuration management. This is probably the most important part of network management. It includes changes, additions and deletions from the network.
- Accounting. This function enables charging for use of managed objects.
• Performance management. This function gathers statistical data on behaviour and effectiveness of managed objects and maintains and examines logs of system state history.

• Security management. Supports authentication, maintains and examines security logs, controls and manages authorization facilities, etc.

Regarding configuration management of an ATM network, the following functions need to be performed by the network manager:

• Network element addition, removal or initialisation.

• Permanent Virtual Path Connections (VPCs) and Virtual Channel Connections (VCCs) management.

• Workgroup management.

• End-user attributes (privileges, addresses, etc.) management

To do this, the network manager needs primitives like: setup VPC/VCC, release VPC/VCC, define workgroup, remove workgroup, add user to workgroup, remove user from workgroup, add user to network, modify attributes of user (access rights, max. bandwidth, etc.), etc.

A.1.2 Subnetwork services

A network can be divided into multiple subnetworks. The interactions between different subnetwork (SN) service layers are:

• SN address assignment control.

• SN routing control.

• SN topology control (to be able to route smartly).

• SN traffic/performance control.

• end-to-end SN PVC setup.

Other interactions are for further study.
A.1.3 Subnetwork elements interactions

The identification of subnetwork elements needs further study. Possible interaction between elements of a subnetwork are:

- Peer-to-peer location update (mobility control).
- SN_connection control (peer to peer).

Other interactions are for further study.

A.2 Interworkings

The definition of interworking is for further study.
Appendix B ATM-on-top-of-Ethernet calculations

This Appendix presents a calculation of the maximum bit-rates on the ATM payload level of the four proposes packing methods for ATM-on-top-of-Ethernet, as described in chapter 8.

B.1 Proposed methods

The four methods that have been proposed are:

- Storage of complete ATM cells in the payload field of an Ethernet frame.
- Storage of cells with a 1-octet header in the payload field of an Ethernet frame.
- Storage of complete ATM cells in the payload field and the Source Address and Destination Address fields of an Ethernet frame.
- Storage of cells with a 1-octet header in the payload field and the Source Address and Destination Address fields of an Ethernet frame.

B.1 shows an overview of the four proposed packing methods and their characteristics.

<table>
<thead>
<tr>
<th>Method</th>
<th>Cell size</th>
<th>Cell header size</th>
<th>Ethernet header+trailer size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method 1</td>
<td>53</td>
<td>5</td>
<td>26</td>
</tr>
<tr>
<td>Method 2</td>
<td>49</td>
<td>1</td>
<td>26</td>
</tr>
<tr>
<td>Method 3</td>
<td>53</td>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>Method 4</td>
<td>49</td>
<td>1</td>
<td>14</td>
</tr>
</tbody>
</table>

B.2 Maximum bit-rate calculations

The IEEE 802.3 standard specifies that the gap between two Ethernet frames that are sent on the medium must be at least 9.6 μs long. At a bit-rate of 10 Mbits/s, this corresponds to 96 bit-periods. To reckon with this gap, 12 octets need to be added to the 26-octet Ethernet frame header and trailer. The maximum bit-rate at the cell payload level is defined as the total number of ATM payload octets in one Ethernet frame divided by the total number of transmitted octets. Now, let N be the number of cells in an Ethernet frame and \( r \), the
maximum bit-rate at the cell payload level. Then, the maximum bit-rates of the four proposed methods \((r_b - r_u)\) are given by

\[
\begin{align*}
  r_{b1} &= \frac{N \cdot 48}{N \cdot 53 + 26 + 12} \cdot 10^7 \quad (N = 1, 2, \ldots, 28), \\
  r_{b2} &= \frac{N \cdot 48}{N \cdot 49 + 26 + 12} \cdot 10^7 \quad (N = 1, 2, \ldots, 30), \\
  r_{b3} &= \frac{N \cdot 48}{N \cdot 53 + 14 + 12} \cdot 10^7 \quad (N = 1, 2, \ldots, 28), \\
  r_{b4} &= \frac{N \cdot 48}{N \cdot 49 + 14 + 12} \cdot 10^7 \quad (N = 1, 2, \ldots, 30).
\end{align*}
\]

The permissible values of \(N\) are obtained by taking into account the maximum possible Ethernet frame size of 1526 octets.

Figure B.1 shows a plot of the maximum bit-rates of the four methods. Two observation can be made from the figure. In the first place, the maximum bit-rate increases with the number of cells in a frame but the difference rapidly decreases when the number of cells increases. The figure also shows that method four provides the highest bit-rate, but the gain compared to method three is small, especially when a large number of cells is packed in one Ethernet frame.

Another issue is the delay introduced by the network when data is transmitted from one system to another. Especially when voice is transmitted, the delay must be small and constant. The magnitude of the delay increases with the size \(S_f\) in octets of the Ethernet frame. The network transports data at a rate of 10 Mbits/s, so the time \(T_f\) to transmit a single frame is given by

\[
T_f = \frac{S_f \cdot 8}{10^7}. \quad (B.5)
\]

It is difficult to determine which value is acceptable for the delay caused by the packing of cells in Ethernet frames. However, as good rule of thumb, this delay ITU recommendation G.114 [G.114] specifies a mean one-way delay of 1.5 ms for a digital local exchange (1.950 ms with 0.95 probability of not exceeding), measured as the time between the analogue signal entering the switch at an inlet and the analogue signal leaving the switch at an outlet. This delay includes the packing/unpacking in frames, transferring and queuing of a cell. This requirement seems a little stringent for ATM-on-top-of-Ethernet switches, but it is a good target.

In addition to packing delay, the propagation of the frame causes a delay of approximately 4 \(\mu\)s/km (coaxial cable).
APPENDIX B: ATM-ON-TOP-OF-ETHERNET CALCULATIONS

Figure B.1  Maximum bit-rates at the cell-payload level

Figure B.2 shows the timing diagram of the ATM-on-top-of-Ethernet protocol. The diagram shows a single time-frame in which both the switch (IS) and the end-system (ES) transmit one fixed-size Ethernet frame. The switch starts transmitting a frame at the beginning of the time-frame. This frame, which takes T_f seconds to be transmitted, arrives T_p seconds later at the end-system, T_p being the propagation delay of the link. The end-system synchronises after the reception of the end of the frame and starts transmitting its own frame T_E seconds after the synchronisation moment. Since T_E is equal to the gap between two Ethernet frame it must be at least 9.6 μs long. After the switch has received the frame sent by the end-system, the link is idle for T_s seconds after which the next time-frame starts. Like T_E, T_s is equal to the gap between two Ethernet frames so it too must be at least 9.6 μs long. According to the figure, the length of a time-frame T is equal to

\[ T = 2 \cdot T_f + 2 \cdot T_p + T_E + T_s = 2 \cdot T_f + T_i , \]

where T_i is the total idle time on the medium.

Now that the structure of a time-frame has been described, the optimal Ethernet frame size S_f can be determined. The following requirements affect the determination of S_f:

- The end-to-end delay introduced by the network must be small, especially when transporting voice. Increasing the size of the Ethernet frame increases the transmission delay, so the end-to-end delay is decreased by decreasing the Ethernet frame size.
The bit-rate on the ATM-payload level must be as high as possible. Increasing the size of the Ethernet frame increases the efficiency of transport and thus the maximum bit-rate.

- The time $T$ which is the time between two consecutive Ethernet frames must be an integer multiple of 125 $\mu$s. This requirement facilitates the transportation of 8KHz voice.

From the third requirement, the potential lengths of the time-frame are limited to the following values:

\[ T = n \cdot 125 \cdot 10^{-6} \quad (n=1,2,...) \]  

(B.7)

The maximum applicable Ethernet frame size, which must be smaller than or equal to 1526 octets, can be calculated from equation (B.5) and (B.6). If the propagation time $T_p$ is neglected, and $T_E$ and $T_S$ are determined at their minimal values of 9.6 $\mu$s, the total idle time $T_i$ is equal to 19.2 $\mu$s. The maximum Ethernet frame size is therefore

\[ S_{max} = \left[ \frac{10^7 \cdot T - 12}{16} \right] \leq 1526. \]  

(B.8)
The number of cells that can be accommodated in one Ethernet frame is equal to
\[ N = \frac{S_{I,\text{max}} - H_f}{S_c}, \]  
where \( H_f \) is the size of the Ethernet header + trailer and \( S_c \) is the total cell size.

The one-way bit-rate on the cell payload level, which is equal to the total number of payload octets divided by the length of a time-frame, can be obtained from equation (B.8) and (B.9):
\[ r_b = \frac{N \cdot 48 \cdot 8}{T} = \frac{384 \cdot \left\lfloor \frac{10^7 \cdot T - 12}{16} \right\rfloor - H_f}{S_c \cdot T}. \]  

Table B.2 shows the time between two consecutive frames \( T \), the number of cells in an Ethernet frame \( N \), the bit-rate \( r_b \) and the total idle time \( T_i \) for each of the proposed methods. The bit-rates are also plotted in Figure B.3.

The table and the figure show results which are roughly comparable to those of Figure B.1. Some values of \( T \) lead to long idle times which results in relatively low bit-rates. Therefore, packing more cells in a frame does not automatically lead to a higher bit-rate.

The optimal packing method is obtained by considering whether the increased bit-rate balance the increase in complexity of a method. The optimal number of cells is obtained by
Table B.2 Possible frame sizes

<table>
<thead>
<tr>
<th>T</th>
<th>Method 1</th>
<th>Method 2</th>
<th>Method 3</th>
<th>Method 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N</td>
<td>r_s</td>
<td>T_i</td>
<td>N</td>
</tr>
<tr>
<td>125</td>
<td>0</td>
<td>0.00</td>
<td>83.4</td>
<td>0</td>
</tr>
<tr>
<td>250</td>
<td>2</td>
<td>3.07</td>
<td>38.8</td>
<td>2</td>
</tr>
<tr>
<td>375</td>
<td>3</td>
<td>3.07</td>
<td>79.0</td>
<td>4</td>
</tr>
<tr>
<td>500</td>
<td>5</td>
<td>3.84</td>
<td>34.4</td>
<td>5</td>
</tr>
<tr>
<td>625</td>
<td>6</td>
<td>3.69</td>
<td>74.6</td>
<td>7</td>
</tr>
<tr>
<td>750</td>
<td>8</td>
<td>4.10</td>
<td>30.0</td>
<td>8</td>
</tr>
<tr>
<td>875</td>
<td>9</td>
<td>3.95</td>
<td>70.2</td>
<td>10</td>
</tr>
<tr>
<td>1000</td>
<td>11</td>
<td>4.22</td>
<td>25.6</td>
<td>11</td>
</tr>
<tr>
<td>1125</td>
<td>12</td>
<td>4.10</td>
<td>65.8</td>
<td>13</td>
</tr>
<tr>
<td>1250</td>
<td>14</td>
<td>4.30</td>
<td>21.2</td>
<td>15</td>
</tr>
<tr>
<td>1375</td>
<td>15</td>
<td>4.19</td>
<td>61.4</td>
<td>16</td>
</tr>
<tr>
<td>1500</td>
<td>16</td>
<td>4.10</td>
<td>101.6</td>
<td>18</td>
</tr>
<tr>
<td>1625</td>
<td>18</td>
<td>4.25</td>
<td>57.0</td>
<td>19</td>
</tr>
<tr>
<td>1750</td>
<td>19</td>
<td>4.17</td>
<td>97.2</td>
<td>21</td>
</tr>
<tr>
<td>1875</td>
<td>21</td>
<td>4.30</td>
<td>52.6</td>
<td>23</td>
</tr>
<tr>
<td>2000</td>
<td>22</td>
<td>4.22</td>
<td>92.8</td>
<td>24</td>
</tr>
<tr>
<td>2125</td>
<td>24</td>
<td>4.34</td>
<td>48.2</td>
<td>26</td>
</tr>
<tr>
<td>2250</td>
<td>25</td>
<td>4.27</td>
<td>88.4</td>
<td>27</td>
</tr>
<tr>
<td>2375</td>
<td>27</td>
<td>4.37</td>
<td>43.8</td>
<td>29</td>
</tr>
<tr>
<td>2500</td>
<td>28</td>
<td>4.30</td>
<td>84.0</td>
<td>31</td>
</tr>
</tbody>
</table>

Considering the importance of increasing the bit-rate and the importance of decreasing the delay. According to Figure B.3, a favourable result is achieved by using the fourth packing method and packing 9 cells in a frame. Further increasing the number of cells yields little gain. The time between two consecutive frames is equal to 750 μs.

The total idle time of this implementation, however, may cause problems. Since T_g needs to be at least 9.6 μs, the maximum value of T_E is 12.4 μs. In this time, the end-system needs to perform the following tasks:

1. The Ethernet hardware must observe that it has received an entire frame and indicate this to the software of the enhanced MAC layer.

2. The software or hardware implementing the Enhanced MAC layer must instruct the Ethernet hardware to start transmitting a frame.
(3) The Ethernet hardware must initiate transmission.

Furthermore, the propagation delay $T_p$ has been neglected. If, for example, the length of the cable between the end-system and the switch is 250 m, the total propagation delay is equal to 2 $\mu$s. Now, the maximum value of $T_E$ is equal to 10.4 $\mu$s.

If it turns out that the end-system needs more time to accomplish this, $T$ may be increased to 1000 $\mu$s, which yields the same bit-rate, but more delay and a total idle time of 27.2 $\mu$s.

In conclusion, a comment on the implementation of packing method four. This method uses a portion of the Ethernet header to store payload so the payload is transported in two units. Since this complicates the packing and unpacking process, it may be advantageous to slightly modify the packing method. Especially when the packing and unpacking process is executed in hardware, which is the case in the switch, it is simplified by using the first unit of 12 octets to store the cell headers and the second unit to store to cell payloads. However, since the length of a cell header is one octet and only 9 cells are stored in an Ethernet frame, 3 octets are wasted by this modification. Table B.3 compares the modified method 4 to method 1 and 2. Since only 12 octets are available in the Ethernet frame header, the modification cannot be applied to method 3 and therefore method 3 is not included in Table B.3. The table shows that method 2 and 4 with $T$ equal to 625 $\mu$s yield the best results. Apparently, the differences between the two methods have disappeared. Since method 2 is less complicated, it is preferred over method 4.

To legitimately select the optimal implementation, one first needs to determine the minimal value of $T_E$.

**Table B.3** Comparison modified method 4 to method 1 and 2

<table>
<thead>
<tr>
<th>$T$</th>
<th>Method 1</th>
<th>Method 2</th>
<th>Method 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N$</td>
<td>$r_s$</td>
<td>$T_i$</td>
</tr>
<tr>
<td>500</td>
<td>5</td>
<td>3.84</td>
<td>34.4</td>
</tr>
<tr>
<td>625</td>
<td>6</td>
<td>3.69</td>
<td>74.6</td>
</tr>
<tr>
<td>750</td>
<td>8</td>
<td>4.10</td>
<td>30.0</td>
</tr>
<tr>
<td>875</td>
<td>9</td>
<td>3.95</td>
<td>70.2</td>
</tr>
<tr>
<td>1000</td>
<td>11</td>
<td>4.22</td>
<td>25.6</td>
</tr>
</tbody>
</table>
Appendix C Windows architectures

This appendix describes the various Microsoft Windows families and how they provide networking and multimedia functionality. The described information is gathered from various sources, which are listed at the end of this appendix.

C.1 Windows architectures

Microsoft Windows comes in many different forms, each of which belongs to one of the three Windows families: Windows 3.x, Windows NT, and Windows 95. At the time of writing, Windows 95 has not yet been released, so this document focuses on the Windows 3.x and Windows NT architectures.

C.1.1 Windows 3.x architecture

Windows 3.x currently comprehends four Windows versions: Windows 3.0, Windows 3.1, Window for Workgroups 3.1, and Windows for Workgroups 3.11. The first member of the Windows 3.x family was version 3.0, which introduced a whole new architecture and caused the great success of this platform. This version has been succeeded by version 3.1 which introduced some new features. Because Windows 3.0 has been entirely replaced by version 3.1, the latter version is described below. The Windows for Workgroups version are described in the Windows networking sections.

Generally, Windows 3.1 architecture can be divided into three primary components: the Windows Application Programming Interface (API), the Windows core and Windows extensions, and the Windows drivers (Figure C.1).

Figure C.1 Windows 3.1 architecture
The goal of the Windows architecture is to abstract the components in such a way that software developers need only understand the published behaviour of the exposed programming interfaces. That is, a software developer can use the Windows API to write an application without knowing the details about how the Windows core routines and the device drivers work. Likewise, a hardware vendor can write a Windows device driver without knowing the details of the Windows core or application internals.

The **Windows core** is the heart of the Windows environment. This layer contains the operating system kernel and support routines. Extensions to the Windows environment are present in the form of Dynamic Link Libraries (DLLs) and provide functionality that can be used by both Windows-based applications and by elements of Windows itself.

The Windows core consists of the kernel, user and GDI routines. The Kernel routine provides the operating system kernel functionality and is responsible for file I/O, memory management, and performing system task scheduling. The User routine is responsible for handling user input and output, which includes managing the keyboard, mouse, sound driver, timer, and handling communication ports. It also manages the user interface, including windows, icons, and dialogue boxes. The Graphics Device Interface (GDI) is responsible for managing the drawing of graphic primitives, manipulating bitmaps, and handling interactions with the device-independent drivers layer, including display and printer output devices.

The **Windows extensions** include additional routines for supporting common functionality universal to all Windows-based applications, such as standardized dialogue boxes for common user interface operations (COMMDLG.DLL), managing Dynamic Data exchange (DDEML.DLL), and multimedia functionality that includes playing and recording audio with waveform and MIDI audio devices (MMSYSTEM.DLL). Video processing is not supported by the multimedia DLL, but is implemented as a separate set of DLLs, referred to as "Video for Windows".

The lowest layer of the Windows environment consists of the **Windows device drivers**. Each device driver tells Windows how to communicate with a particular hardware device, such as a display adapter, printer, network, or audio device.

The display driver, which is of special interest when looking at multimedia applications, manages all screen output for Windows applications. The display driver is a dynamic-link library that consists of a set of graphics functions for a particular display device. These functions translate device-independent graphics commands from the graphic-device interface (GDI) into the commands and actions the display device needs to take to draw graphics on the screen. The functions also give information to Windows and Windows applications about colour resolution, screen size and resolution, graphics capabilities, and other advanced features that may be available on the hardware. Applications use this information to create the desired screen output. If a GDI function is not supported by the display adapter, the GDI implements the function in software.

GDI functions, frequently used by video applications are: StretchDIBits, SetDIBits, and BitBlt. To ensure fast video displaying, these functions must be implemented by the display hardware rather than by the GDI. This imposes some restriction upon the video display hardware.
C.1.2 Windows NT architecture

Although the user interface of Windows NT is more or less identical to the Windows 3.1 interface, Windows NT is a completely new operating system. It supports Windows 3.1, MS-DOS, and OS/2 1.x applications in addition to native Windows NT applications. Windows NT is a 32-bit, preemptive multitasking operating system with a modular architecture.

As Figure C.2 illustrates, the structure of Windows NT can be divided into two parts: the user-mode portion of the system and the kernel-mode portion. Windows NT servers are called protected subsystems because each one resides in a separate process whose memory is protected from other processes by the NT executive’s virtual memory system. Because the subsystems do not automatically share memory, they communicate by passing messages. The solid lines in Figure C.2 represent these messages. All messages pass through the executive, but for simplicity’s sake, those paths are not shown in the figure.

![Figure C.2 Windows NT architecture](image)

Windows NT has two types of subsystems: environment subsystems and integral subsystems. An environment subsystem provides an API specific to an operating system. When an application calls an API routine, the call is delivered through the Local Procedure Call (LPC) facility to the environment subsystem. The environment subsystem executes the API routine and returns the result to the application process by sending another LPC. The most important environment subsystem is the Win32 subsystem, which makes Microsoft’s 32-bit Windows
API available to application programs. In addition, Win32 provides Windows NT's graphical user interface and controls all user input and application output. Windows NT also supplies a POSIX environment subsystem, an OS/2 environment subsystem, a 16-bit Windows subsystem, and an MS-DOS subsystem. These subsystems provide APIs but use the Win32 subsystem to receive user input and to display output.

**Integral subsystems** are servers that perform important operating system functions. The security subsystem records the security policies in effect on the local computer. Several networking components are also implemented as integral subsystems, such as the workstation service and the server service.

The NT executive is the kernel-mode portion of Windows NT. It consists of components, each of which implements two sets of functions: system services, which environmental subsystems and other executive components can call, and internal routines, which are only available to other executive components within the executive. Executive components are independent from one another so a component can be replaced with one that operates differently as long as the new version implements all the system services and internal interfaces correctly.

The executive components are:

- **Object manager.** Creates, manages, and deletes NT executive objects: abstract data types that are used to represent operating system resources.

- **Security reference monitor.** Enforces security on the local computer.

- **Process manager.** Creates and terminates processes and threads. Threads allow a process to execute different parts of its program on different processors simultaneously. The process manager also suspends and resumes the execution of threads and stores and retrieves information about NT processes and threads.

- **Local procedure call (LPC) facility.** Passes messages between a client process and a server process on the same computer.

- **Virtual memory manager.** Implements virtual memory, a memory management scheme that provides a large, private address space for each process and protects each process's address space from other processes. It also transfers selected memory contents to disk when memory usage is too high, and reloads these contents when they are required.

- **Kernel.** Responds to interrupts and exceptions, schedules threads for execution, synchronises the activities of multiple processors, and supplies a set of elementary objects and interfaces that the rest of the NT executive uses to implement higher-level objects.
I/O system. Comprises a group of components for processing input from and delivering output to a variety of devices. The I/O system includes the following sub-components:

- I/O manager. Implements device independent I/O.
- File systems. Translate file-oriented I/O requests into I/O requests for a particular device.
- Network drivers. File system drivers that transmit and receive remote I/O requests.
- Device drivers. Low-level drivers that directly manipulate hardware.
- Cache manager. Improves the performance of file-based I/O by storing the most recently read disk information in system memory.

- Hardware Abstraction layer (HAL). Hides hardware-dependent details such as I/O interfaces, interrupt controllers, and multi-processor communication mechanisms. NT executive components maintain maximum portability by calling HAL routines rather than accessing the hardware directly.

The video display drivers deserve special attention when looking at multimedia applications. As Figure C.3 shows, Windows NT video display drivers are implemented in three parts: the port driver, the Miniport driver, and the display driver.

The port driver is a hardware independent driver that communicates with the Windows NT I/O Manager and the Miniport driver. Only one port driver is needed because it contains no video adapter specific information. The port driver initializes the Miniport driver, handles the synchronisation of multi-threaded requests, and handles many higher level video driver functions, such as video memory mapping and initializing the video for virtual MS-DOS machines (VDMs).

The Miniport driver communicates directly with the video adapter and contains video adapter specific information. There is a different Miniport driver for every video adapter, but because the port driver handles much of the higher-level functionality of the video display and the Miniport driver only provides an interface to the port driver, Miniport drivers are very small. The Miniport driver handles mode switching, interrupt requests, and I/O control codes not processed by the port driver.

The display driver operates in user mode and communicates between the I/O Manager and the Windows NT graphical device interface (GDI) and device driver interface (DDI). The display driver is a portion of the Win32 subsystem, and like the port driver, knows nothing about the video adapter hardware. It does, however, have the ability to request such information if needed by an application. Because the display driver is highly performance sensitive, it is granted direct access to physical memory.

The above-mentioned description shows that Windows NT is a very complex operation system. It requires top-end hardware (486DX 66 Mhz with at least 12 Mb of RAM or better) and it introduces all kind of programming difficulties because it supports multiple processors.
C.1.3 Windows 95 architecture

For further study.

C.2 Windows networking

The networking capabilities of Microsoft Windows have evolved with the Windows version itself. Windows 1.x and 2.x only supported NetBIOS (Network Basic Input/Output System) interface, an API supported by the original Microsoft MS Net and by IBM's PC LAN network software. Windows 3.0 acknowledged the importance of networking by providing special built-in software in the form of a network driver. The conceptual model was a single network sitting under Windows. Windows 3.1 continued this model, but many workplaces do not fit the single network scenario. Windows for Workgroups (3.1) extends the model by supporting a secondary network. Windows NT is a whole different story. A variety of network services may coexist under Windows NT all hidden under the Win32 API. The difference between local and network printers has almost disappeared. Being a 32-bit operating system,
Windows NT includes 32-bit network drivers for which it has developed a standard interface: NDIS 3.0. The latest version of Windows for Workgroups, 3.11, includes the support of this interface and thus 32-bit drivers. For Windows 95, Microsoft has updated NDIS to version 3.1.

C.2.1  Windows 3.x networking

Windows 3.x provides several APIs for network programming:

- NetBIOS. The protocol inherited from MS-DOS and Windows 2.x
- WNet. The networking function calls introduced with Windows 3.0. These functions manage network resources like connections and printers from within Windows.
- Windows Sockets. The API for TCP/IP provided as a DLL, modelled after the Berkeley UNIX API.
- MNet. The Multinet functions unique to Windows for Workgroups. These functions offer support for more than a single network provider.
- Network DDE. The Network Dynamic Data Exchange API, introduced with Windows for Workgroups and Windows NT. This API allows DDE over a network between two cooperating applications. Each program thinks it is communicating with an application local to its machine.
- Other. Vendor-specific networking APIs for Windows. The use of these APIs may be necessary where the native Windows API is not adequate.

C.2.2  Windows NT networking

The Windows NT networking architecture is shown in Figure C.4. The networking model begins at the MAC sublayer of the Datalink layer where the network adapter card drivers reside. These drivers link Windows NT to the network adapter cards.

The networking model includes two important interfaces: the NDIS 3.0 interface and the Transport Driver Interface (TDI), which is the 32-bit equivalent of the Network Control Block (NCB) interface (which is the means by which to communication with the NetBIOS protocol). These interfaces isolate one layer from the next by allowing components to be written to a single standard interface.

Between the two interfaces are transport protocols, which define how data should be presented to the next receiving layer and package the data accordingly. The following transport protocols are available:

Data Link Control (DLC). Mainly used to communicate with IBM mainframes and printers. DLC is fast, but not a full transport protocol.

- NWLink. The Windows NT IPX/SPX protocol stack implementation for Novell Netware support.

- TCP/IP. The Windows NT implementation of the popular UNIX protocol stack.

- Other TDI protocol. Any other transport protocol.

Above the TDI are redirectors, which redirect local requests for network resources to the network. Besides redirectors, Windows NT includes two other components that provide links to remote computers: NetBIOS and Windows Sockets. These two APIs are supplied by separate DLLs, which bypass the Windows NT redirector and communicate with protocol drivers directly using the TDI. The NetBIOS over TCP/IP (NBT) is used for NetBIOS on top of TCP/IP.

### C.2.3 Windows 95 networking

For further study.
C.3 Windows multimedia architectures

C.3.1 Windows audio architecture

Windows provides audio services through high-level audio functions, Media Control Interface (MCI) device drivers, low-level audio functions, the MIDI mapper, and low-level audio device drivers. Figure C.5 shows the relationship between an application and the elements of Windows that provide audio support.

![Windows audio architecture diagram]

Figure C.5  Windows audio architecture

C.3.2 Video for Windows architecture

Video for Windows (VfW) distinguishes between three types of applications: capture applications, playback applications, and editing applications. Figure C.6 shows an overview of Video for Windows.

The following components are included:

- AVICAP.DLL. This component provides services for capture applications. It hides the details of the AVI file format, video and audio buffer management, and the low-level access to video and audio device drivers.
Figure C.6 Overview of the Video for Windows modules

- **MSVIDEO.DLL.** This component provides different services for different types of applications.
  For capture applications, MSVIDEO provides a video channel between the AVICap window and the video-capture device driver. It also enables the AVICap application to take advantage of the Installable Compression Manager (ICM) video codec services.
  The MCIWnd window class can be used to create a window that can playback any MCI media by sending messages to the windows.
  For editing applications, the DrawDib functions of MSVIDEO provide stretching capabilities and low-end display adapter support, in addition to Installable Compression Manager (ICM) support.

- **MCIAVI.DRV.** This file contains the MCI command interpreter. It is used by playback applications to specify instructions for device operations.

- **AVIFILE.DLL.** AVIFILE provides easy read and write access for AVI files. The library is extensible, enabling developers to create custom stream and file handlers. It also provides access to memory files.

- **Installable Compression Manager (ICM).** The ICM provides services to compress or decompress video-image data stored in AVI files.
• Audio Compression Manager (ACM). The ACM provides audio compression and decompression. It operates automatically when a compressed wave stream is rendered.

C.4 Windows timing

C.4.1 Windows timers

The Windows timer is a relatively simple extension of the timer logic built into the IBM PC's hardware and ROM BIOS. PC's contain an Intel 8254 timer chip, which generates a hardware interrupt on IRQO. This interrupt is mapped to INT 8 and can be intercepted by any application that wants a hook to it. The timer chip is programmed to generate an interrupt every 54.925 msec, or about 18.2 times per second. Among other purposes, the BIOS uses Interrupt 08H to update a "time-of-day" value stored in the BIOS data area. MS-DOS uses this value to calculate the current time.

Programs written for the IBM PC and compatibles can use the timer tick interrupt by intercepting Interrupt 08H or Interrupt 1CH (a software interrupt called by the BIOS Interrupt 08H handler). When the hardware interrupt occurs, the program currently running is suspended, and control passes to the interrupt handler. When the interrupt handler is done, it passes control back to the interrupted program.

In Windows, the SYSTEM.DRV driver handles hardware timer interrupts. SYSTEM.DRV sets a new Interrupt 08H vector address during initialization and restores the original vector address before Windows terminates. The Interrupt 08H routine within SYSTEM.DRV calls the original Interrupt 08H handler before doing its own processing so that underlying system functions that require this interrupt will continue to work normally. When SYSTEM.DRV receives an Interrupt 08H, it calls a routine within the USER module of Windows that decrements counters for each timer set by Windows applications. When a counter reaches 0, USER places a WM_TIMER message in that application's message queue and resets the counter to the original value.

Because a Windows application retrieves WM_TIMER messages from the normal message queue, one never has to worry about a program being interrupted by a sudden WM_TIMER message while doing other processing. In this way, the timer is similar to the keyboard and mouse: The driver handles the asynchronous hardware interrupt events, and Windows translates these events into orderly, structured, serialised messages.

SYSTEM.DRV does not attempt to reprogram the 8253 timer chip in the IBM PC. The Windows timer has the same 54.925-msec resolution as the underlying PC timer. This fact has two important implications:

• A Windows application cannot receive WM_TIMER messages at a rate faster than about 18.2 times per second when using a single timer.
• The time interval that is specified in the SetTimer call is always rounded down to an integral multiple of clock ticks. For instance, a 1000-msec interval divided by 54.925 msec is 18.207 clock ticks, which is rounded down to 18 clock ticks, which is really a 989-msec interval. For intervals shorter than 55 msec, each clock tick generates a single WM_TIMER message.

The fact that WM_TIMER messages are placed in the normal message queue and ordered with all the other messages has an important implication: the reception of WM_TIMER messages by a program is not guaranteed to be accurate. As long as another application is busy, the program will not get any WM_TIMER messages. Only when the other application yields control to Windows the program retrieves its next WM_TIMER message from the queue. Windows combines several WM_TIMER messages in the message queue into a single message.

C.4.2 Windows Multimedia timers

The multimedia extensions for Windows include improved timer capabilities: Timer interrupt services, which provide improved timer resolution with up to one-millisecond accuracy. It is suspected that Windows achieves this by reprogramming the timer component.

The multimedia timer services allow an application to schedule timed periodic interrupts or one-time interrupt events at a higher resolution than is available through the standard Windows timer services.

Unlike the Windows timer services, the multimedia timer services are interrupt-based. Instead of posting WM_TIMER messages to a message queue, the multimedia service calls a specified function at interrupt time. Because the callback code is accessed at interrupt time, it must adhere to strict programming guidelines. In particular, only a very limited set of system function calls may be made, and the callback function must reside in a fixed-code dynamic-link library (DLL).

Also, high-resolution, periodic interrupt events require significant processor time. This can drastically affect the performance of your application and any other application running at the same time.

The timeGetDevCaps function is used to determine the minimum and maximum timer-event periods provided by the timer services. These values may vary across computers and can vary depending on the mode in which Windows is running. However, it appears that the minimum and maximum timer-event periods are currently hardcoded 1 ms and 65,535 ms.

C.4.3 Virtual timer services

When running in enhanced mode, Windows takes advantage of the virtual machine (VM) capabilities of the Intel386 hardware. All MS-DOS-based applications run as if each had the entire system to itself. All Windows-based applications run together in a single VM. All VMs, if none are paged, reside simultaneously in physical memory, but only one is active at a time. To arbitrate among the VMs, Windows loads a virtualisation layer when running in Enhanced mode. This layer acts as a traffic cop, directing hardware events to the appropriate VM on the basis of which VM is active.
One of the many hardware events that is virtualised is INT 8, the timer. When a VM is not the active VM, it does not receive INT 8 events at the standard 18.2 Hz rate. The Windows VM not receiving INT 8 events at a standard rate plays havoc with the value returned by the GetTickCount function as well as with any active timers set by Windows-based applications. In this environment, WM_TIMER messages are at the mercy of the Enhanced mode scheduler and the relative priority of the Windows VM. Probably more problematic is the value returned from GetTickCount. Because this value increments by 55 only when the Windows VM sees an INT 8, it is not at all accurate in Enhanced mode. Applications that require accurate event timing must not use GetTickCount to determine time delays.

Windows timers operate erratically in enhanced mode when MS-DOS-based applications are active. The GetTickCount function cannot be used to time events accurately because it is not regularly updated when the MS-DOS VMs are active. GetTickCount is inaccurate in both standard mode and enhanced mode.

C.5 Windows and multimedia networking

Currently, the only available Windows-based multimedia networking applications are video conferencing applications. This section describes some of these applications. All current video conferencing systems for PCs are basically identical, and only differ in the transport medium or compression technique being used. The three most interesting systems are the AT&T Vistium 1200, the Intel Proshare Video System 200, and the PictureTel Live PCS 100. The systems comprise one or two add-in cards, a video camera, and a microphone or speakerphone system. They all rely on the ISDN for the transport of the data and use compression technology to shrink the data and only send the changes that occur between frames. PictureTel uses hardware compression, the other two systems software compression. Even with compression, ISDN-based video does not allow frame rates higher than 10-15 frames per second, which is significantly lower than the 25 or 30 frames per second rate of television. The products use proprietary systems to handle video and data, which prevents them from talking to systems from other manufacturers. The H.320 model, which relies on MPEG-1, standardises the video transmission part of conferencing and is (or will be) supported by most systems. Alas, H.320 has no provision for sharing applications or data. The T.120 standard covers application sharing without moving video images. Regardless of the compression scheme used, all conferencing systems exhibit a transmission lag of about half a second. Although the H.320 standard specifies 320x240 images transmitted at 15 frames per second, the maximum video window size of above-mentioned systems is 192x144 pixels or less. MPEG-2 strives to transmit 640x480 pixel frames at 30 frames per second.

One of the largest problems is the synchronisation of the video and audio streams with the network. All systems incorporate both the ISDN port and the audio port on the same add-in card, or connect the ISDN card and the audio card via a special cable. This allows the synchronisation of the outgoing audio signal with the ISDN. The video stream, which is sent to a standard video adapter is more or less synchronised to these streams. Corrections are made by omitting or inserting video frames. This technique may be suitable for video streams, but it hardly is for audio streams because it will inevitably cause clicks.

The PictureTel system includes a board that mixes video and computer graphics and outputs SVGA signals
to the monitor. Since the data is sent directly to the screen, Windows is unburdened from transport of large amounts of video data.

C.5.1 Windows multimedia platforms

There have been some efforts to design new media architectures. IBM is working on "Lakes", which defines a way to control data streams from an application instead of moving the data to the application. Brooktree Corp. believes that the way PC multimedia systems are built makes real multimedia applications impossible. In response, Brooktree has developed a media-packet architecture that offloads media processing from the host CPU, and capitalizes on the increase DMA bandwidth available with the PCI bus. The architecture, called "BtV", handles graphics, video, and audio data types from the PCI bus, and allows true-color (24-bits per pixel) video at 30 frames per second. A BtV based PCI card that processes MPEG-1, audio, graphics and captures full screen video is expected to sell for about $350.

C.6 Literature

- Desmond, M. Video conferencing coast to coast and face to face. PC World, March 1995, p177-186.