

National Library of Canada

Bibliothèque nationale du Canada

Direction des acquisitions et

des services bibliographiques

Acquisitions and Bibliographic Services Branch

NOTICE

395 Wellington Street Ottawa, Ontario K1A 0N4 395, rue Wellington Ottawa (Ontario) K1A 0N4

Your file - Volte reference

Our tile - Notre reference

AVIS

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments. La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canadä

Virtual Hardware Based ATM Switching Node Test System

Stéphane Gagnon

Electrical Engineering Department



McGill University, Montréal July 1995

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfilment of the Master's degree of Engineering

copyright Stéphane Gagnon 1995



National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street Ottawa, Ontario K1A 0N4 Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington Ottawa (Ontario) K1A 0N4

Your life - Votre rélérence

Our Ma Notre reférence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque Canada nationale du de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission. L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-612-07976-7

Canadä

Abstract

As Asynchronous Transfer Mode (ATM) emerges as the best technology for suitably integrating the various traffic classes of present and future broadband integrated services digital networks, testing methods and equipments associated with this technology have to be devised. The high bandwidth of ATM as well as its flexibility in terms of supported traffic types prevent the use of conventional approaches to network technology testing. Particularily, the conformance assessment of ATM switching nodes becomes a very challenging task because of their complex functionality and the sophisticated performance issues involved. This thesis presents a design and implementation of an ATM switching node test system based on static RAM type Field-Programmable Gate Array (FPGA) technology. Real-time reprogrammability of the FPGA technology used results in a high functionality *virtual hardware system* requiring very few hardware components. The test system features the high performance Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) fibre protocols as the physical link for the connection to the switching nodes under test.

Résumé

Alors que la technologie du Mode de Transfert Asynchrone (MTA) s'affirme progressivement comme étant la seule capable d'intégrer les différents types de trafics supportés par les réseaux numériques à intégration de services (RNIS) à large bande présents et futurs, les méthodes et équipements de test correspondants doivent être mis au point. La large bande d'information ainsi que la grande flexibilité du MTA en terme des différents types de trafics qu'il supporte empêchent l'utilisation des procédures conventionelles de test de réseaux. En particulier, la vérification de conformité des commutateurs MTA peut relever du défi en raison de leur grande complexité ainsi que des questions de performance sophistiquées qui sont impliquées. Cette thèse présente une architecture et un prototype de système de test pour commutateurs MTA basé sur l'utilisation de Field-Programmable Gate Array (FPGA) de technologie à mémoire statique. La reprogrammation du FPGA en temps réel résulte en un système à ressources virtuelles ne requérant qu'un minimum de composantes physiques. Le système de test mis au point utilise pour lien physique avec le commutateur sous test les protocoles de transmission sur fibre populaires Synchronous Optical Network (SONET) et Synchronous Digital Hierarchy (SDH).

Acknowledgments

I am deeply grateful to my supervisor, Professor Ted Szymanski for his support, his dynamism and his trust. I would like to thank the Natural Science and Engineering Research Council of Canada for the generous postgraduate scholarship that supported me through these last two years. The Microelectronics and Computer Systems Laboratory of McGill University is acknowledged for providing me with an excellent scientific environment. Professor David Plant from the Photonics Group at McGill University is acknowledged for giving me full access to the photonics laboratory. Texas Instruments ATM program manager David Copeland and test engineer Carlos Valdivia are acknowledged for providing device samples and assistance.

I would like to thank the MACS laboratory students who made my life easier during these last two years. Among others, I especially salute Boonchuay, Michael and Palash.

Table of Contents

| Chapter 1 | Introduction | page 1 |
|-----------|--|---------|
| 1.1 | Thesis objectives | page 3 |
| 1.2 | Organization of the thesis | page 3 |
| Chapter 2 | ATM technology overview | page 5 |
| 2.1 | Circuit versus Store-and-forward switching | page 5 |
| 2.2 | Asynchronous Transfer Mode | page 6 |
| 2.3 | B-ISDN switching technology | page 8 |
| | 2.3.1 Layered reference model | page 9 |
| | 2.3.2 Quality of service concept | age 11 |
| 2.4 | ATM switching node | age 12 |
| | 2.4.1 Input - Output controllers | age 13 |
| | 2.4.2 Switching fabric p | age 14 |
| | 2.4.3 Control complex p | age 17 |
| 2.5 | Congestion control | age 17 |
| | 2.5.1 Connection level controls | age 18 |
| | 2.5.2 Cell level controls p | age 20 |
| | 2.5.2.1 Preventive control | age 20 |
| | 2.5.2.2 Reactive control | age 22 |
| 2.6 | Signaling | age 25 |
| 2.7 | SONET and SDH physical layers f | age 26 |
| | 2.7.1 STS-1 frame structure | bage 26 |
| | 2.7.2 Multiplexing | bage 28 |
| | 2.7.3 Transport of ATM cells | age 29 |

ļ

| Chapter 3 | Test sys | stem requirements | page 31 |
|-----------|----------|---------------------------------------|---------|
| 3.1 | Definiti | on of testing in ATM | page 31 |
| 3.2 | Networl | k wide versus Network element testing | page 32 |
| | 3.2.1 | Network wide testing | page 32 |
| | 3.2.2 | Network element testing | page 36 |
| 3.3 | Partitio | ning of test parameters | page 37 |
| | 3.3.1 | Functional test parameters | page 37 |
| | 3.3.2 | Performance test parameters | page 38 |
| 3.4 | Generat | tor-Analyzer structure | page 40 |
| | | | |
| Chapter 4 | ATM ti | raffic modelling | page 43 |
| 4.1 | A taxon | omy of traffic source modelling | page 43 |
| | 4.1.1 | Memory based generation | page 44 |
| | 4.1.2 | Stochastic process based generation | page 44 |
| | 4.1.3 | Physical sources based generation | page 45 |
| 4.2 | General | modelling concepts | page 45 |
| 4.3 | Some m | odelling approaches | page 48 |
| | 4.3.1 | Voice traffic | page 48 |
| | 4.3.2 | Video traffic | page 57 |
| 4.4 | PARAS | OL project source modelling | page 58 |
| 4.5 | Stochas | tic hardware | page 61 |
| | 4.5.1 | Random number generation | page 62 |
| | 4.5.2 | Random variable generation | page 65 |
| | | | |
| | | | |

| Chapter 5 | Functional specifications | page 68 |
|-----------|---|---------|
| 5.1 | Statement of specific system design goals | page 68 |
| 5.2 | Presentation of system | page 69 |
| | 5.2.1 Control computer | page 70 |

v

| | 5.2.2 | Test system board | page 71 |
|-----------|------------|---------------------------------|---------|
| | 5.2.3 | Virtual Hardware System | page 72 |
| 5.3 | System | features | page 73 |
| 5.4 | Virtual | hardware system modules | page 74 |
| | 5.4.1 | TDC1500 | page 75 |
| | 5.4.2 | File_transfer | page 76 |
| | 5.4.3 | Cell_error | page 77 |
| | 5.4.4 | Cell_delay | page 78 |
| | | | |
| Chapter 6 | Softwa | re and hardware | page 81 |
| 6.1 | Softwar | re | page 82 |
| | 6.1.1 | File transfer | Dage 82 |
| | 6.1.2 | Cell error | page 83 |
| | 6.1.3 | Cell delay | Dage 84 |
| 6.2 | Hardwa | are | page 85 |
| | 6.2.1 | FPGA design methodology | page 85 |
| | 6.2.2 | Virtual hardware system modules | page 86 |
| | | 6.2.2.1 File_transfer | page 88 |
| | | 6.2.2.2 Cell_error | page 90 |
| | | 6.2.2.3 Cell_delay | page 91 |
| | | | |
| . | a . | | 05 |
| Chapter 7 | System | rototype evaluation | page 95 |
| 7.1 | FPGA | | page 95 |
| | 7.1.1 | Synthesis results | page 95 |
| | 7.1.2 | Speed limitations | page 98 |
| | | 7.1.2.1 Synthesis tool | page 99 |

- 7.1.2.2 Forced routing penalty page 101
- - 7.2 Alternative architectures page 103

| 7.2.1 | Host interface paradigm | page | 103 |
|-------------------|-------------------------|------|-----|
| 7.2.2 | Scalability issues | page | 106 |
| Chapter 8 Conclus | ion | page | 108 |
| Appendix A | | page | 111 |

Chapter 1 Introduction

Over the last decade, local and wide area networks have emerged as a means to increase the utilization of installed resources and reduce the overall costs. Bus and ring technologies have become ubiquitous as their cost dropped to a very affordable level. Nevertheless, these shared media technologies are characterized by a sequential processing of traffic and they are not suitable for the high bandwidth multimedia applications appearing or expected to appear in the future. These applications that are highly desired or needed in an information hungry society have to turn to a new network technology to supply the bandwidth they require.

Asynchronous Transfer Mode (ATM) is a network layer protocol proposal to create a broadband packet switching network capable of transporting a wide variety of services in an integrated fashion. It features a small constant packet size as well as the quasi absence of error control and flow control on a link-to-link basis. The use of reliable optical fibre for the transport of the packets makes end-to-end error control sufficient. Proper buffer dimensioning across the network coupled with rudimentory flow and congestion control mechanisms can insure a minimum quality cell transfer across the network. As opposed to synchronous transfer mode networks where the clients have to reserve a constant bandwidth for the whole duration of a connection, ATM allows clients to reserve bandwidth non exclusively. Each link bandwidth is therefore statistically multiplexed among all of its users. This multiplexing makes ATM very suitable for variable bit rate traffic transport such as computer data or compressed video signals.

At the heart of an ATM network are the switching fabrics. These switching fabrics are patterned collections of simple switching building blocks that serve to move cells transparently from their origin to their destination. In addition to the switching activity itself, they can also be the siege of signaling, flow control and congestion control functions. All switching nodes of the network are connected together to make the traffic forwarding possible and also to support the signaling system needs.

As ATM technology is emerging in both wide area networks and local area networks, the needs for efficient test mechanisms arise. First, at the highest level, the network has to include some embedded mechanisms performing real-time cell transfer quality monitoring, fault monitoring, fault identification and fault location. Then, at a lower level, the various elements composing the network have to be individually evaluated before being incorporated to the network. This evaluation consists of a thorough conformance assessment of the equipments proceeding from manufacturer specifications. For instance, cell loss probability as well as cell transfer delay through a switching node have to be measured. Given the high bandwidth and the high flexibility in terms of supported traffic types in ATM, the conformance assessment of the ATM equipment can become a very challenging task.

This thesis proposes a design and implementation of an ATM switching node test system. The system allows the evaluation of cell transfer quality inside a switching node in terms of bit error rate, cell loss probability, cell loss misinsertion and cell transfer delay. Additionally, the system can evaluate various functionalities of the switching nodes such as traffic policing and signaling. The test system proposed makes use of hardware that can change its functionality in real-time and therefore it constitutes an application of what is sometimes called virtual hardware, hardware subroutines or silicon multi-tasking. A static RAM based Field-Programmable Gate Array (FPGA) is used to provide the run-time hardware metamorphosis needed. The test system uses the synchronous optical network (SONET) and synchronous digital hierarchy (SDH) protocols to connect to the switching node under test.

1.1 Thesis objectives

The objectives of this thesis are three-fold. First, it constitutes a research and survey of the design issues and features associated with an ATM switching node test system. Second, it is targetted at the integration of the current ATM networking parcelled knowledge into a coherent and updated big picture. Indeed, the ATM switching node test system concerns all aspects of the ATM technology spectrum, from the switching to the signaling. Third and last, this thesis evaluates the use of run-time reconfigurable field-programmable gate array technology as the heart of a virtual hardware system.

The ATM switching node test system presented in this thesis is associated with an article published in the Proceedings of the 3rd Canadian Workshop on Field-Programmable Devices (FPD'95) and entitled *Field-programmable gate array based ATM switching node test system* [1].

1.2 Organization of the thesis

Chapter 2 consists in a broad overview of the current status of ATM technology. It includes the description of the building blocks of the network as well as the various switching node architectures. It presents a survey of the various congestion control mechanisms envisioned to be used in ATM. Finally, the two most popular optical fibre communication protocols, namely SONET and SDH, are presented in view of ATM cell transport.

Chapter 3 depicts the task of testing in ATM. It partitions the testing activity into embedded mechanisms responsible for run-time performance monitoring of cell transfer in the network and individual network element testing using dedicated test equipment. Further, the network element testing parameters are extracted and classified as functional parameters and performance parameters. The functional structure of an ATM switching node test system is derived.

Chapter 4 presents the issues associated with the cell generation process required by the test system. A taxonomy of traffic source modelling techniques is presented as well as some interesting modelling approaches found in the literature. Modelling concerns for voice and video sources are addressed. An impressive traffic generation system engineered by the Research & Development in Advanced Communications Technologies in Europe (RACE) called PARASOL is also presented. Finally, hardware synthesis of stochastic processes is studied as it is required by the test system proposed.

Chapter 5 proceeds to the functional specifications of the proposed ATM switching node test system. These specifications enumerate and describe the functionalitites of the system without entering into the implementation details. The system is partitioned into four different modules composing the virtual hardware system. Each module is responsible for a specific test and becomes an icon in the graphics user interface program controlling the test system hardware.

Chapter 6 describes the hardware and software specifications of the test system architecture devised. Each module of the virtual hardware system is described in terms of a FPGA configuration and a set of software routines. The FPGA design methodology and the synthesis tools are also presented.

Chapter 7 proceeds to the evaluation of the architecture devised and the prototype that was built. Each FPGA design is characterized in terms of area and timing statistics. The timing limitations of the system are analyzed and explained. Alternative architectures are introduced and compared with the proposed architecture.

Chapter 8 draws the conclusions from the ATM switching node test system proposed and implemented.

Chapter 2 ATM technology overview

This chapter comprises a broad overview of the current status of ATM technology. It includes the description of the building blocks of the network as well as the various switching node architectures. It presents a survey of the various congestion control mechanisms envisioned to be used in ATM. Finally, the two most popular optical fibre communication protocols, namely SONET and SDH, are presented in view of ATM cell transport.

2.1 Circuit versus Store-and-forward switching

Circuit switching and *store-and-forward* switching are the two main paradigms in the field of networking technology [2]. In circuit switching, the bandwidth of each communication link is usually split among clients using time division multiplexing. Each link carries consecutive equal length frames, each being composed of a fixed number of fixed length time-slots assigned to the clients. Alternatively, the link multiplexing can be done in the frequency or wavelength domains. In any case, the connection between two end-points of a circuit switched network requires that a specific time-slot or frequency-wavelength channel be reserved on each link along the end-to-end path for the duration of the session. The switching nodes of such networks accomplish both timefrequency-wavelength switching and space switching of client channels. The communication delays are mainly due to propagation times and the jitter on delay is practically non existent.

The main shortcoming of circuit switching is its poor ability to offer variable bandwidth and bandwidth on demand. Systems offering variable bandwidth are called multirate circuit switching, as they allow clients to be allocated multiple basic channels. These systems remain complex because the individual channels of a connection have to be switched simultaneously in order to keep delays equal among the channels of the connection. Fast circuit switching is another improved variation targetted at bandwidth on demand through dynamic basic channel allocation to the various connections.

In the store-and-forward switching mode, end-to-end connections are established without necessarily reserving the required transmission bandwidth. In this mode, the multiplexing of link bandwidth among users is rather done on an as-needed basis rather than fixed basis. The information of clients is transmitted through the network as streams of packets or messages that are stored in each node before being forwarded to the next using the full bandwith of the physical link. If this switching mode leads to an effective utilization of link bandwith, in return it introduces a series of distributed queueing delays that can be hard to control.

Accordingly with the common taxonomy associated with store-and-forward switching [2], message switching consists in sending messages as unit entities rather than breaking them into packets, namely packet switching. The routing of information in store-and-forward switching can be connection oriented (virtual circuit routing) or connectionless (datagram routing).

2.2 Asynchronous Transfer Mode

Asynchronous transfer mode is a packet switching technology sometimes qualified as *fast packet switching* and defined by the American National Standards Institute (ANSI) and the International Telephony Union (ITU), formerly known as Comité Consultatif International de Télégraphie et de Téléphonie (CCITT). Its key characteristic is its capability of statistically multiplexing variable bit rate sources on transmission links at constant bit rate. Its small fixed packet size and the quasi absence of error control and flow control on link-to-link basis make it suitable for the integration of many digital services. The need for link-to-link error control is eliminated by the use of high quality physical communication media such as optical fibre, which is known to achieve a bit error probability as low as 1×10^{-10} . Proper resource allocation and queue dimensioning in the network is expected to guarantee a minimum quality of service without the use of complex flow control mechanisms. The cell structure used in ATM consists of a 5-byte header and a 48-byte payload, as illustrated in Table 2.1.

| BYTE | 8 | 4 3 | 2 | | |
|------|--|----------------------------------|--------------------------|--|--|
| L | Generic Flow Control (GFC) or Virtual Path Identifier (VPI) | Virtual Path Identifier (VPI) | | | |
| 2 | Virtual Path Identifier (VPI) | Virtual Circuit Identifier (VCI) | | | |
| ં 3 | Virtual Circuit Identifier (VCI) | | | | |
| 4 | Virtual Circuit Identifier (VCI) | Payload Type (PT) | Cell Loss Priority | | |
| 0.5 | Header Error Control (HEC) | | | | |
| 6 | Payload Byte # 1 | | | | |
| | • • | • | | | |
| 53 | Payload 1 | Byte # 48 | | | |

Table 2.1 ATM cell format

The most important functionality of the cell header is to provide routing information through the use of its virtual path identifier (VPI) and virtual circuit identifier (VCI) fields. The VPI provides the coarse level of routing information whereas the VCI provides the finer level of routing information. Alternatively, a VPI can be seen as a bundle of VCIs that could all belong to the same end-user, a corporation head office for instance. The header error control byte provides error detection and correction capabilitites for the first 4 bytes of the cell and is also used for cell delineation in ATM receivers.

The ATM network itself consists of an interconnection of ATM switches spanning a geographical area whose size depends on the particular network implementation, namely LAN, MAN or WAN. Through special devices providing user-network interface (UNI), clients can access the network in order to send and receive cells. ATM being primarily meant to be connection oriented, a connection establishment phase prior to each call will trace a suitable path between the sender and receiver, and will add the corresponding routing entries in each switch database crossed by the particular connection being setup. More precisely, this routing entry consists of an incoming VPI/VCI pair, an outgoing VPI/VCI pair and a switch output port identifier. When a cell reaches a switch, its VPI/VCI is used as the key of a look-up table search in the switch database. From this search, the new VPI/VCI values of the cell are obtained as well as the output port identifier where the cell should be forwarded. So, while an ATM cell crosses the various network switches leading to its destination, its header VPI/VCI values get changed successively and this process is referred to as *header translation*. This translation is necessary since the routing fields of a cell are only meant to have a local significance, as opposed to an otherwise end-to-end or global significance. Given this local significance, a specific value of the routing fields can be reused at will across the network.

2.3 **B-ISDN** switching technology

A broadband integrated services digital network (B-ISDN) is a network architecture designed to accomodate various types of traffics like data, voice, images and video. Applications and services are expected to expand rapidly once these networks acquire a bigger share of the WAN and LAN markets. On the basis of its numerous strengths, ATM has been chosen by standards committees (ANSI T1, ITU SG XIII) to be the uniting underlying transport technology of B-ISDN.

2.3.1 Layered reference model

In order to describe the functionality of B-ISDN, CCITT recommendation 1.321 [3] introduced the B-ISDN protocol reference model. This model is a layered architecture following the seven layer reference model of Open Systems Interconnection (OSI) defined by the International Standards Orgranization (ISO). The layered approach to data network specification consists in partitioning the networking task into layers containing specific modules. Each module implements a function (e.g., provides a service) in support of the overall task and is implemented through a software process or a hardware device. The particularity of these modules is that they are distributed. For instance, when a connection is setup between two nodes of the network, corresponding modules in corresponding layers are created at both ends of the connections. One such pair of distributed modules is called a pair of peer processes and could be responsible for end-to-end exchanges of flow control, congestion control or error control information related to the connection. The protocol reference stack for B-ISDN is presented in Figure 2.1. All elements of the B-ISDN such as switching nodes and user network interfaces (terminals) have to comply with this reference model.



Figure 2.1 B-ISDN protocol reference stack

The B-ISDN reference model is partitioned into the usual layers and additionally into planes. The user plane can be considered as the most important as it regards the transfer of user information. The control plane is responsible for call establishments through signaling functions and for other connection control functions. The plane management and layer management planes provide management functions and allow the interworking of the user and control planes.

In the B-ISDN reference model, ATM appears as a set of three layers shared by the user and control planes. The ATM adaptation layer (AAL) is the highest layer of ATM and provides functions for converting user information into the 48 byte payload units that are required by the ATM layer for transmission. The ATM adaptation is a two-step packetization process of user information. First, this information is packetized into variable length units called *convergence sub-layer protocol data units* (CS-PDU) by the convergence sub-layer of AAL. Then, these CS-PDUs that can be as long as 64KB in the case of AAL-3/4/5 are packetized into smaller 48 byte units called *segmentation and reassembly sub-layer protocol data units* (SAR-PDU) by the segmentationreassembly (SAR) sub-layer and are passed down to the ATM layer for transmission. Both sub-layers of AAL add their own overhead to the user information for such purposes as protocol data units ordering and end-to-end error control. On the receive side, the whole process is reversed such that the SAR-PDUs received from the ATM layer are reassembled into CS-PDUs, then the payload is extracted from these CS-PDUs and passed to the upper layer, which may or may not be the final application. Different AAL types are defined (AAL-1 through AAL-5) for supporting different classes of traffic.

The ATM layer is responsible for appending the suitable header to SAR-PDUs received from AAL before forwarding them down to the physical layer. Similarly, cells received from the physical layer have their header stripped off and are passed to AAL. In the case of switching nodes, the ATM layer provides the header translation and routing functions.

The physical layer transmission functions perform the formatting of the transmitted cells according to the transmission protocol and medium used. The associated receive functions provide bit timing, cell delincation, HEC verification and extraction of idle cells.

2.3.2 Quality of service concept

With the advent of B-ISDN and in particular its underlying asynchronous transfer mode, the concept of fixed quality of a connection (which was taken for granted in synchronous transfer mode environments) does not apply anymore. The asynchronous nature of ATM leads to dynamic variations of the load across the network and in turn, these introduce variations in the quality of service achieved on the various established connections. The Quality of Service (QoS) of a particular connection can usually be described in terms of cell transfer delay, cell delay variation and cell loss probability. In order for B-ISDN and ATM to be interesting from the user's point of view, the network has to be designed, operated and managed in such a way that the QoS can be guaranteed. Given that ATM's most interesting features rely on statistical multiplexing, the guarantees mentioned here are not strict but rather have to be expressed by means of likelihood and probabilities.

Accordingly with the taxonomy developed around ATM, a QoS class is defined as a set of objective values for each of the performance parameters of a connection, namely cell delay and cell loss ratio. It is expected from B-ISDN implementations to provide the users with various QoS classes. In Figure 2.1, four such preliminary classes A,B,C and D are shown. Each class has its parameter objective values chosen to fit a particular type of traffic source. Upon call setup, the user should be able to specify the QoS class desired for its connection. Then, the associated quality of service should be guaranteed by the network for the whole duration of the session.

Essentially, the QoS can be guaranteed in the network by first preventing congestion and, in case of congestion, by selectively penalizing the connections according to their requested QoS. Mechanisms to prevent and manage congestion include bandwidth allocation, rate flow control, window flow control, credit flow control, transmission scheduling, buffer space management, cell tagging and selective cell discard. They are described in section 5 of this chapter.

2.4 ATM switching node

The ATM switching node is the corner-stone of ATM technology. It consists of a switching fabric transmitting and receiving traffic through its input-output controllers and that is under the authority of a control complex. A switching node architecture is generally characterized by its queueing strategy (input, output, shared) and its traffic processing priority scheme. The various logical components of the switching node are illustrated in Figure 2.2.



Figure 2.2 ATM switching node

2.4.1 Input - Output controllers

The input and output controllers are responsible for interfacing the switching node with the incoming and outgoing physical links. They are the siege of the implementation of the physical layer of the switching system. The input controllers extract the cells from the various incoming physical protocols used (SONET, SDH, TAXI, DS1, DS3) whereas output controllers provide the cell formatting capacities for the various outgoing physical protocols. Input controllers can be considered as cell demultiplexers that are responsible for translating the header of the incoming cells and for separating these cells according to their respective destinations. They can also provide cell buffering and cell duplication for multicast connections, depending on the type of switching node implemented. The input controllers may include sensor functions associated with

operation and management tasks like traffic monitoring, policing and extraction of OAM cells. Output controllers can be considered as cell multiplexers that are responsible for pooling the cells destined to the same output port in a common queue before they are transmitted.

2.4.2 Switching fabric

The switching fabric is responsible for effectively transporting incoming cells from input links to their destined output links. Given the asynchronous mode, incoming cells on different input ports of the switch will sometimes compete for the same output link. This output contention phenomenon is what makes the design of ATM switches such an enormous challenge, as it brings along the need for queueing of cells inside the switches.

A wide variety of ATM switching fabric architectures have been devised during the past few years. Each of them is characterized by the particular way it handles output contention, the structure or medium it uses to forward the cells to the output ports of the switch and its traffic processing priority scheme. An attractive classification of switching architectures taken from [4] is presented in Figure 2.3.

The first fork in the classification tree partitions the switches according to the time and space division of the switching task. Time division switching usually involves the sharing of a single resource among the many input ports of the switch. This single resource that is being shared can be a ring, a bus or a memory. In all cases, the access to the resource by input ports must be mutually exclusive, thereby preventing the scalability of this type of architecture. Indeed, the more inputs are added to the switch, the shorter is the access time to the shared resource. These architectures usually employ some form of internal speed-up or a high degree of parallelism in order to maximize their aggregate bandwidth.



Figure 2.3 ATM switching fabrics classification

Shared memory architectures [5,6,7,8] differ from shared bus architecture in that the buffering space is shared among all input ports. Shared bus architectures have individual dedicated buffers for all inputs or outputs. Sharing the memory among all switch ports leads to a more efficient use such that for comparable performance, less memory is required than for the case of non-shared memory architectures. Neverthless, the use of more expensive multi-ported fast access memory with wide datapaths is needed to counteract the sequential nature of the shared memory and improve the aggegate bandwidth.

Space division switching architectures are characterized by their ability to forward many cells concurrently. These architectures usually result from the interconnection of small building blocks like two-input two-output nodes (2x2 nodes). For instance, N^2 such 2x2 nodes can be connected in a N by N array to form the classic non-blocking N by N crossbar switching fabric. Alternatively, a fewer number of 2x2 building blocks can be connected together in a multistage configuration (Banyan, Delta, Shuffle) resulting in a N by N internally blocking switching fabric. A switching fabric is qualified as

internally blocking when some of the permutations of its inputs cannot be achieved without internal contentions. Then, in order to use these internally blocking switching fabrics in ATM without risking cell losses, buffering can be provided inside the 2x2 building blocks. Alternatively, in a more complex approach, a front-end switch controller [9] can be used to continuously schedule batches of cells that can be forwarded concurrently without producing internal blocking.

Thus, the internal blocking property of some classes of space switches can be taken care of by proper buffering and scheduling strategies. Additionally, as with all space switches, the concurrent forwarding of cells gives rise to the second type of switching conflict known as output contention. This conflict can be taken care of by queueing contending cells at each input port. Simple input queueing is easy to implement but has been shown to limit the line utilization factor to 0.586 [10] under a uniform traffic assumption. The relatively low utilization factor is due to the head of line effect, or in other words, the fact that a buffered cell can be prevented from accessing an available output port because the cell ahead of it in the buffer is blocked due to output contention. This head of line effect can be ruled out if special input queues allowing departure of cells at arbitrary positions (bypass queues) are used instead of the simpler first-in first-out type queues. A front-end controller is then used to schedule batch departures of cells going to distinct outputs. The combination of bypassed input queues and scheduling can force the line utilization factor to approach one [11].

Instead of using these complex bypassed input queues and their associated cumbersome scheduler, the switching fabric can be built as a series of parallel space division switches coupled with output queues. Advantages of having parallel switching planes are two-fold. First, cells competing for the same output can be forwarded simultaneously on different planes. Second, if the planes are realized as inexpensive internally blocking space switches, the front-end scheduler preventing internal blocking will be simpler than if the switch only had a single plane. The reason for this is that the multiple planes available will potentially lead the switching fabric to full line utilization

even though the individual planes themselves are not used optimally. This non optimal use of the individual planes will allow the scheduling algorithm and therefore the scheduling resources to be minimum.

In summary, space division type switching fabrics tend to be far more scalable than their time division counterpart. In the context of B-ISDN where switches with thousands of high speed ports are needed, scalability is the leading issue. Surveys of switching architectures for ATM can be found in [12,13,14,15] whereas issues of scalability and physical limitations of large size ATM switching fabrics are surveyed in [16,17,18].

2.4.3 Control complex

The control complex of the switching node is not involved in the actual task of switching but rather in the operation and maintenance activities required for the network to function. Information related to these activities is carried out through the network by operation and maintenance cells (OAM) whose format follows some network wide signaling protocol. Through its local implementation of the signaling protocol, the control complex can receive and send OAM cells. Such OAM cells may be used for such tasks as connection setup and exchange of congestion information between nodes.

2.5 Congestion control

Under the first impression, it may seem unreasonable to consider congestion control mechanisms in broadband ATM networks, given the high bandwidth characterizing their communication links. The fact is that images and video applications, among others, are traffic sources whose burstiness and peak cell rate are enormous. Additionally, the future needs of an information hungry society will surely bring along new services with unsuspected characteristics. Despite the fact that ATM was first forseen to be a *best effort* technology, extensive research has been going on about the trade-offs of integrating some congestion control mechanisms in ATM. As will be seen, many of these mechanisms rely on the regulation of source traffics according to the state of congestion along the connection paths. Obviously, modulating a source that exhibits a real-time character (voice, video, remote process control) does not make much sense. That is why a new ATM service category called the available bit rate (ABR) service [19] is being added to the existent continuous and variable bit rate service categories. The ABR category is introduced to support applications with vague requirements of throughput and delays. This is precisely this service category that could be regulated by congestion control mechanisms in order to fill the bandwidth gaps in the networks. Congestion prevention mechanisms are usually partitioned into *connection level* and *cell level* mechanisms and they are presented next.

2.5.1 Connection level controls

Connection level congestion control mechanisms are active during the setup of each connection and they are mainly responsible for :

- Path selection and admission-rejection of a new connection.
- Bandwidth allocation-deallocation for the new or torn down connection.

The path selection and the admission-rejection of a new connection proceed from the traffic descriptors provided by the user in the connection request. Such descriptors shall include peak cell rate, average cell rate and maximum burst duration. From these parameters, the routing process consists in finding a path between the source and the requested destination whose links can accomodate the additional statistical traffic multiplexing of the requested connection. The additional traffic brought by the new connection should not jeopardize the maintain of the quality of transmission (cell loss probability, cell delay variations) of other connections. Therefore, in order to dynamically setup and tear down connections in the network, every link state (remaining unused statistical bandwidth on a link) should be accessible through a distributed



database. The database content should be updated continuously to reflect the current state of the network links.

Each time a new connection is added to or removed from a link, the current status of the link in the database should be updated accordingly and this action is called bandwidth allocation or deallocation. Real-time modifications of connections in the network pose a stringent requirement on the simplicity and efficiency of the algorithms used for bandwidth allocation. An attractive algorithm reported in [20] introduces the concept of equivalent capacity (cj) as being the link bandwidth required by a connection j with peak rate Rj cells/sec, mean rate mj cells/sec and mean burst duration bj sec, when the available buffer space for feeding the link is equal to X cells and the desired buffer overflow probability is ϵ . In practice, the equivalent capacity expressed in cells/sec can be computed using Equation 2.1 and its value lies somewhere between Rj and mj. The difference (cj - mj) can be seen as the cost of limiting the buffer overflow probability to ϵ .

$$c_{j} = R_{j} \frac{y_{j} - X + \sqrt{[y_{j} - X]^{2} + 4X\rho_{j}y_{j}}}{2y_{j}}$$
(2.1)

$$y_j = \ln\left(\frac{1}{\varepsilon}\right) b_j (1-\rho_j) R_j$$
 (2.2)

$$\rho_j = \frac{m_j}{R_j} \tag{2.3}$$

Now, the aggregate capacity of a link with buffer space X and exposed to N sources with parameters $\{(R_j, m_j, b_j) \mid j \in \{1, 2, ..., N\}$ can be obtained using the equivalent capacity (c_j) of each multiplexed source, as shown in Equation 2.4. The aggregate capacity required for the multiplexed link will be equal to the summation of the individual equivalent capacities in the worst case and will otherwise be smaller as a

consequence of the statistical multiplexing.

$$C_{aggregate} = \min\left(m + \alpha\sigma, \sum_{j=1}^{N} c_j\right)$$
(2.4)

$$\alpha \approx \sqrt{2 \ln \left(\frac{1}{\varepsilon}\right) - \ln(2\pi)}$$
 (2.5)

$$m = \sum_{j=1}^{N} m_j$$
, $\sigma^2 = \sum_{j=1}^{N} \sigma_j^2$, $\sigma_j^2 = m_j (R_j - m_j)$ (2.6)

Connection admission and bandwidth allocation are thus the main mechanisms of congestion control at the connection level.

2.5.2 Cell level controls

Cell level congestion mechanisms are those acting on each individual connection and during their whole duration. They are otherwise known as flow control mechanisms. They can follow a preventive or reactive philosophy and they constitute an important field of research in ATM because this technology will not be useful if it cannot offer the quality of service promised, that is, some guarantees on the quality of information transfer to its users.

2.5.2.1 Preventive control

Preventive congestion control mechanisms are also known as open loop control mechanisms since they operate with predefined rules rather than considering the current state of congestion of the network. A mechanism that does not consider the current state

of the network might at first seem over simplistic. Precisely, these preventive methods have for incentive their implementation simplicity. Also, if the state of congestion of the network is a process whose variation dynamic is fast (as would be the case with variable bit rate type traffic), then the benefits of modulating the traffic sources with the congestion information are not guaranteed. For instance, by the time the congestion information is used for the traffic source modulation, the state of congestion may already have evolved into something different.

Rate control is a class of preventive mechanism consisting in the individual regulation of the rate of traffic of the various connections. This regulation can be applied at the network entry points as well as inside the network itself. The most popular rate control mechanism is called *leaky bucket* and is illustrated in Figure 2.4. It is composed of a token pool that is regenerated at a certain rate. The head of line cell in the data buffer will only leave the buffer if a token is available from the token pool. Thus, the leaky bucket controller does not eliminate burstiness from the incoming traffic. In fact, the controller cell departure process will allow a maximum burst duration equal to the capacity of the token pool.



Figure 2.4 Leaky bucket controller

The goal of the leaky bucket is not to smooth traffic but rather to restrict each connection traffic to the parameters contracted at setup phase. Indeed, the parameters of the leaky bucket, namely token pool size and token arrival rate can be derived from the connection parameters in such a way that the leaky bucket output traffic conforms to the parameters contracted for the connection [21,22]. Alternative methods for rate control include the jumping window and the moving window [23] mechanisms which consist in limiting the number of cells from a source to a certain number N during a given time window T. Other methods claimed to be more effective, control the probability density function of the sources rather than their peak and mean cell rate [24].

Transmission scheduling [25] is a type of control applied at each output port of switching nodes. It is a mechanism that controls how many cells from each connection will be sent on an output link of the switch during a certain interval and also the order in which the cells are going to be issued. This virtual scheduling of cells is particularly interesting in the presence of traffic classes having different delay requirements since it allows the prioritization of traffic processing.

Finally, buffering policies creating a partitioning and prioritization of buffering space in the switching nodes can be used to distribute the consequences of a state of congestion. For instance, high priority buffer space can be reserved for classes of traffics with stringent quality of service parameters, thereby pushing eventual congestion effects toward low priority classes of traffic.

2.5.2.2 Reactive control

Reactive control mechanisms are those behaving like feedback systems. Using some periodically updated measure of the congestion status of the network, they regulate the cell emission of each connection accordingly. Again, this traffic regulation may be applied directly at the sources or may be distributed all over the network. Two schemes of reactive mechanisms are often proposed [26], namely credit *credit based* and *rate based* mechanisms.

The credit based scheme is also qualified as link-by-link window flow control since it is meant to act individually on every connection of every link. The receiving end of every link logically or physically reserves some fraction of its total buffering space for each flow controlled connection. Let θ be this buffering space reserved for each flow controlled connection and expressed as a number of cells. On each link, the receiving end keeps track of the count of forwarded cells for each connection whereas the transmitting end keeps track of the count of transmitted cells for each connection. Each time the receiver proceeds to the forwarding of Δ cells belonging to a certain connection, it sends the updated forwarded cell count to the transmitter. When the transmitter receives the forwarded cell count, it knows that the receiver can accomodate the transmission of $(\theta - transmitted cell count + forwarded cell count)$ additionnal cells belonging to the particular connection for which the credit transfer occurred. As the transmitter sends down the cells, it updates the credit balance for the particular virtual connection (VC) accordingly. The process starts all over again when the receiving end has forwarded an additional Δ cells. If we define N as the number of equal bandwidth connections multiplexed on the link and RTT as the round-trip time between transmitter and receiver expressed in terms of cell transmission time units, the maximum average bandwidth that each connection can achieve expressed as a fraction of 1 is [27]

í.

$$Bandwidth_{\max.avcrage} = \frac{\theta}{RTT + \Delta N}$$
(2.7)

Therefore, it can be seen that for a given connection to use full bandwidth of the link at burst time given RTT and N fixed, the reserved buffering space θ and the frequency of credit transfer from the receiver to the transmitter (1/ Δ) will have to be high. This observation implies that effective credit based congestion control mechanisms may necessitate a high complexity of implementation. The per-VC queueing required to implement hop-by-hop per-VC based credit scheme significantly affects the complexity of the switching node architecture as cells from all live VCs have to be accessible in a random fashion. This accessibility can be provided by using a linear

linked list in RAM for each live VC. An interesting consequence of the per-VC queueing required is that a suitably fair service policy as weighted round-robin would allow an excellent control of the per-VC cell latency. Nevertheless, if a lessening of implementation complexity is required, the credits and buffering space could be managed on a group of connections basis or on a traffic class basis rather than on an individual connection basis.

Rated based reactive control schemes regulate the traffic on a connection basis at the source end only. *Forward explicit congestion notification* (FECN) [39] is such a method in which the switching nodes of the network are able to monitor their congestion state. In case of critical congestion on a link, they tag the header of the cells affected. When these tagged cells reach their destination, they trigger the transmission of congestion warning messages back to their respective sources. In a variation of this method called *backward explicit congestion notification* (BECN) [39], the congestion of a switching node results in the immediate transmission of congestion warning cells from the point of congestion to the sources of the affected connections. With this variation, the feedback information is obtained faster but it comes at the expense of more intelligent nodes. In order fo these FECN and BECN methods to be effective, the switching should be able to recognize link congestion and further, which connections on a link are responsible for the congestion.

While ATM technology has not matured to the point where congestion control mechanisms could be standardized precisely, it is foreseen that rate and credit schemes may have to coexist [28]. The reason for this is that the credit schemes are very effective on short distances (LAN environments) and become outrageously inefficient over longer distances as pointed out by Equation 2.7, thereby forcing the use of simpler rate control schemes. There is an obvious trade-off between the simplicity of a congestion control mechanism and its effectiveness.

2.6 Signaling

Signaling can be defined as the set of functions allowing the exchange of operation and management information between the switching nodes and the users. Signaling comes in two flavors, that is *user-network interface signaling* for service establishment-initiation and network node interface signaling for exchange of callhandling information between switching nodes. The definition of a signaling system or protocol consists of the description of the format of the various signaling messages (e.g., connect request, disconnect request, call processing) as well as the description of a means of transportation of these messages across the network. In the context of ATM technology, the first phase of the user-network interface signaling messages format is presented in the ATM User-Network Interface specification V3.0 [39] and has been recently standardized under the name ITU-T Q.2931. Signaling messages are submitted to and received from the network through a reserved virtual channel called SiVC and characterized by VPI=0 and VCI=5 [29]. A special ATM adaptation layer protocol dedicated to signaling messages and called SAAL is responsible for the conversion required between the signaling messages and the signaling cells carried on the SiVC. SAAL resides in the control plane of the reference model (Figure 2.1) and provides reliable delivery of 0.2931 signaling messages. Further details concerning SAAL arc available in ITU standards documents Q.2130, Q.2110 and I.363.

There is a wide expertise in the field of signaling systems because they are also extensively used in synchronous transfer mode networks and in other packet networks. The difficulty with signaling in ATM arises from the fact that B-ISDN defines many services involving multiple connections and parties per call. This makes current signaling systems rather unsuitable [30] and explains why standards in this domain are still in development. This weakness of standards lead early switching equipment manufacturers to use semi-proprietary signaling systems, on the premise that multiple signaling systems can coexist in ATM [31].



2.7 SONET and SDH physical layers

Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) are similar framing and multiplexing standards originating from Bellcore and CCITT respectively. Fibre optics becoming the medium of choice in high speed digital networks and the proliferation of proprietary interfaces drove the need for such standards. SONET being the north-american standard, its features and its relation to ATM are presented in the following. Differences between SONET and SDH are only minor and well documented [32].

2.7.1 STS-1 frame structure

The basic building block and first level of the SONET hierarchy is called Synchronous Transport Signal Level-1 (STS-1). The STS-1 frame is drawn in Table 2.2 as an array of bytes having 9 rows and 90 columns. The frame is transmitted one row after the other and from left to right at a line rate of 51.84 Mbs resulting in a frame duration equal to 150µs. The STS-1 frame includes overhead bytes for functions such as framing indication and error monitoring. These overhead bytes are partitioned into three distinct groups so that path, line and section levels of a connection can be controlled and monitored independently. Path overheads regard the management of a connection at the end-to-end level. Line overheads manage the segments of a connection that are between pairs of transport nodes and section overheads manage the segments of a connection that are between pairs of regenerators or regenerator and transport node. This partitioning of the overhead resources of SONET frames allows for an easier fault localization.

SONET receivers can recover the timing of the incoming signal reliably because frames are transmitted synchronously (that is, without gaps between them) and they are scrambled prior to transmission using a polynomial generator, in order to eliminate long
strings of ones and zeros that could unlock receivers phase-locked loop circuitry. Overhead bytes A1 and A2 are constant unscrambled bytes that allow the receivers to locate the beginning of each incoming frame.

| | | | | Synchronous Payload Envelope | | | |
|-------------------------------------|-----------------|------------------------|-------------------------|------------------------------|---------------------|------------|------------------------|
| | Tra | nsport Overl | iead | Path Overhead | Pay | load Capac | iy. |
| | Framing A1 | Framing A2 | STS-1 ID C1 | Trace J1 | Payload Byte 1 | ••• | Payload Byte 86 |
| Section Overhead | BIP-8 Bl | Orderwire El | User F1 | BIP-8 B3 | Payload Byte 87 | ••• | Payload Byte 172 |
| | Data Com D1 | Data Com D2 | Data Com D3 | Signal Label C2 | Payload Byte 173 | ••• | Payload Byte 258 |
| | Pointer H1 | Pointer H2 | Pointer Action H3 | Path Status G1 | Payload Byte 259 | ••• | Payload Byte 344 |
| | BIP-8 B2 | APS K1 | APS K2 | User Channel F2 | Payload Byte 345 | ••• | Payload Byte 430 |
| Line Overhead | Data Com D4 | Data Com D5 | Data Com D6 | Indicator H4 | Payload Byte 431 | ••• | Payload Byte 516 |
| | Data Com D7 | Data Com D8 | Data Com D9 | Growth Z3 | Payload Byte 517 | ••• | Payload Byte 602 |
| | Data Com D10 | Data Com D11 | Data Com D12 | Growth Z4 | Payload Byte 603 | ••• | Payload Byte 688 |
| a (1997) 1997) 1997) 1997) | Growth Zl | Growth / FEBE Z2 | Orderwire E2 | Growth Z5 | Payload Byte 689 | ••• | Payload Byte 774 |

Table 2.2 SONET STS-1 frame structure

2.7.2 Multiplexing

STS-1 signals originating from various sources can by carried on the same fibre in the form of a higher level SONET signal. For instance, N individual STS-1 signals can be byte interleaved to result in a STS-N signal whose line rate is N times faster that the fundamental 51.84Mbs associated with STS-1. Also, a single signal whose bandwidth exceeds the capacity of STS-1 can be carried using a concatenation of N STS-1 signals resulting in a STS-Nc signal. A STS-Nc signal differs from a STS-N signal in that most overhead bytes of its STS-1 underlying signals (except the first STS-1) are not processed.

One of the most revolutionary features of SONET and SDH is the way they handle the multiplexing and demultiplexing of signals [33,34,35,36]. More specifically, the way they perform the correction of the frequency and phase mismatches of plesiochronous⁺ signals is very efficient. Among conventional solutions for this frequency justification of signals, one consists in bit-interleaving the input signals into a resulting multiplexed framed signal. The frequency mismatch is taken care by positive bit stuffing in fixed locations of the resulting frame. This method leads to a high degree of complexity when multiplexing is applied recursively because each multiplexing injects another framing stage in the hierarchy along with its stuffing bits. Then, the extraction of one of the low level signals results in the complex unframing and destuffing process of all previous levels of multiplexing.

Another conventional multiplexing method consists in mapping signals into fixed locations of the resulting multiplexed framed signal. In this method, the frequency justification between signals is absorbed by frame skipping, which is made possible through the use for each input signal of a buffer whose length is given by Equation 2.8.

⁺ Signals whose frequencies have the same nominal value and whose frequency deviations are constrained within specified bounds.



$Buffer Length = \frac{multiplexer output frame length}{multiplexer input number}$ (2.8)

This method is attractive because the absence of bit stuffing results in an easy demultiplexing. Nevertheless, the buffers required increase transmission delays and the overall hardware resources.

SONET and SDH combine the best features of the two previous methods. The need for input buffers at each input of a multiplexer is eliminated by allowing the payload of STS-1, namely the Synchronous Payload Envelope (SPE) to float inside the frame. The precise location of the starting point of the SPE in each frame (J1 overhead byte) is indicated by special pointer overhead bytes H1 and H2. Then, positive byte-stuffing in a frame is accomplished by incrementing the SPE pointers H1, H2 and nulling the SPE byte following H3 whereas negative byte-stuffing is accomplished in a frame by decrementing the SPE pointers and including overhead byte H3 in the SPE.

2.7.3 Transport of ATM cells

Since SONET and SDH have been selected in the context of B-ISDN as the primary physical layer to be used for ATM because of their scalable high performance capabilities, they are obviously well suited for ATM cell transport. SONET and SDH frames even have some built-in mechanisms for ATM cell transport. The overhead byte H4 is defined as a start of cell pointer and therefore could be used in the cell delineation process at the receiver [37]. In practice though, H4 is not used by the receiver and the cell delineation rather proceeds from the header error control (HEC) byte of each cell. In order for the receiver to localize the starting point of a cell, it continuously computes the modulo-2 division of latest 4 SPE bytes received and shifted left eight bits, by the generator polynomial used for the HEC byte generation, namely $g(x) = 1+X^1+X^2+X^8$. When the remainder obtained from the division matches the fifth byte received, the receiver can infer the starting point of a cell within a certain interval of confidence. For

more accuracy then, receivers usually implement an algorithm through which cell synchronization is only inferred after a minimum number of matching remainders separated by the reception of 53 SPE bytes are obtained.

The transmitter places the cells contiguously in the SPE and the pointers H1-H2 are simply left to some constant values. Because the cell delineation is done using the HEC byte of the cells, upon system reset, the very first cell transmitted can be placed anywhere inside the SPE. Then later, when there is no complete cell to transmit, stuffing ATM cells are i ` to the SPE instead. These special idle cells have a special reserved header allowing the receiver to recognize them and discard them. Without these idle cells insuring that each SPE is packed with cells, the receiver could not maintain its cell synchronization and therefore could lose cells.

Chapter 3 Test system requirements

This chapter depicts and defines the role of testing in ATM. It partitions the testing activity into embedded mechanisms responsible for run-time performance monitoring of cell transfers in the network and individual network element testing using dedicated test equipment. Furthermore, the network element testing parameters are extracted and classified as functional parameters and performance parameters. The functional structure of the proposed ATM switching node test system is derived.

3.1 Definition of testing in ATM

In the context of ATM technology, system level testing tasks are usually partitioned in two classes. At the highest level, the term *network wide testing* refers to the set of operation and maintenance functions that are embedded in the various elements constituting the network. Then, at a lower level, *ATM element testing* is defined as the conformance assessment of the individual ATM elements such as the switching nodes, add/drop multiplexers and terminals. Network wide testing and ATM element testing are not standardized keywords but they are often seen in the literature.

Whereas network wide testing is suitable for providing fault management and rudimentary performance monitoring to the network operator or user, ATM element testing is necessary in order for network operators or equipment manufacturers to thoroughly check the functionality and performance of equipment against specified values. This conformance testing is especially important for newly manufactured equipments or newly installed equipments.

3.2 Network wide versus Network element testing

Accordingly with the ATM testing definitions above, the task of assessing the functionality and performance of an ATM network may be seen as a two level process. The typical ATM network consisting primarily of an interconnection of switches, add/drop multiplexers and terminals spanning an arbitrarily large geographical area, the first level of testing concerns the individual evaluation of these elements using high functionality custom test systems. In particular, switching nodes being the elements the most likely to affect the quality of service of the network, their correct behaviour has to be confirmed with great care. The second level of testing goes beyond individual elements of the networks as it rather regards the network as a whole entity.

3.2.1 Network wide testing

Network wide testing is accomplished through mechanisms that are part of the network, or more precisely, standardized mechanisms that are embedded into the various elements of the network. These mechanisms provide some level of run-time performance monitoring, fault management and facility testing. They are part of the layer management plane of the B-ISDN Protocol Reference Model introduced in chapter 2 and they have been standardized in ITU I.610 [38] for the specific case of the user-network interface. They can be partitioned into three classes, depending if they concern AAL, ATM or Physical layer.

The Operation and Maintenance (OAM) functions associated with SONET STS-3c

physical layer of the user-network interface are illustrated in Table 3.1 as taken from [39]. The right most column identifies the SONET frame overhead bytes used by each function. When necessary, a specific field inside a byte is indicated in parentheses. The convention used labels the bits of a SONET byte from 1 to 8, the first bit transmitted on the line being 1. Similar OAM functions also exist for other ATM physical layers like DS3, 100 Mbs multimode fibre, unshielded twisted pair voice grade (UTP-3) and data grade (UTP-5) cables.

| | Functions | SONET STS-3c Overhead Bytes | |
|------------------|--|--------------------------------|--|
| Performance | Cell Header error monitoring | Error Type | |
| Monitoring | Line Error Monitoring | B2(1-24), Z2(18-24) | |
| | Path Error Monitoring | B3(1-8), G1(1-4) | |
| 1 | Section Error Monitoring | B1(1-8) | |
| Fault | STS Path Alarm Indication Signal (AIS) | Н1, Н2, Н3 | |
| Management | STS Path Remote Defect Indicator (RDI) | G1(5) | |
| | Loss of cell delineation / Path RDI | G1(5) | |
| | Line AIS and Line RDI | K2(6-8) | |
| Facility Testing | Path connectivity verification trace | J1 | |

Table 3.1 Physical layer OAM functions

As shown in Table 3.1, the functions are grouped into three categories called *performance monitoring, fault management* and *facility testing*. Performance monitoring functions detect coding violations at the section, line and path levels using SONET overhead parity bytes B1, B2 and B3 respectively. These parity bytes result from an even bit interleaved parity (BIP) check that is conducted at the transmitter for each transmitted frame and at the three levels of section, line and path. The receiver applies the same parity check rules and compares the results obtained with the BIP bytes received for each frame. Each bitwise difference between a computed BIP and a

received BIP indicates that at least one byte was corrupted during transmission. The receiver is responsible for counting these block errors in each frame and for conveying back the count to the upstream equipment through overhead bytes Z2(18-24) for line level and G1(1-4) for path level. These error count back propagating overhead bytes are called line Far End Block Error (FEBE) and path FEBE respectively.

Fault management functions are intended to detect, isolate and correct failure conditions in the network. They are triggered by receptions of alarm indication signals (AIS), remote defect indicators (RDI) and incoming signal failures such as loss of signal, loss of frame and loss of pointer. AIS is used to signal an upstream failure to downstream nodes through overheads H1, H2, H3 and K2(6-8). RDI is used to signal a downstream failure to upstream nodes through overheads G1(5) and K2(6-8).

The Layer Management plane of the B-ISDN protocol reference model also includes OAM functions for the ATM layer of the user-network interface. These functions regard fault management through alarm surveillance and connectivity verification [40]. By opposition to physical layer OAM functions that use SONET overhead to communicate, ATM layer OAM functions exchange information through special ATM cells called OAM cells. There are two types of OAM cells, namely F4 for virtual path connections and F5 for virtual circuit connection. These cells have the same structure as ordinary ATM cells but their payload is partitioned into specific fields as illustrated in Figure 3.1.

F4 OAM Cells



Figure 3.1 OAM cells types

Alarm surveillance involves detection, generation and propagation of virtual path connection and virtual circuit connection failure information. The failure indication signals are of two types, namely Alarm Indication Signal (AIS) and Far End Receive Failure (FERF). The signal type is encoded in the function type field of the OAM F4 and F5 cells, as indicated in Figure 3.1. AIS cells are used to warn downstream nodes of an upstream failure. When an AIS cell reaches the public user-network interface endpoint, a FERF signal cell is injected backwards in the network to warn upstream nodes of a downstream fault. In definitive, AIS and FERF cells are the ATM layer counterparts of the AIS and RDI physical layer OAM signals. The third type of ATM layer OAM cell is qualified as *loopback cell* and allows to perform connectivity verification at the virtual path or virtual circuit level. Through a reserved field in the loopback OAM cell, the loopback location along a virual connection can be specified.

All operation and management mechanisms presented in this section are coarsely

standardized by various bodies of the ATM field and they mainly concern the connectivity verification at physical and ATM layers. Alternatively, more extensive OAM mechanisms are proposed [41,42] so that in-service cell transfer performance can be monitored. There is an obvious trade-off between the complexity of the OAM functions implemented and the simplicity of the resulting network operations.

3.2.2 Network element testing

Network element testing refers specifically to the individual conformance assessment of the various elements composing the network such as add/drop multiplexers, switching nodes and terminals. This verification is accomplished through the use of high capability probing custom equipment that is not necessarily part of the network itself. In the context of this work, we are primarily interested in switching node test systems, since these switching nodes have the most critical impact on the overall network performance.

Because of the high bandwidth involved, the diversity of traffic classes it handles and the various physical media used, ATM network testing brings a whole new challenge. While conventional approaches to protocol testing implement traffic analysis with software, ATM technology rather requires traffic monitoring at line speed with only the erroneous traffic being presented to the network operator [43]. In order for such a real-time traffic monitoring to be possible, custom hardware circuits like pattern matchers and decision engines the can check protocol rules must be used. In addition to these high-performance issues, the monitoring equipment should also allow some degree of programmability and modularity to suit the variety of traffic classes and physical media seen in ATM.

3.3 Partitioning of test parameters

The task of conformance testing of ATM switching nodes can be described as a series of test parameters that have to be measured and checked against given specifications. All test parameters can be partitioned into two classes, namely the functional parameters and the performance parameters. Functional parameters regard the various functionalities of the switching nodes as described in the B-ISDN protocol reference model introduced in chapter 2. Performance parameters rather concern the quality of cell transfer throughout the switching nodes. In the language of testing science, the ease or difficulty inherent to a test parameter measurement will be determined by the controllability and the observability of that particular parameter.

3.3.1 Functional test parameters

As far as functionality testing is concerned, considering that the ATM telecommunication network has already been logically partitioned into functions and services, testing each function and service individually appears as being the most natural way to assert the proper functionality of the system. Table 3.2 presents a classification of the most important functional test parameters as gathered from [44].

| Classification | Function | |
|------------------------------|---|--|
| Physical layer | Interfacing to transmission equipment Cell delineation and synchronization, header verification | |
| ATM layer | Cell switching functions, including header translation Traffic concentration and segregation Policing functions Buffer management | |
| Control plane | Signaling protocol handling at the access and network interfaces Call and connection related control functions Resource management features | |
| Operation and maintenance | Traffic management, billing and fault management functions Interworking related functions | |

Table 3.2 Functional test parameters

Some of the functional test parameters are straightforward to measure since they are well defined and usually do not depend on the load of the system. Furthermore, parameters such as physical layer parameters, header translation and routing are easily controllable and observable. Other parameters such as signaling and usage parameter control are more problematic because they concern the evaluation of the execution of complex real-time algorithms inside the switching nodes. Again, it is the controllability and observability that will determine their ease of evaluation. Complex switching nodes are therefore required to come with special interface providing some level of internal information and control.

3.3.2 Performance test parameters

Performance testing is related to queueing effects in ATM networks. Because ATM is based on statistical multiplexing, queue transients and overflows occurring under network congestion can lead to cell losses, cell delays and cell delay variations violating the quality of service contracted with the users. ITU recommendation I.356 defines many parameters to evaluate the performance of the network and in the case of the ATM layer, they are cell delay, cell delay variation, cell loss ratio, cell misinsertion rate and errored cells ratio. The test equipment should therefore be able to monitor these ATM layer performance parameters. This monitoring has to be executed while the network undergoes a normal load, in other words, while the switching nodes are exposed to streams of ATM cells having a certain mean and peak emission rate as well as a certain burstiness. This load must be obtained artificially through the use of traffic generators if the test cannot be performed in a real environment. The traffic generation is an intricate task since it should be as realistic as possible but few experiences exist on real traffic sources. Table 3.3 presents performance parameters of a B-ISDN switch as gathered from [44].

| Classification | Function | | |
|--|---|--|--|
| Cell Level | Cell loss rate due to buffer overflow | | |
| | Cell loss rate due to Usage Parameter Control (policing) function | | |
| | Average cell delay / delay jitter | | |
| Call Level | Call blocking probability | | |
| | Call setup delay | | |
| | Call handling capability of the switch control processor | | |
| Control plane | Signaling protocol handling at the access and network interfaces | | |
| | Call and connection related control functions | | |
| | Resource management features | | |
| Operation and | Traffic management, billing and fault management functions | | |
| maintenance Interworking related functions | | | |

Table 3.3 Performance test parameters

ATM technology and standards have not matured to the point where it would be possible to provide an exhaustive list of the test parameters, test parameter objective values and test methodologies. Nevertheless, objective values for the cell level performance parameters have been specified by the industry through the ATM Forum [39]. These specifications are provided for the four service categories of constant bit rate (CBR), variable bit rate (VBR), available bit rate (ABR) and unspecified bit rate (UBR). Given that the cell level performance requirements can be obtained more naturally on an end-to-end connection basis, interpolation has been used in order to express them on a per-switch basis. This way, the manufacturers can have a clearer performance target for the switching architectures they devise. These per-switch performance objective values are presented in Table 3.4.

| Category | Cell Loss Rate | Cell Transfer Délay | Cell Delay Variation |
|----------|-----------------------|---------------------|----------------------|
| CBR | 1.7×10^{-10} | 150 μs | 250 μs |
| VBR | 1.0×10^{-7} | 150 μs | 250 μs |
| ABR | 1.0×10^{-7} | none | none |
| UBR | none | none | none |

Table 3.4 Cell level performance parameter objective values

3.4 Generator-Analyzer structure

The functional and performance parameters needing measurement and monitoring being expressed, the switching node test system required to achieve the testing goals can be sketched. As it turns out, the test system required has the same basic stimulusanalyzer structure as other test systems used for analog and digital circuits. It primarily consists of a cell generator and a cell analyzer that are connected to the device under test. The basic ATM switching node test system structure that is usually found in the ATM literature is shown in Figure 3.2.

The cell generator creates a flow of test cells that are injected in the switching

node under test whereas the cell analyzer proceeds to parameter extraction from the incoming flow of test cells. The figure also shows a special connection between the test system and the control complex of the switch. At the time being, this connection is not defined in standards but it is meant to increase the controllability and observability of the switching node, thereby simplifying its conformance assessment.



Switching Node Test System

Figure 3.2 Switching node test system structure

The monitoring of the aforementioned test parameters should be carried out while the switching equipment undergoes various realistic levels of load. The first reason motivating this is that it is highly interesting to analyze the variation of parameters as a function of the network load. Second, some perfomance parameters like cell losses and cell delay variations only reach critical values under some level of network congestion. Thus, the cell generator has a double purpose. First, it has to create *foreground cells*, in other words, cells that are recognized and used by the cell analyzer for the various parameter evaluations. Second, it has to create *background cells* whose only purpose is to simulate the background network load under which the various tests must be carried out. If the tests are carried out in a live environment, this background load generation may not be required anymore.

In the specific test system structure presented, the traffic generator and analyzer are side by side physically such that they have access to a common synchronisation clock. This structure can be qualified as single box system. Transmission delay of cells through a single switching node can simply be monitored by time stamping the foreground cells using the system clock since both the cell generator and the cell analyzer have access to it. For the case where delay measurement between two sites is needed, the single box system can still be used at one site while the test cells are looped back at the other site. Alternatively, the structure of the test system could be such that the cell generator and cell analyzer be located in two different sites. Then, delay measurements would be complicated by the fact that precise clock synchronisation between remote sites remains a problem [45].

In summary, the typical ATM switching node test system consists of a traffic generator and a traffic analyzer whose physical interfaces correspond to the ones of the equipment under test. Froper traffic generation and analysis can lead to a reliable evaluation of the various functional and performance parameters constituting the conformance assessment of an ATM switching node.

Chapter 4 ATM traffic modelling

This chapter presents the issues associated with the cell generation process required by the test system. A taxonomy of traffic source modelling techniques is presented as well as some interesting modelling approaches found in the literature. Modelling concerns for voice and video sources are addressed. An impressive traffic generation system engineered by the Research & Development in Advanced Communications Technologies in Europe (RACE) called PARASOL is also presented. Finally, hardware synthesis of stochastic processes is studied as it is required by the test system proposed.

4.1 A taxonomy of traffic source modelling

As stated earlier, the monitoring of most test parameters should be carried out while the switching equipment undergoes some realistic load. It is needless to say that the accuracy and validity of the parameter measurements strongly depends upon the quality of the traffic generation achieved by the test equipment. Unfortunately, whereas the performance parameters themselves are well known, the identification and standardization of a reference load associated with their measurement has never been done. Therefore, in the absence of such standards and reference documents, we are forced to proceed from a set of assumptions on ATM traffic characteristics. From the test equipment point of view, a good quality traffic generation consists of a traffic model that resembles as closely as possible the traffic that would be present if the switching node were used in its intended environment. More precisely, the synthetic traffic should be composed of a comparable number of sources, comparable traffic classes and comparable traffic class ratio. The difficulty arising here is that poor documentation or knowledge is available concerning real ATM traffic. Most traffic classes are well known (statistically speaking) at the source when they enter the network, however the queueing, multiplexing, prioritized processing, congestion and flow controls spanning the network tend to reshape the traffic. Given that the switching nodes are effectively exposed to this reshaped traffic, it is precisely this traffic that the test cell generator should reproduce. As reported in [46], there are mainly three source modelling approaches for ATM traffic, namely *memory based, stochastic process based* and *physical sources based*.

4.1.1 Memory based generation

This method simply consists in the recording and playback of a cell sequence. The recording of real traffic can be executed in a switching system in use. The worst drawback of this method is the prohibitive amount of storage medium required to produce a significant length cell stream. In addition, the use of a recorded cell stream prevents any traffic parameter from being further varied and thus is not very flexible.

4.1.2 Stochastic process based generation

This traffic generation method consists in modelling the sources of an ATM link by a single stochastic process whose parameters are adjusted carefully. The use of a single statistical process encompassing all the sources of a link can be seen as a black box approach. In other words, the stochastic process ignores the statistical characteristics of the individual sources and its only use is to create a traffic that resembles the aggregation of all sources of the link. The statistical process should ideally lead to an easy and practical implementation. An important drawback of this method is that given the absence of a direct relationship between the traffic sources modelled and the stor hastic model, the model cannot be easily tuned to reflect a load change. A separate model would then have to be engineered for each load level and mixture desired. It is unclear how easy and how accurately sources could be modelled by a single stochastic process.

4.1.3 Physical sources based generation

In this method, instead of using a single stochastic process to model all sources of a link at once, a distinct stochastic process is used for each traffic source. Then, these individual stochastic processes are grouped to produce the resulting traffic of the link. This model can be seen as a linear combination of the individual source models. In this context, varying the load simply consists of adding, deleting or modifying some of the individual processes. Physical sources based generation usually leads to models that are more cumbersome than stochastic based generation models.

4.2 General modelling concepts

Provided a general ATM network consisting of a multitude of interconnected switching nodes, the traffic reaching each node will have a specific history. For instance, switches located near the boundaries of the network will be exposed to traffic consisting mainly in a direct multiplexing of the physical sources (phones, video-phones, computers). On the other hand, switches located at the core of the network will receive traffic that has been reshaped by surrounding network elements such as the switches and the add/drop multiplexers. This reshaping of traffic occurring in the ATM network should be considered in the present study of traffic modelling since a switching node test system should ideally model this reshaped traffic and not merely a direct multiplexing



of physical sources.

In the study of traffic reshaping produced by multiplexers and switches, the ATM multiplexer appearing in Figure 4.1 constitutes a good basis because in the limit, each output port of an N by N switching node is nothing else than an N-to-1 multiplexer. The output link (*Out*) of the multiplexer is shared statistically among all incoming links (I_0 to I_n). The queue is used to accomodate the statistical nature of the inputs being multiplexed and it should be properly dimensioned (K) to constrain the packet loss probability below the value prescribed by the Quality of Service requested by the multiplexed traffics. A logical partitioning of the queue can allow a prioritized processing of traffic classes using different Quality of Services [47].



Figure 4.1 ATM traffic multiplexing

From the viewpoint of queueing behavior, a buffered multiplexer or an output queueing switch output port can be modelled by a single server with deterministic service time. The input to the queue results from the superposition of all incoming traffic streams. A fundamental theorem in queueing theory, Burke's theorem, states that for M/M/1, M/M/m or M/M/ ∞ systems with arrival rate λ , the departure process is also Poisson with parameter λ . Making use of Burke's theorem to characterize the departure process from the queue of ATM multiplexers would be interesting for its tractability but not very accurate. First, the incoming traffic to the queue results from the merging of many streams that may not be individually suitably represented by a simple Poisson process. Therefore, the merged process itself should not necessarily be considered Poisson. Furthermore, the service time of the server is deterministic, not memoryless.

Theoretically speaking, the most embarrassing characteristic of the departure process from a multiplexer or a switch port is that it is a *non-renewal* process [48]. Given this so-called non-renewal character, the interdeparture times of the cells can no longer be considered independent and identically distributed. As is observed in simulations, the length of successive interdeparture intervals are highly correlated. The Poisson process being memoryless by definition, it cannot represent this interdependence between departure intervals. When facing the need for accurate traffic modelling as is the case in queueing analysis, this correlation of traffic should be included in the models as its influence on queueing behavior is crucial. In the context of the present work, this correlation modelling may also prove necessary in order for the performance parameters of the switches to be evaluated precisely.

Additionally, studies reported in [49] show that the cell arrival processes corresponding to Telnet, FTP and remote login sessions have their burstiness strongly underestimated by the Poisson process modelling. In these studies, the assessment of the quality of modelling by a Poisson process is a two-step procedure. First, a minimum number of samples over which the arrival rate is constant is selected and the interarrival times of the sequence are checked for an exponential distribution using the Anderson-Darling [50] test or the more conventional χ^2 test. Then, the interarrival times are checked for independence using the autocorrelation of the sequence as an indicator.

In practice, it has been shown through various simulations that the departure process from an ATM node tends to be less bursty than the corresponding arrival process. This phenomenon called *traffic smoothing* consequently reduces the mean queue lengths observed in downstream nodes of each connection. Simulations found in [51] report that using the squared coefficient of variation of interarrival times as a measure of traffic burstiness, the smoothing effect of a single node varies between 1% and 4% depending on the source traffic model used and the line utilization factor. The squared coefficient of variation mentioned is defined as

$$\frac{E[(X - \mu_x)^2]}{(E[X])^2}$$
(4.1)

where X denotes the interarrival time and μx its mean value. In the same study, the beneficial decrease in the mean waiting time caused by the smoothing of a single node was estimated to be in the 1%-5% range, again depending on the load and traffic model used.

4.3 Some modelling approaches

Some traffic modelling approaches are introduced in this section. Three methods are presented for modelling the superposition of voice traffics or more generally, any traffic that can be considered being ON/OFF type. These methods are namely the *Single Poisson Process*, the *Markovian Modulated Deterministic Process* (MMDP) and the *Markovian Modulated Poisson Process* (MMPP). Then, the modelling issues concerning specifically video sources are introduced.

4.3.1 Voice traffic

Characteristics of digitized voice signals are well known since the modelling of speech signals has been a continuing research activity for at least fifty years. Figure 4.2 presents the packetization of a voice signal using adaptative differential pulse code modulation (ADPCM). The voice signal itself consists of successive intervals of speech and silence. The average duration of the speech and silence intervals is 350ms and 650ms respectively [52]. In the context of ATM, the packetization process is expected to make use of a *speech activity detector* so that no cells get generated during the silence periods of the source. Therefore, burstiness is introduced in the resulting cell stream in favor of an average bandwidth requirement reduction. During the activity periods, cells

are emitted at a constant rate corresponding to 32kbs in the specific case of ADPCM source coding. From a statistical viewpoint, the packetized voice signal becomes an 'ON/OFF' source whose active-inactive widths have mean values $1/\mu$ and $1/\lambda$ respectively, as indicated on Figure 4.2. For the sake of simplicity, it is generally assumed that the successive talkspurt and silence periods constitute an alternating renewal process, e.g. the width of each type of interval is an independent random variable. These ON/OFF interval widths can be assumed to be independent and geometrically or exponentially distributed, as it has been shown to be consistent with measurements [52].



Figure 4.2 Voice packetization process

For the reasons mentioned above, the statistical modelling of a single packetized voice signal is reasonably simple. The situation is quite different when it comes to characterizing statistically the superposition of many packetized voice streams into the queue of an ATM multiplexer or switching node. In most cases, complexity precludes any kind of exact analysis so assumptions have to be made in order to obtain a traffic multiplexing model that is tractable.

One trivial way to model the multiplexing is to rely on the fact that the

probability density function (PDF) or probability mass function (PMF) of the summation of independent random variables is equal to the convolution of the individual PDFs or PMFs. If this method leads to an exact distribution for the resulting superposition of traffic streams, it usually does not provide any tractable mathematical model for the resulting process [53]. Also, the convolution operation being irreversible forces the model to start from scratch whenever an input stream is added to or removed from the multiplexed traffic stream. The following will present three different approaches to composite traffic modelling of ON/OFF sources. These approaches are namely the *Single Poisson Process*, the *Markovian Modulated Deterministic Process* (MMDP) and the *Markovian Modulated Poisson Process* (MMPP).

Modelling the superposition of multiple voice or ON/OFF type packet streams can be done in a very attractive manner if it is assumed initially that each voice packet stream is a Poisson process of parameter

$$\frac{\Delta \lambda}{(\lambda + \mu)} \quad cells/\sec$$
 (4.2)

where Δ , λ and μ have the meaning introduced earlier in Figure 4.2. The Poisson process being additive, the resulting process is simply Poisson with a parameter equal to the summation of the individual parameters. It is known that a single packetized voice signal gets poorly represented by a Poisson process, however, when a large number of such independent voice signals are approximated by a Poisson process for the study of mean waiting time in the queue of an ATM multiplexer, the results are reasonably accurate when the utilized multiplexer bandwidth remains below 0.7 [54]. It must be noted here that the fact that the single Poisson process being able to predict the queue waiting time under certain conditions is no guarantee that it is a good model for the superposition of multiple traffic sources.

The Markovian Modulated Deterministic Process (MMDP) can be used to model the superposition of multiple ON/OFF sources and it has been shown to be simple and reasonably accurate [55,56]. The MMDP model for (m-1) homogeneous sources consists of the following parameters :

- X(t) : Finite, irreducible, continuous-time Markov process with state space S={0,1,...,m-1} representing the number of sources in the active mode, (m-1) being the number of ON/OFF sources modelled.
- $B_j \in \{B_0, B_1, \dots, B_{m-1}\}$: B_j is a constant cell arrival rate associated with state j.
- $\gamma_j \in (\gamma_0, \gamma_1, ..., \gamma_{m-1})$: $1/\gamma_j$ is the mean value of the exponentially distributed sojourn time associated with state j.
 - $P_{[m \times m]}$: The m by m probability transition matrix of the Markov process

The MMDP Markov chain model is illustrated in Figure 4.3. The state transition probabilities appear as arrows and they have not been labelled for more clarity. A constant deterministic traffic B_j is attached to each state of the Markov chain. Each state of the chain has a distinct mean sojourn time $(1/\gamma_j)$ that is exponentially distributed with parameter γ_j . The parameter values of the model (X(t), B, γ , P) can be obtained from the number of sources modelled (m-1) and the traffic characteristics (Δ , λ , μ) of the homogeneous ON/OFF sources modelled using Equations 4.4 to 4.8 with N=1.



Figure 4.3 MMDP Markov chain

A simple expansion of the model [55,56] can lead to an heterogeneous traffic model, or in other words, a model representing

$$\sum_{i=1}^{N} M_i \tag{4.3}$$

sources belonging to N different traffic classes. Mi represents the total number of sources of traffic class i, each traffic class i having its own set of parameters Δ_i , λ_i and μ_i . The N-dimensional state space of the model becomes $\Lambda = \{x = (x_1, x_2, ..., x_i, ..., x_N) | 0 \le x_i \le M_i, i = 1,2,...,N\}$ where x_i is the number of active sources of traffic class i. Physically, each state of the Markov chain now represents a distinct combination of active sources among all sources of all N classes. Again, each state of the chain has a distinct mean sojourn time $(1/\gamma_x)$ that is exponentially distributed with parameter γ_x . To simplify the model, it is assumed to be birth-death type such that each transition will only allow a single source of a single class to switch state. The birth-death assumption makes the state transition matrix P become tridiagonal and therefore easier to handle. All parameters of the heterogeneous MMDP model (X(t), B, γ , P) can be obtained using Equations 4.4 to 4.8. The intuitive meaning associated with Equation 4.5 is that the mean sojourn time in state x $(1/\gamma_x)$ is determined by λ_i when most sources of traffic

class i are inactive and by μ i when most sources of traffic class i are active. The intuitive meaning associated with Equation 4.6 is that the probability of a transition from state x to z, where a traffic class i source switches from inactive to active depends on the contribution of the inactive sources of class i to the mean sojourn time in state x $(1/\gamma_x)$. Similarly, the probability of a transition from state x to z, where a traffic class i source depends on the contribution of the inactive to inactive depends on the contribution of the active to inactive depends on the contribution of the active sources of class i to the mean sojourn time in state x is source sources of class i to the mean sojourn time in state x ($1/\gamma_x$).

$$B_x = \sum_{i=1}^N x_i \Delta_i \tag{4.4}$$

$$\gamma_{x} = \sum_{i=1}^{N} \left[(M_{i} - x_{i}) \lambda_{i} + x_{i} \mu_{i} \right]$$
(4.5)

$$P_{x,z} = \begin{cases} \alpha_x^{(i)} & \text{when } z = x_i^+ & \text{and } i = 1,...,N \\ \beta_x^{(i)} & \text{when } z = x_i^- & \text{and } i = 1,...,N \\ 0 & \text{otherwise} \end{cases}$$
(4.6)

$$\begin{aligned} x_i^+ &= (x_1, x_2, ..., x_i + 1, ..., x_N) \\ x_i^- &= (x_1, x_2, ..., x_i - 1, ..., x_N) \end{aligned}$$
 (4.7).



$$\alpha_{x}^{(i)} = \begin{cases} \frac{(M_{i} - x_{i}) \lambda_{i}}{\gamma_{x}} & x \in \Lambda , i = 1, ..., N \\ 0 & otherwise \end{cases}$$

$$\beta_{x}^{(i)} = \begin{cases} \frac{x_{i} \mu_{i}}{\gamma_{x}} & x \in \Lambda , i = 1, ..., N \\ 0 & otherwise \end{cases}$$

$$(4.8)$$

The Markovian Modulated Poisson Process (MMPP) is another model for the superposition of multiple ON/OFF sources and is used extensively to approximate the superposition of packet arrival processes and the queueing delays in network queues. It is very similar to MMDP in that it is based on a continuous-time Markov chain. The main difference is that the traffic attached to each state of the chain in no longer deterministic but rather random with a specific Poisson distribution for each state of the chain. Also, MMPP models usually come with fewer states than MMDP [57] in order to produce more tractable models. As was similarily the case for MMDP, MMPP is characterized by the parameters :

- X(t) : Finite, irreducible, continuous-time Markov process with state space S = {0,1,...,m-1} representing the active Poisson process.
- $\lambda_j \in \{\lambda_0, \lambda_1, ..., \lambda_{m-1}\}$: λ_j is the Poisson parameter associated with state j.
- $\gamma_j \in (\gamma_0, \gamma_1, ..., \gamma_{m-1})$: $1/\gamma_j$ is the mean value of the exponentially distributed sojourn time associated with state j.
 - $P_{\{m \times m\}}$: The m by m probability transition matrix of the Markov process.

The quality of the approximation obtained from MMPP depends on what statistics

of the superposed processes are used to derive the parameters of the model and on how well these statistics are translated to model parameters. In the following, one such specific methodology for the extraction of the MMPP model parameters from the traffic characteristics is reported [58]. The Markov chain illustrating the process appears in Figure 4.4. The chain only has two states (1,2) and the Poisson traffics are indicated by λ_1 and λ_2 respectively. Γ_1 and Γ_2 represent the rate of switching between the states or equivalently, the inverse of the mean sojourn times in each state. The sojourn time is again considered being exponentially distributed, by virtue of the continuous-time Markov chain definition.



Figure 4.4 MMPP Markov chain

Strictly speaking, the superposition of traffics is known to result in a complex nonrenewal process in which interarrival times are correlated. Nevertheless, the specific approach reported here draws the following statistical characteristics from the multiplexed voice signals using the renewal theory as a first approximation :

- 1. Mean arrival rate
- 2. Variance-to-mean ratio of number of arrivals

- 3. Long term variance-to-mean ratio of the number of arrivals
- 4. Third moment of the number of arrivals

All these quantities are expressed as functions of the individual voice stream parameters, namely peak cell rate during talkspurt Δ , silence mean duration 1/ λ and talkspurt mean duration 1/ μ , as introduced earlier in Figure 4.2. Then, these same four quantities are expressed in term of the parameters of the MMPP model (λ_1 , λ_2 , Γ_1 , Γ_2). By equating both expressions obtained for each of the quantities 1 to 4, the model parameters (λ_1 , λ_2 , Γ_1 , Γ_2) can finally be expressed as functions of the number of sources and the voice traffic parameters Δ , λ and μ in a very attractive way [58].

Despite their resemblance, the two specific models of MMDP and MMPP presented herein constitute quite different approaches. Following the traffic modelling taxonomy introduced in chapter 4, the MMDP can be qualified as a *physical sources based generation* model since every single source directly participates to the model. In contrast, the MMPP presented is rather a *stochastic process based generation* model since individual traffic sources do not participate directly to the model construction. It is rather the statistical characteristics of the combination of the individual sources that participate to the model construction. Through the two examples of MMDP and MMPP presented and as outlined in the model taxonomy earlier, the *physical source based* philosophy has lead to a model whose construction is natural but whose use is computationally expensive, especially for the case of heterogeneous sources. In contrast, the *stochastic process based* philosophy has lead to a much simpler model (fewer states) whose relationship with the underlying individual sources is less apparent.

4.3.2 Video traffic

In the present task of broadband traffic modelling and generation, the characteristics of video sources should be closely analyzed since this type of traffic is

expected to use an important fraction of the total bandwidth of the future B-ISDN. Given the statistical multiplexing at the basis of the asynchronous transfer mode, variable bit rate coding of video is very important as it increases the efficiency of the multiplexing. Variable bit rate video coding keeps the picture quality constant by sending update information at a rate proportional to the rate of change of the picture. Once again, as was the case for voice signals, we are interested in characterizing statistically the interarrival distribution of a single packetized source and then of multiple multiplexed packetized sources.

The difficulty arising with the statistical characterization of single video sources is that the great variety of coding methods used and types of scenes or images transmitted make it a very complex and specialized topic. Coding methods can be classified as intra-frame or inter-frame, depending if the intra-frame or the inter-frame correlation (redundancy) of information is exploited by the coder. Some adaptative coders can also switch between the intra-frame mode and the inter-frame mode, depending on the nature of the current scenes being transmitted. For instance, an action scene with fast movements would be best suited by intra-frame coding since the interframe correlation is smaller in this case. In summary, each operation mode of the coder exhibits a distinct set of statistical characteristics and consequently, the entire process becomes very specific and difficult to describe [59]. Nevertheless, statistical descriptions of video sources and multiplexed video sources exist for certain specific coding methods and scene types [60,61,62].

4.4 PARASOL project source modelling

PARASOL is a program that was established under RACE to define measurement and validation methods for ATM system and to develop prototype equipment. In the context of the PARASOL project [63], the *physical sources based generation* philosophy has been chosen in the development of traffic generation equipment. The source modelling principles that it uses are presented in the following, as they constitute an efficient and practical road toward real-time modelling for traffic generation systems.

Under PARASOL, the traffic on a link is modelled by the combination of a set of stochastic models. A hardware-software module manages the evolution of the stochastic models and translates them in real-time into a corresponding flow of cells. The traffic generated consists of a set of N sources, each source belonging to a specific source type $k \in \{1, ..., K\}$. Each traffic type is modelled by a continuous-time Markov chain in which a specific cell departure pattern is attached to each state. At any time t, each source of each type k has a specific current state i into its type Markov chain. If we define $m_i^{(k)}(t)$ as the number of sources of type k that are in state i of their type Markov chain at time t, the total number of sources modelled by the system is a constant N and can be expressed as

$$N = \sum_{k} \sum_{i} m_{i}^{(k)}(t)^{*}$$
(4.9)

Figure 4.5 shows the typical Markov chain used to model a traffic source of some traffic type k. It shows the various states $i \in \{1, ..., I\}$ with their respective traffic patterns λ_i as well as the state transition probabilities p_{ij} represented by arrows between the states. The main difference between this approach and the MMDP model presented in section 4.2.2 is that the traffic attached to each state of the Markov chain is no longer a constant cell departure rate but rather a special cyclic departure sequence called a *pattern*. In each pattern, a dot represents a cell departure during a cell time-slot whereas the absence of dot represents the absence of cell departure. Depending on the number and distribution of the dots inside a pattern, various traffic characteristics can be simulated. For instance in Figure 4.5, the pattern associated with state i=I-1 simulates a bursty type traffic whereas the one of state i=2 simulates a constant rate traffic.



Figure 4.5 Traffic Type Markov chain

As mentioned previously, the PARASOL approach to traffic modelling of multiple sources of multiple types consists in keeping a distinct current state for each traffic source of each traffic type. In other words, each source continuously contributes to the total traffic with the cell pattern λ_i associated with its current state. The traffic generated by the system consists in the aggregation of the patterns of all sources into a global pattern. When a state change occur, a specific source changes state, thereby changing its contribution to the aggregate traffic. The old contribution pattern from the changing source is then removed from the global pattern and the new contribution pattern is inserted. It is assumed here that the system should be able to remember the exact contribution of every source in the global pattern vector.

The state changes are exponentially distributed with a parameter given by Equation 4.10. In the summation, indexes k and i refer to traffic type and source state respectively. $T_i^{(k)}$ refers to the mean sojourn time of state i of traffic type k and is chosen by the user. $m_i^{(k)}(t)$ is the number of sources of type k that are in state i of their type Markov chain at time t.

$$\sum_{k} \sum_{i} \frac{m_{i}^{(k)}(t)}{T_{i}^{(k)}}$$
(4.10)

When a state change occurs in the system, only one source of one type will be affected. The probability that the combination of source type k and source state i is affected by a state change is given by Equation 4.11. The probability that a specific source of the selected type k and state i is chosen is given by Equation 4.12. Once the source is selected for the current transition, the new state is simply determined using the state transistion probabilities attached to the current state of the source type Markov chain [46].

$$\frac{m_i^{(k)}(t)}{\sum_k \sum_i m_i^{(k)}(t)}$$
(4.11)

$$\frac{1}{m_i^{(k)}(t)}$$
 (4.12)

4.5 Stochastic hardware

.

In the previous sections, methods have been presented in order to model ATM cell streams with various stochastic processes. Care has been taken so that the modelling processes share as many statistical characteristics with real traffics as possible. In the present section, methods are presented to allow implementing arbitrary random processes using digital electronics.

4.5.1 Random number generation

The building block required in order to synthesize to hardware the modelling processes introduced previously is the random number generator. For our purpose, a random number will be defined as the value of a random variable uniformly distributed over the interval [0,1]. In the field of computing machinery or digital electronics, there is no such thing as a random event but this intrinsic determinism can be circumvented to allow producing events that in a local sense, appear non-deterministic or as often qualified, *pseudorandom*.

One of the most common approaches for generating pseudorandom numbers is called the multiplicative congruential method [64] and consists in starting with an initial value X₀ called the seed and then recursively computing X_n using

$$\boldsymbol{x}_n = \left(\boldsymbol{a} \times \boldsymbol{x}_{n-1}\right) modulo \ m \tag{4.13}$$

where a and m are positive integers. The X_n calculated this way is usually referred to as a pseudorandom number and is taken as an approximation of a uniform [0,m-1] random variable. The X_n sequence repeats itself after some finite number less or equal to m. The values a and m should be chosen such that

- For any initial seed, the sequence has the appearance of a uniform [0,m-1] random variable.
- For any initial seed, the sequence generated before repetition should be large.
- The modulo operation should be easily executed by a computer

Practically, choosing m as a large prime number can satisfy most of the conditions [64]. Obviously, the multiplicative congruential method is well adapted for use with general pipelined processors or digital signal processors where powerful arithmetic logic units
are available.

As an alternative to the use of the aforementioned arithmetic method, there exists fast and area efficient digital structures called Linear Feedback Shift Registers (LFSR) that serve the same purpose of pseudorandom number generation [65]. An LFSR is a linear interconnection of memory elements where the most upstream element is being fed by the modulo-2 addition of some of the other elements. The memory element outputs used for the feedback are called the tapping points and depending upon their number and their location, the periodicity of the bit pattern generated at the output of the LFSR will vary. A typical LFSR using 2 tapping points and 5 memory elements is shown in Figure 4.6. The LFSR being a recursive structure, its output at one time can be expressed as a combination of previous outputs and more precisely, as a modulo-2 addition of previous outputs. In the specific case of the LFSR of Figure 4.6, the recursive relation is given by Equation 4.14.



Figure 4.6 Linear Feedback Shift Register

$$a_m = a_{m-3} \oplus a_{m-5} \tag{4.14}$$

When a LFSR of size *n* memory elements is used as a pseudorandom pattern generator, it is desirable to select the tapping points such that the bit pattern produced at the output only repeats itself after the maximum value of 2^{n} -1 clock cycles. Given a



certain LFSR structure, the properties of its output sequence $\{a_m\} = \{a_0, a_1, a_2, ...\}$ where a_i is '1' or '0', depending on the state of the output at time 'i', can be studied arithmetically [65]. As illustrated by Equation 4.16, if the LFSR output sequence $\{a_0, a_1, a_2, ..., a_n, ...\}$ is used to build a power series, this series can then be represented by a generating function G(x) using modulo-2 adders and modulo-2 scalar multipliers.

$$G(x) = a_0 + a_1 x^1 + a_2 x^2 + \dots + a_n x^n + \dots = \sum_{m=0}^{\infty} a_m x^m$$
(4.16)

Using the recursive relation of the LFSR, the generating function can be shown to reduce to a form that is only a function of the initial state $\{a_{-1}, a_{-2}, ..., a_{-n}\}$ and the feedback coefficients $\{c_1, c_2, ..., c_n\}$ of the LFSR. The resulting expression for G(x) is given by Equation 4.17. Each feedback coefficient $c_i \mid i \in \{1, ..., n\}$ is '1' when there is a feedback connection on memory element 'i' and '0' otherwise.

$$G(x) = \frac{\sum_{i=1}^{n} c_{i} x^{i} (a_{-i} x^{-i} + \dots + a_{-1} x^{-1})}{1 - \sum_{i=1}^{n} c_{i} x^{i}}$$
(4.17)

If the initial state of the LFSR is selected as shown by Equation 4.18, then G(x) nicely reduces to Equation 4.19.

$$a_{-1} = a_{-2} = \dots = a_{-(n-1)} = 0$$
, $a_{-n} = 1$ (4.18)

$$G(x) = \frac{c_n}{1 - \sum_{i=1}^n c_i x^i}$$
(4.19)

The denominator of Equations 4.17 and 4.19 is called the characteristic

page 64

polynomial of the sequence $\{a_m\}$ and it solely determines the period of the generated sequence through the following theorem :

Given an LFSR with initial conditions given by Equation 4.18, the LFSR output sequence $\{a_m\}$ is periodic with a period which is the smallest integer k for which the characteristic polynomial evenly divides $1-x^k$ [65].

LFSRs producing a sequence {a_m} with a period equal to the maximum value of 2ⁿ-1 clock cycles are said to have a primitive characteristic polynomial.

4.5.2 Random variable generation

Using a method called *Inverse Transform* [66] it is possible to transform a uniform [0,1] random process into another random process having any arbitrary distribution desired. Figure 4.7 presents the transformation process where a discrete uniform random variable U is transformed into an arbitrary distributed discrete random variable Y.



Figure 4.7 Inverse Transformation Method

The algorithm implemented by the inverse transformation method is presented next. Let the random variable Y produced by the transformation be required to have the same distribution as another discrete random variable X with distribution $P{X=x_j}=p_j$. Then, using the uniform random variable U, the inverse transformation Equation 4.20 produces Y as required.

$$Y = \begin{cases} y_0 & if & U < p_0 \\ y_1 & if & p_0 \le U < p_0 + p_1 \\ \cdots & & & \\ y_i & if & \sum_{j=1}^{i-1} p_j \le U < \sum_{j=1}^{i} p_j \\ \cdots & & & \\ \end{array}$$
(4.20)

For the specific case of the generation of a Poisson random variable X, the various probabilities $P{X=x_i}=p_i$ can be obtained with less computation using the following recursive relations.

$$p_{i+1} = \frac{\lambda}{i+1} p_i , \quad i \ge 0$$

$$p_0 = e^{-\lambda}$$
(4.21)

Then, in the Poisson case, the inverse transform procedure can be algorithmically described by the following five steps :

- STEP 1 : Generate the random number U
- STEP 2 : i=0, p=e^{-λ}, F=p
- STEP 3 : if U < F then X=i and stop
- STEP 4 : $p=\lambda p/(i+1)$, F=F+p, i=i+1
- STEP 5 : Go to step 3

The inverse transform being presented, we are now interested in relating it with a proper hardware structure that will allow fast and efficient generation of a random variable Y having the same distribution as a target random variable X. First, the generation of the uniform random variable U that is at the base of the inverse transform method does not cause any problem since it can be obtained from a simple LFSR whose length and structure are appropriately chosen. The second part of the transformer, namely the process of the generation and the accumulation of the probabilities of the target random process X, brings some important problems in the context of a real-time implementation. The complexity of the generation of probabilities of the discrete target process X depends on the complexity of its probability mass function and on the cardinality of its sample set $\wp(X)$. For example, a Poisson distribution requires few computations since the probabilities can be obtained recursively as shown previously. By contrast, an exponential distribution would require a complex Taylor's expansion. In addition to the complexity of the probabilities is sequential by nature. The average number of generations and accumulations of probabilities of the target distribution random variable X for each generated value of the random variable Y is given by Equation 4.22, where μ_x is the mean of X.

$$\frac{\mu_x - \min(X)}{\max(X) - \min(X)} \mathcal{O}(X) \tag{4.22}$$

From a real-time implementation viewpoint, the use of an ALU to realize all computations of the inverse transform algorithm is not very attractive given the complexity of the ALU itself. Besides, the sequential nature of the algorithm prevents high execution speed. An alternative way to generate the random variable could be to use a look-up table being initialized with values obtained from the inverse transform method. The LFSR simulating the uniform random variable U could simply be used as an index into the table.

Chapter 5 Functional specifications

This chapter proceeds to the functional specifications of the proposed ATM switching node test system. These specifications enumerate and describe the functionalities of the system without entering into the implementation details. The system is partitioned into four different modules composing the virtual hardware system. Each module is responsible for a specific test and becomes an icon in the graphics user interface program controlling the test system hardware.

5.1 Statement of specific system design goals

The youthfullness of ATM technology makes it a field where it is particularily interesting to develop architectures because there is still an important amount of knowledge to be acquired and applied. A lot of freedom is therefore experienced throughout the design process. The development of the present system does not proceed from a rigid set of product specifications as would be the case in an industrial environment. It is rather meant to be a design space exploration targetted at finding affordable system architectures whose functionalities are compatible with most ATM switching node test system needs. These needs still lack a clear specification at the present time, thus the test system architecture sought must be very flexible and include

page 68

a high degree of programmability in order to adapt to further evolution.

5.2 Presentation of system

The switching node test system is implemented following the cell generator-cell analyzer structure presented in chapter 3. Concerning the resources used for the realization of the system, the need for an attractive and easy to use graphics user interface (GUI) was recognized from the start and a personal computer was chosen for providing this interface. The SONET/SDH physical layer of the test system is handled by a pre-production custom device provided by Texas Instruments as part of their university support program. This device called *SONET/ATM processor* [67] provides a transmitter circuit for the formatting of ATM cells into SONET STS-3c or SDH STM-1. It also provides a receiver circuit for the extraction of ATM cells received from SONET STS-3c or SDH STM-1. Finally, the circuitry acting as the heart of the test system, located between the GUI and the SONET/ATM processor consists of a field-programmable gate array. The FPGA technology chosen is static RAM based and therefore dynamically reprogrammable, in order to allow building a system with wide functionality and few hardware resources.

Physically, the test system consists of a control computer and a separate printed circuit board (PCB) referred to as the *test system PCB*. The whole system as well as its interconnection to a switching node under test is illustrated in Figure 5.1.





Figure 5.1 Switching node test system

5.2.1 Control computer

The control computer used in the system is a IBM-PCTM compatible with 80486 processor. It is responsible for the realization of the graphics user interface of the system as well as for executing the control program of the test system PCB. The graphics user interface was designed as a Microsoft WindowsTM application, programmed in ANSI C and compiled with Borland 3.1 C++ compilerTM for Windows. Through the compiler used, Windows operating system provides easy access to a rich set of windows, dialog boxes and graphics capabilities [68].

Communications between the PCB and the computer are achieved through the

computer parallel port [69]. This port is standard on all IBM PC compatible computers and consists of :

- 5 status lines (TTL inputs from computer viewpoint)
- 4 control lines (TTL bidirectionnal)
- 8 data lines (TTL outputs from computer viewpoint)

Each of these three sets of communication lines has a corresponding entry in the microprocessor I/O addressing space. The aggregate communication bandwidth provided by the parallel port depends strongly on the length of the cable used and is practically independent of the processor speed. Quantitively, for a ten foot cable, the maximum bandwidth is around 640 kbs

5.2.2 Test system board

All integrated circuits used in the test system are housed on a VME 6U prototyping PCB. The PCB has a power supply plane on one side and a ground plane on the other side. It is populated with Speedwire⁺ pins allowing easy to implement low noise connections from chip to chip. The various circuits found on test system PCB are described in Table 5.1.

⁺ Speedwire is a trade mark from Vero Electronics

| Part Number./Manufacturer | Generic Name | Function |
|---|--|--|
| TDC1500APCM Texas Instrument | SONET / ATM Transmitter Receiver | Mapping of ATM cells into SONET STS-3c and extraction of cells from SONET STS-3c. |
| XC4010pg191-6 Xilinx | Field Programmable Gate Array | Cells production, processing and transfer between PCB memory and ATM/SONET processor. |
| IDT6168SA Integrated Device Technology | Static RAM 8 modules 4 x 4 Kbits (15 ns access time) | Storage of transmitted and received packets. Storage of arrival times and cells identifiers. |

| Table | 5.1 | Test | system | PCB | devices |
|-------|-----|------|--------|-----|---------|
|-------|-----|------|--------|-----|---------|

5.2.3 Virtual Hardware System Concept

In the context of the design space exploration of ATM switching node test system architectures, the easy reprogrammability provided by the static RAM based FPGA technology selected is found to be an interesting (if not required) asset. Even though this reprogrammability comes at the expense of a speed performance decrease relatively to one-time programmable devices, financial resources can dictate its use. Beyond these prototyping stage concerns, SRAM based FPGA technology was chosen primarily so that the system could be built as a Virtual Hardware System (VHS) [70,71]. A VHS can be defined as a system whose hardware circuitry metamorphoses dynamically according to its needs. It is sometimes qualified as silicon multi-tasking system, hardware subroutine system or hardware multiplexing system. In the same way that software subroutines are loaded and executed in response to specific conditions or external events, a reconfigurable logic array such as a SRAM based FPGA can be reprogrammed at runtime with various circuits as required. This allows the implementation on silicon of hardware subroutines that are individually optimized for various specific situations.

In order to build the test system as a VHS, a precise chart of all the hardware resources required by the system is first setup. Then, all these hardware resources are

partitioned into a certain number of equal complexity subcircuits whose respective executions are mutually exclusive in time. The size of each subcircuit is limited to approximately 85%-90% of the total internal resources of the FPGA used, in order to later insure a successful synthesis and routing of the subcircuits. Obviously, such a virtual hardware concept may have its application limited by the time concurrency and the parallelism of the system being built. Nevertheless, in the case of the switching node test system design, it is found that most parameters involved can be monitored sequentially rather than simultaneously. One reason for proceeding to simultaneous measurements could be the need to correlate their results. But still, the overall system can be partitioned and the virtual hardware concept be used if the related parameters join the same subcircuit. From the viewpoint of the virtual hardware test system user handling the mouse, each test parameter measurement becomes an icon of the GUI and the control computer program manages the run-time reprogramming of the PCB FPGA transparently.

Given the current FPGA technology, virtual hardware produces systems whose performance cannot rival with their semi or full custom integrated circuit counterpart. Nervertheless, for cases where a system requires much flexibility, the simplicity of VHS in terms of hardware resources and PCB area can make them attractive and very affordable alternatives to conventional hardware systems. Half-way between the high performance of ASICs and the low cost of virtual hardware lie solutions using general pipelined processors or digital signal processors. These solutions also offer a lot of flexibility and they have been shown to outperform FPGA based VHS in computation intensive applications [70].

5.3 System features

The test system designed supports a subset of the test parameters introduced in chapter 3. The selected subset includes among others the parameters concerning the Quality of Service in ATM. These parameters, as stipulated in [72] and reported in Table 5.2 are of two types, namely cell errors and cell transfer delay.

| Cell Error Parameters | Cell;Transfer Delay Parameters |
|------------------------|--------------------------------|
| cell error ratio | mean cell transfer delay |
| cell loss ratio | 1-point cell delay variation |
| cell misinsertion rate | 2-point cell delay variation |

Table 5.2 Quality of Service parameters in ATM

These parameters require to be monitored on a virtual connection basis because Quality of Service itself is a concept attached to virtual connections. Cell misinsertion rate refers to the rate of incoming cells on a virtual connection that do not belong to that connection but that got through because of previous undetected header errors or routing anomalies. The *1-point* cell delay variation parameter concerns the variability in the pattern of cell arrival events with respect to the negotiated peak cell rate of the connection. The 2-point cell delay variation parameter regards the variability of the pattern of cell arrival events with respect to arrival pattern of same virtual connection at an upstream node of the network. Thus, 2-point measurement provides both mean and variance of delay whereas *1-point* provides only variance.

5.4 Virtual hardware system modules

The ATM switching node test system consists in a virtual hardware system composed of four modules. Each module includes a specific FPGA configuration as well as some associated Windows software routines. Table 5.3 presents these four modules as well as a brief description of their respective functionality.

| Module Name | Module Functionality |
|---------------|--|
| TDC1500 | Provides an interface with SONET/ATM processor configuration registers, control and alarm signals |
| File_transfer | Monitored file transfer and capture using PCB memory |
| Cell_error | Measurement of cell error ratio through pseudorandom traffic generation and analysis. |
| Cell_delay | Measurement of cell delay and losses through pseudorandom traffic generation and analysis. |

 Table 5.3
 Virtual hardware system modules

5.4.1 TDC1500

The TDC1500 module is associated with the management of the SONET/ATM processor. This processor is a BiCMOS device implementing the physical layer of the test system. It includes a transmit queue in which the ATM cells to transmit can be written through an 8-bit wide interface. Internally, the content of the transmit queue is read one byte at a time, converted to a serial stream, formatted into SONET STS-3c or SDH STM-1 frame and sent to the pseudo-ECL differential serial output of the device at a bit rate of 155.52 Mbs. The timing for the serial transmission is provided by an on-chip clock multiplier that multiplies by eight an external low speed 19.44 MHz clock. The SONET/ATM processor also includes a receive queue from which the received cells can be read out through an 8-bit wide interface. The receive queue is used by the processor to store the cells that are received through the pseudo-ECL differential input of the device. The clock recovery from the incoming STS-3c/STM-1 bipolar serial stream is accomplished by an on-chip analog phased-locked loop circuit whereas STS-3c/STM-1 frame synchronization and ATM cells delineation are performed by conventional CMOS circuitry.

The TDC1500 module provides the user with an interface to control, interrogate and configure the SONET/ATM processor through its nine control pins, its nine alarm indication pins and its 8-bit wide data, 8-bit wide address controller interface. Through the TDC1500 module, the user can set the control pins to any values, obtain the state of the alarm signals and access the various configuration registers for read or write operations.

The control signals of the SONET/ATM processor allow among other things the reset of the circuit, the selection of SONET or SDH mode of operation and the bypass of clock generation or clock recovery. The alarm signals provide information on the incoming SONET/SDH stream to the processor. Thus, they signal such events as loss of signal, loss of frame pointer, loss of frame delineation, loss of cell delineation and loss of ATM bytes in the receive queue. The configuration registers include control registers serving more or less the same purpose as the control pins of the device and interrupt registers signaling alarm conditions. Finally, some of the registers give access to the roll-over counters used for automatic monitoring of section, line and path errors through B1, B2 and B3 overheads respectively.

5.4.2 File_transfer

The File_transfer module allows the user of the system to transmit a binary or text file from the computer system to the SONET/SDH serial output of the test system. The file chosen for transmission is packetized with a cell header value selected by the user and is transmitted using a selectable fraction of the full STS-3c/STM-1 bandwidth. The files transmitted using the File_transfer module can simply contain general information like data, sound or images but can also contain signaling cells. For instance, under a specific ATM LAN system a connection could be requested by sending a file with cells containing the traffic parameters requested for the connection and the addresses of both parties. Alternatively, in the presence of a switching node complying with the Q.2931 user-network interface signaling protocol and its associated SAAL⁺, a

+ See chapter 2

page 76

special binary file could be created for each signaling message supported by the node. Then, these special files could be used by the File_transfer module for the management of the switching node.

File_transfer also allows to capture the cells received through the serial STS-3c/STM-1 input of the test system during a certain interval. The captured cells are then displayed in a GUI window to allow the user to analyze eventual bit error patterns by comparing the transmitted and received cells. In the case where the received cells expected are the same as the ones that were just transmitted, as would be the case if the transmitted cells were looped back through a switching node into the receive port of the test system, the system will track down automatically the bit errors that occurred during the transfer. Otherwise, if the received cells do not come from a transmission loopback but are rather the response to some signaling message previously submitted to the switching node, the error tracking capability is not involved. The analysis of the cells captured in response to the transmission of a signaling message can be used as a test of the signaling protocol support by the switching node.

5.4.3 Cell_error

The Cell_error module is meant to provide statistics on bit errors, cell losses and misinserted cells. For that matter, the FPGA module will act as a cell generator with a selectable header and a selectable cell departure rate. The content of each generated cell will be two-fold. First, in order for the system to be able to notice cell losses, each transmitted cell will have its sixth byte used as a cell identifier. The identifier will be incremented for each transmitted cell. Second, in order for the system to monitor bit errors, the content of each cell transmitted (bytes 7 to 53) will be determined by an 8-bit linear feedback shift register acting as a pseudorandom payload generator.

Upon each cell reception by the test system, the three aforementioned types of error are monitored. First, if the header of the cell received is not equal to the header currently used for the cell generation, the misinserted cell count will be incremented. Then, if the actual cell identifier received does not match the expected cell identifier, the cell losses counter will be incremented. Finally, upon each reception of a cell whose header and identifier are the expected ones, a LFSR producing the same sequence as the transmission LFSR will be triggered such that each payload byte of the cell received can be checked for bit errors.

5.4.4 Cell_delay

The Cell_delay module provides cell transfer delay and cell loss measurement through traffic generation, capture and analysis. Three different traffic streams or traffic sources labelled A, B and C are generated concurrently by the system. A is the foreground source whose cells are labelled and time stamped to allow cell delay and cell loss measurement through the switching node under test. B and C are the background sources whose sole purpose is to create realistic congestion levels in the network during the measurement. The cell header of each traffic source is user-selectable. The foreground source is modelled by the discrete-time interrupted binomial process illustrated in Figure 5.2. This process is a discrete-time equivalent of the continuoustime Interrupted Poisson Process (IPP). The IPP itself is a simple two-state Markov modulated Poisson process (MMPP) in which one of the two states does not generate any traffic.



Figure 5.2 Interrupted binomial process

When the source is in the active state, the cell generation follows a binomial process with success probability Probenit. When it is in the inactive state, it does not generate any cell. The state of the source is updated periodically upon each ATM cell time-slot (2.73µs under STS-3c). Upon a state change, the source will either remain active with probability ProbRA or remain inactive with probability ProbRI. Each probability parameter of the model is selectable before the execution of the test. The two background sources are modelled more simply as a non-interrupted binomial process whose success probability used is the same as the emission probability of the foreground source. Equation 5.1 has been derived for the aggregate bandwidth resulting from the three traffic sources (A,B,C). The aggregate bandwidth is expressed as a fraction of the traffic models.

$$STS-3c_{fraction} = \min\left[\left(2 + \frac{1 - Prob_{RI}}{2 - Prob_{RA} - Prob_{RI}}\right) \times Prob_{Emit}, 1\right]$$
(5.1)

page 79

The duration of the test is expressed as a certain number of foreground cells and is also user-selectable. Upon execution of the test, the three traffic sources are multiplexed through a first-in first-out queue and sent to the serial output of the test system. At the same time, the incoming traffic on the serial input of the test system is filtered and processed in such a way that foreground cells coming back from the switching node under test have their cell label and arrival time recorded into test system PCB SRAM. Other cells are simply discarded. From this recording, cell delay and cell loss measurements can be obtained and correlated.

Chapter 6 Software and hardware specifications

This chapter presents the hardware and software specifications of the test system architecture devised. Each module of the virtual hardware system is described in terms of a FPGA configuration and a set of software routines. The FPGA design methodology and the synthesis tools used are also presented.

As was seen previously, the virtual hardware based test system consists of four software modules and four associated hardware modules of similar complexity that are implemented using SRAM based FPGA technology. The complexity of the four hardware modules can be reduced by putting more functionality in each associated software module. This procedure constitutes a classical case of hardware-software partitioning. However, given the poor bandwidth offered by the physical connection to the computer, no real-time function of traffic analysis or generation could be implemented in software. In order to circumvent these computer I/O limitations, the PCB is populated with fast and wide static RAM memory that can accomodate the high bandwidth real-time needs. The memory on PCB can be used for storing cells to be transmitted, for capturing incoming cells or for recording anomalies observed during a test. Before a test run, the memory content can be filled properly from the computer

system at a rate that is suitable for the connection. Similarly, after a test run the memory content can be retreived and analyzed by the computer system.

6.1 Software

As mentioned, the software associated with the test system does not participate to the real-time fraction of each test run. The software is rather used to control, assist and complement each FPGA design. The following sections present the main software modules, namely File_transfer, Cell_error and Cell_delay. The software associated with the TDC1500 module is not presented because of its simplicity.

6.1.1 File_transfer

The File_transfer software module packetizes a file selected by the user for transmission. The user selects the ATM header for the packetization as well as the constant interdeparture time expressed as an integer number of idle cells inserted between every two effective cells. The window associated with the File_transfer module is presented in Figure 6.1.

Once the selected file has been packetized, downloaded into the test system PCB memory and transmitted, the returning cells are written back to memory and are finally uploaded into the computer. The cells are then presented to the user in the *Receiver Cells* box of the window so that they can be compared with the initial cells appearing in the *Transmitter Cells* box. The *Error Track* controls of the window can be used by the user to automate the forward and backward tracking of errors. In Figure 6.1, the *Cell Content* field of the window shows that an error occurred during the transmission of the byte number 4 of the cell number 18. This particular error is explained by the fact that the HEC byte of each cell is left to zero during the packetization process and is computed and inserted in real-time by the SONET/ATM processor upon the transmission.



Figure 6.1 File_transfer module window

6.1.2 Cell_error

The Cell_error software module allows the user to select the parameters of the traffic generation used for the monitoring of cell errors. Thus, the ATM header as well as the constant cell interdeparture time expressed as a number of idle cells can be selected. The traffic generation can be activated and interrupted with the *start* and *stop* buttons of the window. Once the traffic generation is stopped, ATM byte errors, cell losses and misinserted cells statistics are read from some of the FPGA registers and provided to the user in the respective boxes of the window, as illustrated in Figure 6.2.

| CELL_E | RROR |
|----------------------------------|---|
| Heador, Select 200 0x04030201 | ATM Byte Errors 0 DEC |
| Specing Cells | Cell Losnes O DEC Misinserted Cells |
| Siet Stop | 0 DEC |

Figure 6.2 Cell_error module window

6.1.3 Cell_delay

The Cell_delay software module allows the user to select the parameters for the traffic generation process that is used to evaluate cell delay statistics. As such, the user can select the headers for the foreground source (A) and the backgroud sources (B,C). Also, the parameters of the random process generating the traffic namely, *Probability Remain Active*, *Probability Remain Inactive* and *Probability Emission* can be specified, as well as the duration of the test run expressed as a number of foreground cells. Figure 6.3 presents the GUI window associated with the Cell_delay module.

Once a test run is done, this software module examines how many cells have been received by the FPGA and further reads the test system PCB memory to extract for each received cell, the cell identifier as well as the cell arrival time. All this information is presented to the user in the *Cells Arrival Times* box of the window so that cell delays and losses can be correlated.

| , | CELL_DELAY | ` |
|---|--|---|
| | Cell Arrival Times | |
| Cells to Transmit | ATM CELL ID # 15 Arrival Time : 328 | • |
| | ATM CELL ID # 16 Arrival Time : 336 | |
| 4000 | ATM CELL ID # 17 Arrival Time : 372 | |
| | ATM CELL ID # 18 Arrival Time : 383 | |
| | ATM CELL ID # 19 Arrival Time : 452 | <u> </u> |
| Header Source A | ATM CELL ID # 20 Arrival Time : 467 | <u> 1</u> |
| | ATM CELL ID # 21 Arrival Time : 478 | |
| 0x00aaaa00 | AIM CELL ID # 22 Arrival Time : 402 | |
| | AIM CELL ID # 23 Arrival lime: 496 | |
| The second se | ATM CELL ID # 25 Arrival lime: 580 | ्र इ.स. |
| Header Source B | ATM LELL ID # 26 Arrival lime; b12 | 族 |
| 0×00bbbb00 | ATM CELL ID # 27 Arrival Time : 632 ATM CELL ID # 28 Arrival Time : 639 | |
| | | |
| Header Source C | Prob. Active | Prob. Emission |
| 0×00cccc00 | 0.500000 0.750000 | 0.093750 |
| Received Cells | | |
| | | in the second |
| 3614 DEC | Start Help | Exit is a |
| | · | |

Figure 6.3 Cell_delay module window

6.2 Hardware

The various FPGA circuits composing the virtual hardware system are presented next. A high level design floorplan is provided for each circuit as well as a description of its particularities. The interface to the SONET/ATM processor device implementing the physical layer of the system is also described. The FPGA design methodology as well as the various design tools used are presented.

6.2.1 FPGA design methodology

The module partitioning of the VHS has been done such that a single XC4010pg191 FPGA would be used on the test system PCB. This FPGA has 10000 equivalent gates partitioned as 400 configurable logic blocks (CLB). The logic capacity of a programmable device is usually expressed in such an equivalent gate count where

a gate is ostensibly a 2 input NAND [73]. Each CLB has two 4-to-1 function generators, one 2-to-1 function generator and two D type edge triggered flip-flops. Additionally, each user pin of the chip has an input/output block (IOB) that includes three state buffers and an unconditionally controlled flip-flop, in other words, a flip-flop without built-in latch enable control. These flip-flops are meant to shorten the long delays resulting from off-chip accesses by a factor near two.

The design entry level selected for the description of the various hardware modules is VHDL and it has been chosen primarily for the debugging and modification conveniences it offers throughout the design process. Synopsys 3.0b has been used for synthesizing the VHDL code into a low level Xilinx proprietary netlist format called Xilinx Netlist Format (XNF). Finally, the partition, placement and routing of the XNF files was executed by Xilinx Automated CAD Tools (XACT) utilities version 5.0.

6.2.2 Virtual hardware system modules

The main hardware modules of the system are presented next, along with a schematic block diagram showing how the hardware functions are arranged and how they interact with each other. The TDC1500 module will not be presented because of its simplicity. All modules have in common the interface to the SONET/ATM processor transmit and receive queues. This interface follows a standard called Universal Test and Operations Physical Interface for ATM (UTOPIA) [67]. This standard specifies the handshaking associated with the read operations from the receive queue and the write operations to the transmit queue. Data access to both queues is defined as 8-bit wide and is designed to withstand a clock rate in excess of 25MHz. Figure 6.4 presents the schematic block diagram of a generic SONET/ATM processor.



Figure 6.4 SONET/ATM processor block diagram

Each byte of an ATM cell gets written into the transmit queue by maintaining it on the queue input (Transmit_Data) while the write enable signal is being kept low (Write Enable) and the write strobe signal (Write Clock) is pulsed high. Additionally, the first byte of a cell is signaled to the transmit queue by maintaining the start of cell signal (*Start_of_Cell*) high while this first byte is written. The transmit queue also includes an almost-full flag signal (Full_Flag) that goes low to indicate that the transmit FIFO can only store five additional bytes. The almost-full flag signal goes high again when room becomes available for a complete cell. Similarly, each byte of an ATM cell received from the SONET/SDH input of the device can be clocked out of the receive queue output (Receive_Data) by pulsing the read strobe signal (Read_Clock) high while the read enable signal (Read Enable) is being kept low. Additionally, a start of cell signal (Start_of_Cell) goes high when the output of the receive queue currently holds the first byte of a cell. The receive queue also provides an empty-flag signal (*Empty_Flag*) that goes low to indicate that the receive FIFO is empty and that the byte on the *Receive_Data* output is invalid. The empty-flag signal goes back high when there is a complete ATM cell available.

6.2.2.1 File_transfer

The floorplan of the File_transfer module is illustrated in Figure 6.5. The major difficulty associated with this module is that the test system PCB SRAM is required by the cell transmitter and the cell receiver. Indeed, upon transfer start, the content of the SRAM is sent to the transmit queue of the SONET/ATM processor according to the user requested cell departure rate and at the same time (or later, depending on the loopback delay of the cells through the equipment under test) the returning cells have to be written back to SRAM. Two independent banks of SRAM could be used for the cell transmission and the cell reception. However, in order to reduce FPGA pin utilization, a single bank of SRAM is used and it is shared between the cell transmitter and the cell receiver. Sharing the SRAM bank also allows the transmitted files to be twice as large compared to the case where the SRAM is not shared. Given the overlapping of memory accesses that occurs when the SRAM is shared, a 32-bit wide memory interface as well as a word-to-byte converter on the transmit side and a byte-to-word converter on the receive side are used in order to reduce the frequency of the memory accesses to a resonable level.



Figure 6.5 File_transfer module block diagram

Given the high speed performance constraints imposed by the test system, the VHDL designs are such that all off-chip accesses are unconditionally registered. This way, a good synthesis and optimization tool can use the IOB flip-flops of the FPGA for the off-chip signals, thereby reducing the otherwise long propagation delays. Some of these off-chip signal registers are shown in the various circuit floorplans. This unconditional registering of signals proved being helpful but also increased the complexity of the various designs where handshaking is involved. For instance, in the case of the two UTOPIA interfaces, the registering adds one clock cycle delay in both directions of each handshaking chain. Therefore, incoporating the unconditional registering turned out to be much more than the simple addition of a flip-flop per signal.

It rather meant the redesign of all controls and datapaths involved with external handshaking.

6.2.2.2 Cell_error

The Cell_error module is simpler than the other modules in that it does not use the test system PCB memory. The major challenge involved with this design holds in the many counters that are needed to keep track of the various transmission errors. As shown on the floorplan in Figure 6.6, wide counters (16-bit) are used to monitor payload errors, cell losses and cell misinsertions. In fact, given that this test is rather targetted at long term monitoring, the width of the counters could ideally be much larger.



Figure 6.6 Cell_error module block diagram

Binary counters having a bad influence on the register-to-register critical path,

each of them has been described structurally in VHDL as two smaller 8-bit counters with the registered full-flag of the lower byte acting as the count enable of the higher one. This simple application of pipelining reduces the critical path by a factor of two but also introduces temporary discontinuities in the counting sequence. The discontinuity is not important as long as the count is only used a certain number of clock cycles following a count enable clock cycle. If the discontinuity is inadmissible, pipelining can still be used now by using the registered *full-1* flag of a lower stage as a count enable for the higher stage. Whereas the full flag is inherently part of any counter, the obtainment of the *full-1* flag requires the additional use of a comparator having same width as the counter section width. The payload of the transmitted cells is generated in real-time by a simple byte wide primitive polynomial based LFSR. On the receive side, an identical LFSR is reset upon each cell reception such that bit errors in all bytes of the payload can be monitored.

6.2.2.3 Cell_delay

The Cell_delay module is the most complex of all because the measurement of cell transfer delay requires being executed under various realistic traffic conditions. Ideally, this module would generate tens of different sources and each source would be modelled with a complex independent stochastic model. Both these constraints on the number of sources and the accuracy of the stochastic model have lead to outrageously complex circuits whose implementation would have required a ten-fold increase in resources. To achieve a reasonable compromise, the number of sources is brought down to three, as shown on the floorplan in Figure 6.7.

The foreground source (A) is being modelled by an interrupted binomial process whose probabilities Prob_remain_active, Prob_remain_inactive and Prob_emission are user-selectable before the execution of the test. The background sources (B,C) are modelled by simpler binomial processes using Prob_emission as their probability of success. In other words, B and C cells are generated during each cell time-slot with

probability Prob_emission.



Figure 6.7 Cell_delay module block diagram

In terms of hardware, the binomial trials ruling the cell emissions or the changes in the state of the foreground source are generated using an LFSR. It can be noticed that in the local sense, each bit of an LFSR is a binomial event with ½ success probability. In fact, since the all zeros sequence never shows up, the probability of an LFSR bit being one is slightly greater than the zero probability, but these second order effects are ignored here. Using many of these ½ success probability binomial events as building blocks, any arbitrary binomial random variable can be synthesized. For instance, the logical and of two bits of the LFSR creates a binomial event with a success probability of ¼. Similarily, the logical or of two binomial events is equivalent to creating another binomial event whose success probability is equal to the summation of the success probabilities of its components. The hardware structure used for the generation of a binomial event appears in Figure 6.8.



Figure 6.8 Binomial event generation for source modelling

In the Cell_delay module, each of the three binomial events required is built as the logical or of 5 binomial events having success probabilities of 0.5, 0.25, 0.125, 0.0625 and 0.03125. Preceeding the logical or, each binomial event itself is logically anded with a configuration bit provided by the user. These configuration bits appear as shaded flip-flops on Figure 6.8. Thus, the user is able to select the success probability of each binomial event among 32 different values ranging from 0.0313 to 0.969. It should be pointed out here that if a bit of an LFSR is used for the creation of two different binomial processes, these processes will consequently be stochastically dependent. Practically, we quickly run out of LFSR bits such that some of them have to be reused, thereby bringing some of this unwanted stochastic dependence. When using LFSRs as binomial process generators, their structure can be shaped in order to reduce the depency of adjacent bits that is found in the conventional linear structure. These structure modifications lead to the parallel LFSR and the segmented LFSR [65]. In the parallel LFSR, each flip-flop input is fed with the logical exclusive or of other flip-flop outputs. This way, each bit of the LFSR becomes even closer to a binomial event since its dependency upon its neighbors is now separated by a large number of clock cycles. The segmented LFSR is a compromise between the simple and the parallel LFSR as feedback paths are only used in some of the flip-flops, thereby segmenting the basic LFSR into sub-LFSRs. As was shown in Figure 6.8, it is this kind of LFSR with 15 flip-flops and 3 segments that was used for the Cell_delay module.

Chapter 7 System prototype evaluation

This chapter proceeds to the evaluation of the test system architecture devised and the prototype built. Each FPGA design is characterized in terms of area and timing statistics. The timing limitations of the system are then exposed, analyzed and explained. Alternative architectures are introduced and compared with the proposed architecture.

7.1 FPGA design issues

The performance of the test system designed and prototyped is strongly dependent upon the quality of the various underlying FPGA designs. In such a telecommunication application, optimizing the speed of the various FPGA modules was found to be the main constraint. It is a known fact that speed optimization is obtained at the expense of a greater area utilization but in the particular case of the synthesis tool used in the design process, marginal speed enhancements tended to result in prohibitive area increases.

7.1.1 Synthesis results

The synthesis results for the four FPGA designs TDC1500, File_transfer, Cell_error and Cell_delay are presented next. First, Table 7.1 presents the area statistics

for the four designs using the XC4010pg191 FPGA. The area is expressed in terms of the following utilized resources :

- Configurable logic blocks (CLB)
- 4-to-1 look-up table function generators (F-H)
- 2-to-1 look-up table function generators (H)
- CLB flip-flops (D type edge triggered)

The area statistics are presented for both free and forced pin assignment, in order to demonstrate the influence of a forced pin assignment on FPGA resources utilization. Forcing the pin assignment before the partitioning, placement and routing of the designs is necessary in the context of the virtual hardware system since many FPGA configurations have to be accomodated by a common pin assignment.

| | | CLB | | F-G | | H | | Flip-Flop | |
|------------|---------------|-----|----|-----|----|----|----|------------|----|
| | Module | # | % | # | % | # | | # # | % |
| Free | TDC1500 | 151 | 37 | 278 | 34 | 18 | 4 | 209 | 18 |
| Pin | File_transfer | 370 | 92 | 637 | 79 | 99 | 24 | 449 | 40 |
| | Cell_error | 263 | 65 | 472 | 59 | 29 | 7 | 239 | 21 |
| Assignment | Cell_delay | 374 | 93 | 725 | 90 | 38 | 9 | 485 | 60 |
| Forced | TDC1500 | 150 | 37 | 278 | 34 | 18 | 4 | 209 | 18 |
| Pin | File_transfer | 359 | 89 | 637 | 79 | 99 | 24 | 449 | 40 |
| | Cell_error | 263 | 65 | 472 | 59 | 29 | 7 | 239 | 21 |
| Assignment | Cell_delay | 378 | 94 | 725 | 90 | 38 | 9 | 485 | 60 |

Table 7.1 Area statistics

Timing statistics obtained from the Xdelay XACT tool for the four FPGA designs

are presented in Table 7.2. Each design is characterized in terms of pad-to-setup, clockto-setup and clock-to-pad delays all expressed in nanoseconds. Additionally, the expected maximum clock rate (MHz) is given. Each of these metrics is given for each design and for each combination of device speed grade (6 or 4) and pin assignment (free or forced). The speed grade 6 device is the slowest of the two and was used in the system prototype built.

| | | Module | Pad to Setup | Clock to Setup | Clock to Pad | Max. Freq. |
|------------------------------|-----------------------------|---------------|--------------------|----------------------|--------------------|---------------|
| | | TDC1500 | 45.3 | 49.1 | 71.3 | 14.0 |
| | Free Pin Assignment | File_transfer | ansfer 47.5 | | 132.0 | 7.6 |
| | | Cell_error | ror 35.5 89.3 | | 108.0 | 9.3 |
| XC4010pg191 | | Cell_delay | 44.0 | 84.0 | 75.5 | 11.9 |
| Speed Grade 6 | Forced Pin Assignment | TDC1500 | 52.0 | 53.8 | 70.9 | 14.1 |
| | | File_transfer | 40.7 | 93.8 | 91.9 | 10.7 |
| | | Cell_error | 41.2 | 104.8 | 104.2 | 9.5 |
| | | | 57.3 | 89.7 | 84.9 | 11.2 |
| | Free Pin Assignment | TDC1500 | 33.0 | 35.2 | 49.4 | 20.3 |
| | | File_transfer | 34.5 | 96.9 | 95.8 | 10.3 |
| | | Cell_error | 26.3 | 67.7 | 78.8 | 12.7 |
| XC4010pg191 Speed Grade 4 | | Cell_delay | 33.2 | 63.4 | 55.4 | 15.8 |
| | Forced Pin Assignment | TDC1500 | 38.4 | 38.8 | 49.0 | 20.4 |
| | | File_transfer | 30.5 | 69.5 | 66.4 | 14.4 |
| | | Cell_error | 30.7 | 77.8 | 76.3 | 12.9 |
| | | Cell_delay | 44.2 | 65.4 | 61.2 | 15.3 |

Table 7.2 Timing statistics

7.1.2 Speed limitations

As mentioned previously, speed is the main design constraint for the various FPGA designs of the system. The objective clock speed of the designs is dictated by the SONET/ATM processor and more specifically by the SONET STS-3c synchronous payload envelope (SPE) capacity. Each row of each STS-3c frame is 270 bytes wide and the first 9 bytes are used for section/line overhead whereas the 10th byte is used for path overhead. Therefore, 260/270 of each STS-3c frame is effectively used for ATM cell transport and given that the UTOPIA interfaces of the SONET/ATM processor have 8-bit wide datapaths, this results in a minimum required clock speed for the UTOPIA interfaces of 18.72 MHz. A particularity of the TDC1500APCM processor is that the empty-flag of the receive queue is only updated when the last byte of the last cell is removed from the queue. This forces the FPGA designs to interrupt the reading of the receive queue on each cell boundary in order to check the updated state of the emptyflag. This results in a waste of three clock cycles upon each received cell boundary. These three wasted cycles arise from the fact that the UTOPIA interfaces implemented by the FPGA designs are registered in order to reduce off-chip accesses latency. The first wasted cycle corresponds to the FPGA IOB latching of the empty-flag of the receive queue, the second wasted cycle corresponds to the update of the receive enable FPGA internal signal and the third wasted cycle is necessary for the updated receive enable signal to cross the FPGA IOB flip-flop and reach the TDC1500. Without the registering of the off-chip signals, only one clock cycle would be wasted upon each received cell boundary. For these reasons, the objective minimum FPGA clock speed required for sustaining the maximum throughput of UTOPIA interfaces should be around 21 MHz.

The timing statistics reveal that the forced pin assignment required in order to use a permanent FPGA wiring common to all modules of the virtual hardware system does not affect the timing statistics of the various designs significantly. By opposition, the timing statistics are strongly influenced by the device speed grade used. Table 7.2 shows
that the use of the fastest speed grade FPGA (4) would result in speed increases ranging from 35% to 45% with respect to slowest speed grade devices (6).

The timing statistics corresponding to the prototype built appear in the table under the header *device speed grade* 6 and *forced pin assignment*. The specified maximum operating speeds are 14.1MHz, 10.7MHz, 9.5MHz and 11.2MHz for TDC1500, File_transfer, Cell_error and Cell_delay modules respectively. These figures do not compare well with the objective clock speed of 21MHz and this suggests that a higher speed grade device than 6 should be used since the current pipeline oriented FPGA designs are not sufficient to provide the performance required.

7.1.2.1 Synthesis tool

All four FPGA designs were described using VHDL and synthesized using Synopsys 3.0b VHDL compiler with Xilinx 4000 series technology library. The use of VHDL as the design entry proved to be very efficient as it lead to easy circuit description, verification, simulation and modification. Furthermore, the large synthesizable subset of VHDL supported by the synthesis tool used allowed short and easy to read circuit descriptions. An important drawback of the version of the VHDL compiler used is that it does not use the built-in latch enable control associated with each CLB flip-flop of the FPGA. Thus, every conditional registering in the VHDL source gets synthesized as a permanently enabled flip-flop whose input is fed with some combinational function of its own output signal, the condition signal and other signals. A better synthesis tool would simply route the condition signal to the latch enable control of the flip-flop and there would not be any feedback required. The resource overhead associated with each conditional registering roughly consists of a 4-to-1 function generator as well as the associated routing resources. Additionally, a corresponding timing overhead results from this weakness of the synthesis tool.

In general, behavioural VHDL as the entry level for the design process allows fast design space exploration, fewer errors and shorter design cycles. Given that it is

technology independent, its use requires less knowledge of layout-circuit design and therefore becomes very attractive for system designers. Nevertheless, this technological independence can come at the price of a performance decrease with respect to an equivalent design obtained through schematic capture. In the particular case of FPGAs, datapath logic synthesis (synthesis of arithmetic and relational logic) should be technology dependent in order for the technology specific FPGA resources to be used for maximizing the resulting performance. These FPGA specific resources are meant to complement the conventional function generators and flip-flops which are suitable for random logic synthesis (finite state machines and combinational logic). Specific resources for Xilinx 4000 series FPGAs include wide decoders in the periphery of the chip, fast carry generation circuitry in each CLB, IOB flip-flops and conversion of look-up table function generators into RAM cells.

On the other hand, structural VHDL can allow technology dependent circuit descriptions through component instantiations of soft and hard macros of a specific FPGA library. Soft macros are entities that are already synthesized under a specific FPGA technology whereas hard macros are entities that are synthesized, placed and routed for a specific FPGA technology. These macros make an extensive use of the FPGA technology specific internal resources. So, full access to these specific FPGA resources can be obtained by sacrificing the advantages of behavioural technology independent VHDL descriptions to the profit of structural technology dependent VHDL descriptions.

A middle of the road approach that can be used by synthesis tools is to provide a *module generation* [74] library for each FPGA technology. This library can contain pre-defined and user-defined technology dependent implementations of various datapath operators. Whenever a supported arithmetic or relational operator is encountered in the purely behavioural VHDL source, the module generation library is consulted by the synthesis tool for a matching implementation. This technique can constitute a reasonably efficient way of utilizing technology specific resources while maintaining a technology independent top-down design flow with a logic synthesis tool. Synopsys 3.0b uses this approach for the inference of hard macro adders from the '+' VHDL operator.

7.1.2.2 Forced routing penalty

Table 7.1 presented area statistics for both free and forced pin assignment, in order to gauge the influence of a forced pin assignment on resources utilization. The statistics show that this influence is negligible and this is explained by the relatively low pin count used for the designs, namely 96 (60%) for TDC1500, 90 (56%) for File_transfer, 37 (23%) for Cell_error and 82 (51%) for Cell_delay. Additionally, Table 7.2 reveals that the forced pin assignment causes no significant degradation of the critical paths of the various FPGA designs. It is interesting to notice that the effects of a forced pin assignment remain negligible for designs having a CLB utilization as high as 90%. Curiously, the CLB utilization of the File_transfer design is 3% lower for the forced pin assignment case. This is due to the fact that for this particular case, the placement, partitioning and routing tools have been used with higher effort options.

7.1.2.3 FPGA architecture

An FPGA architecture can be characterized in terms of its *logic block* architecture, its routing architecture and its programming technology. The routing architecture used in 4000 series Xilinx FPGAs consists of an array of switching blocks interleaved with the array of logic blocks. The switching blocks are connected in a twodimensional grid such that any two logic blocks of the FPGA can be linked together. This routing is qualified as *segmented* since a connection between two CLBs will use a variable number of metal segments and switching blocks. Therefore, with such a routing architecture, the placement and routing of the logic blocks of a circuit have to be executed in a closely cooperative fashion in order for the critical path requirements to be met. The place and route complexity for the segmented routing architecture has been shown to be an exponentional function of the design complexity and the percentage utilization of the device whereas other routing architectures featuring a greater connectivity and continuous interconnects are known to have a lesser place and route complexity [75,76]. This exponential nature of the problem is circumvented through the use of non-deterministic optimization algorithms by the place and route tools. Throughout the placement and routing of the various FPGA designs of the system, it has been noticed that the non-deterministic character of these algorithms can sometimes lead to significantly different results (from the performance viewpoint) when applied successively to the same circuit.

The logic block architecture of the FPGA used consists of two 4-to-1 look-up table function generators, one 2-to-1 look-up table function generator and two D type edge triggered flip-flops arranged in a CLB. The *magical* 4 number used as the width of the combinational function generators originates from studies conducted on logic and routing area requirements optimization for general applications [77,78]. It has been found that area optimization requires 3-input or 4-input logic blocks whereas delay optimization requires 5-input (or wider) logic blocks. Now, for the specific case of telecommunication applications. As an example, the four FPGA modules of the test system designed all make an extensive use of wide counters and wide pattern matchers. This suggests that an FPGA architecture with wider gates would enhance significantly the timing performances without causing too much area overhead in the form of partly used wide gates.

PROTEUS is a programmable hardware architecture that is specifically targetted at telecommunications applications [79,80]. Its specifications proceed from an exhaustive characterization of telecommunication subsystem resources. Most subsystems analyzed have been found to use resources very similar to the ones of the test system designs, namely wide binary counters, pattern matchers and finite state machines characterized by many states and few transitions. These features are exactly the same as the ones found in the various modules of the ATM switching node test system designed. Applying technology mapping to the various subsystems with different combinations of programmable gate widths, it has been concluded that the most efficient logic coverage is obtained using a mixture of 3-input function generators and wide gates with more than 4 inputs. The size of the basic function generator of an architecture affects the routing complexity dramatically. From the standpoint of logic coverage efficiency, small function generators of 2 or 3 inputs are highly desirable but they lead to very complex routing issues. Despite this, PROTEUS still recommends 3-input function generators on the premise that a large fraction of the connections in the subsystems analyzed are local connections.

7.2 Alternative architectures

A clear limitation of the current test system design is the low bandwidth link that is used to connect the system PCB to the host computer. If a higher bandwidth connection were used, then the cell analyzer could use the computer memory for storage of information such that no or very few memory would be required on the test system PCB. Given that the amount of host memory is much bigger than any realistic amount of test system PCB static memory, the test runs could be made much longer. Given that the ultimate goal for an ATM test system is to allow the generation of an important number of traffic sources and the analysis of much fewer sources, there is an important assymetry in the input and output bandwidth of the system. Whereas the host could be used for traffic analysis provided a suitable connection to the test system, this same host could hardly be used for the traffic generation because of the much higher bandwidth involved. Therefore, in order for the test system architecture to offer some degree of scalability, all or most of the traffic generation should be provided by dedicated hardware.

7.2.1 Host interface paradigm

The design of a test system making use of the host memory is very similar to the design of an ATM host interface, or as sometimes called *ATM adapter*. Therefore,

models of ATM host interfaces found in the literature [81,82,83,84] can be applied directly to the test system design. All ATM host interfaces share two main functionalities, namely data conversion between host format and network format and data transfer between user memory on the host and network transmission medium.

The engineering decisions related to the data conversion functions concern mainly their distribution. In other words, it has to be decided for each layer of the protocol if it gets implemented by the host software or by the interface. Traditional approaches for transport layers other than ATM consisted in keeping most protocol functions in the host. This way, end-to-end performance could scale with each new generation of workstation and the interface implementation could be kept simple and low cost. Now, in the case of ATM technology, the higher traffic bandwidth involved dictates a function distribution that prevents host overloading through a broader interface functionality. Therefore, all three layers of ATM should be implemented on the interface, thereby allowing user application data units (packets) to be transferred directly to the interface with minimal host CPU processing involvement.

In the area of data transfer between host memory and interface, there are various ways leading to various levels of performance and implementation complexity. First, the physical path used for the transfer can be either the host memory bus or the host I/O bus. Host memory buses provide better throughput and latency characteristics than their I/O bus counterpart since they are located higher in the subsystem hierarchy. However, an I/O bus based interface design is more likely to survive many generations of systems. Furthermore, building a system upon a widely publicized I/O bus specification is easier than building it upon some proprietary memory bus specification.

Once the physical path for the transfer is selected, it remains to be selected how the transfer itself will be controlled. In the case of memory bus based interface, some form of shared or dual-ported high speed memory is used to hold transmitted and received cells. For dual-ported memory, control transfer is trivial whereas with shared memory, an arbiter is used to grant the memory bus control to the host or the interface. I/O bus based interfaces can simply be slave devices mapped into the I/O addressing space of the host system. In this case, a transfer from host to interface becomes a simple I/O write operation and a transfer from interface to host becomes an I/O read operation triggered by polling or interrupt. Such programmed I/O solutions lead to outstanding simplicity of the interface itself but also to a severe utilization of CPU time, memory bus and I/O bus.

In order to dismiss the host CPU from the task of the data transfer, the ATM interface can use the direct memory access (DMA) functions that are provided by most I/O buses. This way, when a transfer occurs between the host and the interface in one direction or the other, the host CPU execution can remain uninterrupted through its use of cache memory. Peak information transfer rates using I/O bus DMA functionalities vary according to specific buses. Specifications are given in Table 7.3 for the most common commercial interface buses.

| Bus Interface | Bus Master Peak Burst Transfer Rate MByte/sec |
|---------------|---|
| EISA | 33 |
| MCA | 20 |
| NuBus | 37.5 |
| PCI | 132 |
| VL | 85-160 |
| VME | 40-80 |
| Multibus I | 10 |
| Multibus II | 40 |
| SBus | 100 |

 Table 7.3 IO bus peak transfer rate

Conventional DMA functions usually allow simple word aligned memory block transfers and in the absence of guarantees on the alignement of data provided by the protocol layers to the interface, an additional memory-to-memory data transfer may be required from the host CPU. It has been shown that this additional transfer brings the throughput of simple DMA interfaces down to the level of programmed I/O interfaces [83]. However, the alignment can be done by the interface itself through the use of a barrel shifter and this way, the throughput in the transmit and receive directions get improved significantly with respect to programmed I/O interfaces.

Using such an I/O bus-DMA based architecture for the test system, the host memory could be used for the recording of events and statistics during the test runs. Therefore, the architecture of the cell analyzer of the current test system would change significantly since it would not interface to the test system PCB memory anymore but rather to the host memory through DMA. On the other hand, the architecture of the cell generator of the test system would not change so much because all background sources would still need to be implemented in hardware on the test system PCB. The reason for this is that the DMA bandwidth would no be sufficient to generate the high bandwidth background sources from the host. Nevertheless, the foreground source(s) could be generated by the host, depending on their number and their associated bandwidth.

7.2.2 Scalability issues

From a practical perspective, the current switching node test system could be expanded in terms of the number of physical links and the number of sources generated, in order to match the number of ports and the port speeds of larger scale switching nodes. The expansion of the number of physical links of the system would consist in duplicating the current architecture as needed. Therefore, the FPGA, SRAM and SONET/ATM processor combination would become the building block of the expanded system. Some additional circuitry would be necessary to allow the building blocks to be accessed and controlled individually as well as for providing their synchronization. Concerning the expansion of the number of traffic sources per physical link, the current architecture of the system is also found to scale linearly in terms of resources. For the case where a traffic source is simulated by an interrupted binomial process as is the case in the Cell_delay module, it has been seen that a 15-bit segmented LFSR as well as a 20-input pipelined combinational function for the generation of each binomial event of the model (*Remain_Active, Remain_Inactive, Emission*) are required. Additionally, a finite state machine is required to update the state of each source during each cell timeslot and a FIFO queue is needed to record for each time-slot the identifier of the sources that want to emit a cell. Despite the linear growth of resources as a function of the number of traffic sources generated, the individual resources for each source are so important that generating 10 or 20 sources for a single STS-3c link would lead to a quite hardware extensive system.

In order to alleviate the resource requirements associated with a bigger number of sources, the *stochastic process generation*⁺ paradigm should be adopted so that realtime scheduling of cell departures for all sources of a physical link can be executed by a single stochastic process. Simple discrete-time stochastic processes can be synthesized to hardware easily using LFSRs whereas continuous-time stochastic models involving complex arithmetic could be implemented through the use of a pipelined general processor or a digital signal processor. The FPGAs of the system would then be used to receive command vectors from the scheduler processors, for time stamping of foreground source cells and for event recording in local SRAM.

⁺ See source modelling taxonomy introduced in chapter 4

Chapter 8 Conclusion

After going through many years of queueing, bandwidth allocation, traffic flow and traffic congestion analyses, the asynchronous transfer mode will now be reaching the commercial market massively. It has demonstrated its ability to integrate the various classes of traffic of the future and provide information switching uniformly throughout the whole spectrum of network implementations, from the local area to the worldwide area.

As ATM technology deployment arises, it brings along the needs for suitable testing methods and equipments. Indeed, in order for this technology to become a success, it must not only be effective, reliable and affordable but it must also be easy to test during design, implementation and day-to-day operations. The same technological advantages that benefit ATM broadband networks also engender some new challenges for the testing task. The flexibility of ATM in terms of the traffic classes as well as the physical interfaces it supports and the ever increasing aggregate bandwidth of the switching nodes are examples of ATM features that complicate the testing task.

Testing the ATM network consists first in assessing the conformance of the various network elements such as the switching nodes, multiplexers and user terminals through the use of special purpose high capability probing equipment. Then, the testing task is taken at a higher level as it proceeds to end-to-end connection conformance

assessment through the use of embedded mechanisms providing a minimum transfer performance measurement. The test of switching nodes is particularily important since these are the elements that strongly influence the quality of the cell transfer. This task of switching node test consists in the measurement of a series of test parameters that are of two types, namely functional parameters and performance parameters.

This thesis presents a design and implementation of a test system for ATM switching nodes. In order for the test system to be flexible enough to adapt to evolving standards and to possible vendor specific features, reconfigurable hardware is used. Static read and write memory based field-programmable gate array technology is used to provide the system flexibility required and also to minimize the component requirements through the use of hardware metamorphosis and reuse. This use of in-system reconfigurable hardware for swapping some mutually time exclusive hardware functions is known as virtual hardware, hardware subroutines or silicon multi-tasking.

The system proposed uses a personal computer for the realization of a user friendly interface to the test system printed circuit board. The system provides various tests that the user can configure and activate with the pointing device. Each test execution activates a specific configuration of the system FPGA that is specially optimized. Fast static memory is used on the test system PCB for the real-time recording of events and violations occuring during the test runs. The SONET STS-3c and SDH STM-1 physical interface of the system is provided by a highly integrated, wide functionality application specific circuit obtained from Texas Instruments.

The reconfigurable hardware computing paradigm is recognized to efficiently combine the versatility of a programmable solution with the performance of dedicated hardware. It is emerging in commercial applications as a way of reducing system costs and sizes without sacrificing performance. Moreover, certain new lines of SRAM based FPGAs are specially targetted at this dynamic hardware reuse as they offer more efficient pin assignment capabilities, incremental device configuration and shorter configuration delays. Reconfigurable hardware use is obviously limited by the hardware concurrency of a particular system. Nevertheless, in the case of the ATM switching node test system, it has been shown that it is possible to partition the system into various modules that can be executed sequentially. Therefore in this context, the reuse of the FPGA is found to be very advantageous.

The evaluation of the test system prototype brought to light the heavy influence of the synthesis tool device knowledge on the synthesis results quality, especially with the complex FPGAs available today. The SONET/ATM processor proved to be a very practical device as it automates ATM cell mapping and extraction as well as clock generation and recovery. The use of test system PCB memory for the recording of test events allowed the use of a low speed connection to the computer system and relieved the GUI software from real-time processing requirements. As was shown, the ATM adapter configuration described in section 7.2.1 could be used instead, in order for the host CPU and memory to participate more actively to the traffic generation and analysis tasks. The flexible, affordable and wide functionality ATM switching node test system presented could find its application in the industrial, commercial or educational fields.



Appendix A

Key B-ISDN documents⁺

| ITU-T | I.113: B-ISDN Vocabulary of terms I.121R: Broadband Aspects of ISDN I.150: B-ISDN ATM Functional Characteristics | |
|----------|--|--|
| | | |
| | | |
| | I.211: B-ISDN Service Aspects | |
| | I.311: B-ISDN General Network Aspects | |
| | 1.321: B-ISDN Protocol Reference Model and Its Applications | |
| | 1.327: B-ISDN Functional Architecture Aspects | |
| | I.361: B-ISDN ATM Layer Specification | |
| | I.362: B-ISDN ATM Adaptation Layer Functional Specification | |
| | 1.363: B-ISDN ATM Adaptation Layer Specification | |
| | 1.371: B-ISDN Traffic Control and Congestion Control | |
| | I.413: B-ISDN User-Network Interface | |
| | I.432: B-ISDN User-Network Interface - Physical Layer | |
| | I.610: B-ISDN UNI Operation and Maintenance Principles | |
| Bellcore | TA-NWT-001110: Broadband ISDN Switching System Generic Requirements | |
| | TA-NWT-0011111: Broadband ISDN Access Signaling Generic Requirements | |
| | TA-NWT-001112: Broadband ISDN User to Network Interface and Network Node | |
| | Interface Physical Layer Generic Requirements | |
| | TA-NWT-001113: Asynchronous Transfer Mode and ATM Adaptation Layer | |
| | Protocols Generic Requirements | |
| | TA-TSV-001408: Generic Requirements for Exchange PVC Cell Relay Service | |
| | TA-TSV-001408: Generic Requirements for Exchange Access PVC Cell Relay | |
| | Service | |
| | TA-TSV-001408: Generic Requirements for Exchange SVC Cell Relay Service | |
| | | |

⁺ WorldWide Web sites : http://www.itu.com & http://www.bellcore.com

page 112

J

ł

References

1. Gagnon S., Szymanski T., *Field-programmable gate array based ATM switching node test system*, Proceedings of the 3rd Canadian Workshop on Field-Programmable Devices (FPD'95), Montréal, June 1995.

 Bertsekas D., Gallager R., *Data networks*, Prentice-Hall, New-Jersey, 1992, pp. 16.
 International Telecommunication Union - International Telegraph and Telephone Consultative Committee (ITU-CCITT) Recommendation I.321, *B-ISDN protocol reference model and its application*, Geneve, 1991.

4. Rooholamini R., Cherkassky V., Garver M., Finding the right ATM switch for the market, IEEE Computer, April 1994, pp. 16-28.

5. Fisher W., Fundneider O., Goeldner E.-H., Lutz K.A., A scalable ATM switching system architecture, IEEE Journal on Selected Areas in Communications, Vol. 9, No. 8, Oct. 1991, pp. 1299-1307.

6. Doi Y., Yamada H., A 160 Gbit/s large-capacity ATM switching system using a dynamic link speed controlled switch architecture, IEEE Globecom 1993, Vol. 1, pp. 24-28.

7. Eng K.Y., Pashan M.A., Spanke R.A., Karol M.J., Martin G.D., A highperformance prototype 2.5 Gb/s ATM switch for broadband applications, IEEE Globecom 1992, Vol. 1, pp. 111-117.

8. Endo N., Kozaki T., Ohuchi T., Kuwahara H., Gohara S., Shared buffer memory switch for an ATM exchange, IEEE Transactions on Communications, Vol. 41, No. 1, January 1993, pp. 237-245.

9. Anderson T.E., Owicki S.S., Saxe J.B., Thacker C.P., *High-speed switch scheduling* for local-area networks, ACM Transactions on Computer Systems, Vol. 11, No. 4, Nov. 1993, pp. 319-352.

10. Karol M.J., Hluchyj M.G., Morgan S.P., Input versus output queueing on a space division packet switch, IEEE Transactions on Communications, Vol. COM-35, No. 12, Dcc. 1987, pp. 1347-1356.

11. Sarkies K.W., The bypass queue in fast packet switching, IEEE Transactions on Communications, Vol. 39, No. 5, May 1991, pp. 766-774.

12. De Prycker M., Asynchronous transfer mode for broadband ISDN, Ellis Norwood, New-York, 1993, pp. 147-231.

13. Melen R., *Current architectures for ATM implementation*, European Transactions on Telecommunications and Related Technologies, Vol. 3, No. 2, March-April 1992, pp. 145-155.

14. Pattavina A., Non blocking architectures for ATM switching, IEEE Communications Magazine, Feb. 1993, pp. 38-48.

15. Gupta A.K., Barbosa L.O., Georganas N.D., Switching modules for ATM switching systems and their interconnection networks, Computers Networks and ISDN Systems, Vol. 26, 1993, pp. 433-445.



16. Banniza T.R., Eilenberger G.J., Pauwels B., Therasse Y., *Design and technology aspects of VLSI's for ATM switches*, IEEE Journal on Selected Areas in Communications, Vol. 9, No. 8, Oct. 1991, pp. 1255-1264.

17. Itoh A., Takahashi W., Nagano H., Kurisaka M., Iwasaki S., *Practical implementation and packaging technologies for a large-scale ATM switching system*, IEEE Journal on Selected Areas in Communications, Vol. 9, No. 8, Oct. 1991, pp. 1280-1288.

18. Banwell T.C., Estes R.C., Habiby S.F., Hayward G.A., Helstern T.K., Lalk G.R., Mahoney D.D., Wilson D.K., Young K.C., *Physical design issues for very large ATM switching systems*, IEEE Journal on Selected Areas in Communications, Vol. 9, No. 8, Oct. 1991, pp. 1227-1237.

19. Bonomi F., Fendick K.W., *The rate-based flow control framework for the available bit rate ATM service*, IEEE Network, Vol. 9, No. 2, March/April 1995, pp. 25-39.

20. Gun L., Guerin R., Bandwitdth management and congestion control framework of the broadband network architecture, Computer Networks and ISDN Systems, Vol. 26, 1993, pp. 61-78.

21. Butto M., Cavallero E., Tonietti A., *Effectiveness of the Leaky Bucket policing mechanism*, IEEE Journal on Selected Area in Communications, Vol. 9, No. 3, April 1991, pp. 335-342.

22. Wu G.-L., Mark J.W., Discrete time analysis of leaky bucket congestion control, Computer Networks and ISDN Systems, Vol. 26, 1993, pp. 79-94.

23. Rathgeb E.P., *Modeling and performance comparison of policing mechanisms for ATM networks*, IEEE Journal on Selected Area in Communications, Vol. 9, No. 3, April 1991, pp. 325-334.

24. Tarraf A.A., Habib I.W., Saadawi T.N., A novel neural network traffic enforcement mechanism for ATM networks, IEEE Journal on Selected Area in Communications, Vol. 12, No. 6, Aug. 1994, pp. 1088-1095.

25. Chlamtac I., Zhang T., A counter based congestion control (CBC) for ATM networks, Computer Networks and ISDN Systems, Vol. 26, 1993, pp. 5-27.

26. Newman P., Traffic management for ATM local area networks, IEEE Communications Magazine, Aug. 1994, pp. 44-50.

27. Kung H.T., Morris R., Credit-based flow control for ATM networks, IEEE Networks, Vol. 9, No. 2, March/April 1995, pp. 40-48.

28. Ramakrishnan K.K., Newman P., Integration of rate and credit schemes for ATM flow control, IEEE Network, Vol. 9, No. 2, March/April, pp. 49-56.

29. Minoli D., Dobrowski G., Principles of signaling for cell relay and frame relay, Artech House, Boston, 1995, 305 p.

30. Jeffrey M., Asynchronous Transfer Mode : the ultimate solution ?, Electronics and Communication Engineering Journal, June 1994.

31. Biagioni E., Cooper E., Sansom R., *Designing a practical ATM LAN*, IEEE Network, Vol. 7, No. 2, March 1993, pp. 32-39.

32. ANSI Accredited Standards Committee T!X1.2, Document T1X1.2/93-024R2, A technical report on a comparison of SONET (Synchronous Optical NETwork) and



SDH (Synchronous Digital Hierarchy)

33. Ballart R., Ching Y.-C., SONET: Now it's the standard optical network, IEEE Communications, Vol. 29, No. 3, March 1989, pp. 8-15.

34. Mattews M., Newcombe P., The synchronous digital hierarcy, part 1: The origin of the species, IEE Review, Mayy 1991, pp. 185-189.

35. Ferguson S.P., *Implications of SONET and SDH*, Electronics and Communication Engineering Journal, June 1994, pp. 133-142.

36. Owen H.L., Klett T.M., Synchronous digital hierarchy network pointer simulation, Computer Networks and ISDN Systems, Vol. 26, 1994, pp. 481-491.

37. Sexton M., Reid A., *Transmission networking : SONET and the Synchronous Digital Hierarchy*, Artech House, Boston, 1993, pp. 101.

38. ITU:Recommendation I.610. "OAM principles for the B-ISDN access", Geneva, 1991.

39. The ATM Forum, User-Network Interface Specification (version 3.0), Prentice Hall, New Jersey, 1993, 389p.

40. ITU:Recommendation I.610, OAM principles of B-ISDN access, Geneva, June 1992.

41. Gruber, J., Leeson, J., Performance in evolving SONET/SDH networks, Telesis, No. 95, 1993, pp.17-29.

42. Murakami, H., Sato, N., Okamoto, T., Monitoring method for cell transfer performance in ATM networks, NTT Review, Vol. 4, No. 4, July 1992, pp. 39-44.
43. Dragos Ruiu, Testing ATM Systems, IEEE Spectrum, Vol. 31, No. 6, June 1994, pp. 25-27

44. M. Di Concetto at al., *Testing methods and equipments for ATM switching nodes*, European transactions on telecommunications, Vol. 5, No. 3, May-June 1994, pp. 81-89.

45. Pierre Langlois, ATM Network Testing, Telecommunications International Edition, Vol. 28, No. 2, Feb 1994, pp. 39-45

46. Helvik B.E., Melteig O., Morland L., *The synthesized traffic generator; Objectives, design and capabilities*, Integrated broadband communication networks and services, 1994, pp. 287-302.

47. Chen D. X., Mark J. K., Delay and loss control of an output buffered fast packet switch supporting integrated services, Proc. ICC 1992, pp. 335A.1.1-335A.1.5.

48. Lau X.-C., Li S.-Q., Traffic analysis in large-scale high-speed integrated networks : validation of nodal decomposition approach, Infocom 1993, pp. 11a.2.1-11a.2.10.

49. Paxson V., Floyd S., *Wide-area traffic: The failure of Poisson modelling*, Proceedings of SIGCOMM'94 conference on communications architectures, protocols and applications, Vol. 24, No. 4, Oct. 1994.

50. D'Agostino R.B., Stephens M.A., Goodness-of-fit techniques, Marcel Dekker, 1986.

51. Friesen V.J., Wong J.W., The effect of multiplexing, switching and other factors on the performance of broadband networks, Infocom 1993, pp. 10a.4.1-10a.4.10.
52. Brady P.T., A statistical analysis of on-off patterns in 16 conversations, Bell Syst. Tech. J., vol. 47, no. 1, pp.73-91, Jan. 1968.



53. Unteregelsbacher E., Mouftah H.T., *Pdf based congestion control in ATM networks*, ICC 1991, pp. 6.6.1-6.6.5.

54. Jenq Y.C., Approximations for packetized voice traffic in statistical multiplexers, Proc. INFOCOM84, pp. 256-259.

55. Yang T., Tsang D.H.K., A novel appraoch to estimating cell loss probability in an ATM multiplexer loaded with homogeneous bursty sources, Proc. GLOBECOM 1992, pp. 511-517.

56. Chan J.H.S., Tsang D.H.K, Bandwidth allocation of multiple QOS classes in ATM environment, Proc. ICC 1994, pp. 3c.1.1-3c.1.8.

57. Yegenoglu F., Jabbari B., Performance evaluation of MMPP/D/1/K queues for aggregate ATM traffic models, Infocom 1993, pp. 11a.1.1-11a1.6.

58. Heffes H., Lucantoni D.M., A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance, IEEE Journal on Selected area in Communications, Vol. SAC-4, No. 6, Sept. 1986.

59. Kishimoto R., Ogata Y., Inumaru F., Generation interval distribution characteristics of packetized variable rate video coding data streams in an ATM network, IEEE Journal on Selected area in Communications, Vol. 7, No. 5, June 1989.

60. Maglaris B., Anastassiou D., Sen P., Karlsson G., Robbins J.D., *Performance models of statistical multiplexing in packet video communications*, IEEE transactions on communications, Vol. 36, No. 7, July 1988.

61. Sen P., Maglaris B., Rikli N.-E., Anastassiou D., *Models for packet switching of variable-bit-rate video sources*, IEEE transactions on communications, Vol. 7, No. 5, June 1989.

62. Leduc J.-P., Digital moving pictures - Coding and transmission on ATM networks, Elsevier, Amsterdam, 1994, 558 p.

63. Melteig O., Introduction to the PARASOL project, The 9th Nordic Teletraffic Seminar, Norwegian Telecom Research Department, August 21-23, 1991.

64. Ross S.M., A course in simulation, Macmillan publishing company, New York, 1990, 200 p.

65. Bardel P.H., McAnney W.H., Savir J., Built-in test for VLSI (pseudorandom techniques), John Wiley and Sons inc., New-York, 1987, 354p.

66. Ross S.M., A course in simulation, Macmillan publishing company, New York, 1990, pp. 200.

67. ATM and SONET Broadband solutions for LAN and WAN applications, Advanced System Logic Products, Texas Instruments, April 1994.

68. Petzold C., *Programming Windows 3.1*, Microsoft Press, Redmond, 1992, 983 p. 69.Eggebrecht L.C., *Interfacing to the IBM personal computer*, SAMS, Carmel, 1990, 345 p.

70. Albaharna O.T., Cheung, P.Y.K., Clarke T.J., Area & time limitations of FPGAbased virtual hardware, 1994 IEEE International conference on computer design, pp. 184-188.

71. Hastie N., Cliff R., The implementation of hardware subroutines on field programmable gate arrays, IEEE 1990 Custom Integrated Circuits Conference, pp.



31,4.1-31,4.4.

72. CCITT, Draft Recommendation I.35.B, *Broadband ISDN performance*, WP XVIII/6 (part 2, draft recommendations), TD 15 (XVIII).

73. Brown S.D., Francis R.J., Rose J., Vranesic Z.G., *Field Programmable Gate* Arrays, Kluwer Academic Publishers, Boston, 1992, 206 p.

74. Dekker R., Ligthart M., Lapides L., HDL synthesis for FPGA design, Electronic Engineering, Vol. 66, No. 814, Oct. 1994.

75. Amos D., Interconnect trade-offs: CPLD vs FPGA, electronics engineering, Vol. 67, No. 819, march 1995, pp. 81-84.

76. Rose J., El Gamal A., Sangiovanni-Vincentelli A., Architecture of Field-Programmable Gata Arrays, Proceedings of the IEEE, Vol. 81, No. 7, July 1993, pp. 1013-1029.

77. Rosc J.S., Francis R.J., Lewis D., Chow P., Architecture or programmable gate arrays: The effect of logic block functionality on area efficiency, IEEE Journal on Solid State Circuits, Vol. 25, No. 5, Oct. 1990, pp. 1217-1225.

78. Singh S., Rose J., Chow P., Lewis D., *The effect of logic block architecture on FPGA performance*, IEEE Journal on Solid State Circuits, Vol. 27, No. 3, March 1992, pp. 281-287.

79. Ohta N., Nakada H., Yamada K., Tsutsui A., Miyazaki T., *PROTEUS: Programmable hardware for telecommunication systems*, 1994 IEEE International Conference on Computer Design, pp. 178-183.

80. Ohta N., Hayashi K., Miyazaki T., *Reconfigurable system architecture for high-speed telecommunications*, Proceedings of the Third Canadian Workshop on Field-Programmable Devices, Montreal, June 1995, pp. 173-178.

81. Traw C.B.S., Smith J.M., Hardware/Software organization of a high-performance ATM host interface, IEEE Journal on Selected Area in Communications, Vol. 11, No 2., Feb. 1993, pp. 240-253.

82. Davie B.S., The architecture and implementation of a high speed host interface, IEEE Journal on Selected Area in Communications, Vol. 11, No 2., Feb. 1993, pp. 228-239.

83. Ramakrishnan K.K., *Performance considerations in designing network interface*, IEEE Journal on Selected Area in Communications, Vol. 11, No 2., Feb. 1993, pp. 203-219.

84. Moors T., Cantoni A., ATM receiver implementation issues, IEEE Journal on Selected Area in Communications, Vol. 11, No 2., Feb. 1993, pp. 254-263.