Application of an Automatically Designed Fuzzy Logic Decision Support System to Connection Admission Control in ATM Networks

Maria Fernanda Natário Ramalho

10 December 1996

Submitted for the Degree of Doctor of Philosophy*

Supervised by Dr. Eric M. Scharf
Department of Electronic Engineering
Queen Mary and Westfield College
University of London

* This work has been sponsored by Junta Nacional de Investigação Científica e Tecnológica (Portugal) under Grant no. BD/2730/93-IA after October 1994 and by Centro de Estudos deTelecomunicações - Telecom Portugal before that date.

“ … to all of those who have the courage to be different in order to be themselves.”

(Anon.)

Abstract

This thesis presents a fuzzy logic based approach to implementation of Connection Admission Control (CAC) in Asynchronous Transfer Mode (ATM) networks: Fuzzy logic based CAC (FCAC); which addresses the unpredictability of the effects on cell loss when several connections are multiplexed on an ATM link. Providing there is past knowledge about ATM traffic behaviour, a fuzzy logic approach, by virtue of its flexibility in expressing the non-linearity between the load in the ATM link and the maximum cell loss ratio per connection, in non-homogeneous traffic scenarios, has an advantage over more conventional approaches which are based on traffic (source) models.

Genetic Algorithms (GA) are applied to automatically design the fuzzy rule base and the fuzzy sets for each of the fuzzy variables using a method of identification of fuzzy logic systems from examples. The examples refer to on-line measurements of traffic parameters comprising the peak rate, mean rate, mean burst length and cell loss ratio for of each of the multiplexed connections. Every time a new set of measurements is available, a tuning process is invoked so allowing FCAC to adjust itself to the changing traffic patterns.

Validation experiments of FCAC cell loss prediction were performed for homogeneous and heterogeneous traffic scenarios by considering different link capacities and output buffer sizes. The predicted cell loss values were compared with cell loss obtained either using an ATM simulator or on-line measurements performed on an ATM test bed.

FCAC cell loss prediction was also compared with an “enhanced” convolution based approach, ECA, and a theoretical approach based on Markov chains and queueing analysis, the (M+1)-MMDP approximation, where M is the number of sources in the traffic mix. FCAC does not favour any type of traffic or link configuration. The ECA

cell loss prediction is accurate for small output buffer sizes capable of queueing the cell level fluctuations, but too pessimistic for large buffer sizes. The (M+1)-MMDP approximation is accurate for connections with a burst scale type of behaviour, but due to the computational effort involved in the cell loss calculations, is not adequate for real time CAC response.

FCAC cell loss prediction was also compared with another fuzzy based CAC approach that uses neural networks to create the fuzzy rule base, NFCAC. The cell loss predictions given by FCAC and NFCAC conform to the cell loss value obtained by simulation. FCAC requires more training data than NFCAC to achieve the same accuracy in the cell loss prediction, because FCAC considers not only the influence on the cell loss of the variations in transmission rates but also the influence of the duration of the traffic bursts.

FCAC is recommended for network configurations having large output buffer sizes that can queue not only cells resulting from asynchronous arrivals but also the whole or part of the burst. Thus, FCAC is very useful for ATM services sensitive to cell loss, such as file transfer and image retrieval.

Contents

List of Figures

List of Tables

Acknowledgements

The author gratefully acknowledges support from the "Centro de Estudos de Telecomunicações" - Telecom Portugal and "Junta Nacional de Investigação Científica e Tecnológica" as well as Prof. L.Cuthbert, Prof. E.Mamdani, Mr. V.Lagarto and in particular to Dr. E.M.Scharf, who kindly supervised this research, for all the support done during these studies. Further acknowledgements go to all my research colleagues namely Dr. J.Pitts and Dr. J.Schormans for their precious advice on queueing theory, Dr. S.Parsons for his patience in reading and commenting chapter 3, Mr. M. Fontaine Mr. J.Lazaro and Mr. T.Yang for their generosity in allowing to publish their research results and Dr. P.Fonseca and Mr. S.Winstanley for the discussions on ATM traffic control.

Abbreviations

AI	Artificial Intelligence
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband Integrated Services Digital Network
CAC	Connection Admission Control
CBR	Constant Bit Rate
CPN	Customer Premises Network
CDV	Cell Delay Variation
CLR	Cell Loss Ratio
EP	Evolutionary Programming
ES	Evolution Strategies
FCAC	Fuzzy Connection Admission Control
FL	Fuzzy Logic
FLS	Fuzzy Logic based Systems
GA	Genetic Algorithms
GMDP	General Modulated Deterministic Process
ISDN	Integrated Systems Digital Network
ITU	International Telecommunication Union
LAN	Local Area Networks
LEX	Local Exchange
MMDP	Markov Modulated Deterministic Process
QoS	Quality of Service
RACE	Research into Advanced Communications for Europe
STM	Synchronous Transfer Mode
UPC	Usage Parameter Control
VBR	Variable Bit Rate
VC/VP	Virtual Channel/Path
VCI/VPI	Virtual Channel Identifier/Path Identifier

Errata

page 22, ***line 13***, where it reads "according to a deterministic arrival process instead of a deterministic arrival process", should be read "according to a deterministic arrival process instead of an exponential arrival process".

page 23, ***line 5***, where it reads "The parameters A, α and β can be easily obtained" should be read "The parameters A (bit/sec), α (sec) and β (sec) can be easily obtained".

page 29, ***line 5 counting from the end of the page***, where it reads "the NP decreases" should be read "the network performance (NP) decreases".

page 35, ***line 10 counting from the end of the page***, where it reads "the traffic load is not underestimated if π and μ are to be replaced by R_p and R_m" should be read "the traffic load is not underestimated if p and m are to be replaced by R_p and R_m".

page 40, ***line 2 counting from the end of the page***, where it reads "Depending on the number of different traffic types to be considered and the number of states of each type" should be read "Depending on the number of different traffic types to be considered and the number of rate states of each type".

page 43, ***line 3***, where it reads "the results may be close to the peak allocation scheme." should be read "the results given by the linear CAC may be close to the peak allocation scheme."

page 46, ***line 9 counting from the end of the page***, where it reads "but can also be a generalisation of the same data for similar traffic patterns." should be read "but can also generalise its response for similar traffic patterns."

page 50, ***line 8***, where it reads "if the burst size is long" should be read "if the burst size is such that can be partially or totally queued at ATM buffers".

page 57, ***line 3***, where it reads "$\{x \in U: F(x) \geq 0\}$" should be read "$\{x \in U: F(x) > 0\}$".

page 59, ***line 11***, where it reads "The fuzzy knowledge base contains is a set of fuzzy rules" should be read "The fuzzy knowledge base is a set of fuzzy rules".

page 63, ***line 5***, where it reads "When used for monotonic membership functions, CAC is also referred as" should be read "When used for monotonic membership functions, Centre of Area (COA) is also referred as".

page 63, ***line 9***, where it reads "$w_i = \min(A_i(x_0), B_i(x_0))$" should be read "$w_i = \min(A_i(x_0), B_i(y_0))$".

page 67, ***line 11***, where it reads "The structure identification of a fuzzy model consists of the input variable identification and the rule identification. The input variable identification consists of" should be read ""The structure identification of a fuzzy model consists of the rule identification and the input variable identification. The rule identification consists of".

page 72, ***line 5***, where it reads "$A = (A_1, A_2, ..., A_M)$" should be read "$A_i = (A_{i1}, A_{i2}, ..., A_{iM})$".

page 73, ***line 9***, FORMULA 3.15

page 80, ***line 7***,

page 83, ***line 12***, where it reads "Choquet capacities of order two", should be read "Choquet capacities of order two [Lam, 1989]".

page 87, ***line 12***, where it reads "the fuzzy sets B and $\neg B$ exhaustive (i.e. $B * \neg B = U_2$); more precisely, the bounded sum t-conorm operator, defined as: $a \oplus b = a + b - ab$. The fuzzy inference result obtained using the Choquet integral based on the sum t-conorm operator (see [Camp, 1993, pp.150 for details), yields:" should be read "the fuzzy sets B and $\neg B$ exhaustive, for instance using a Choquet integral in which the maximum operator is replaced by the bounded sum t-conorm (see [Camp, 1993], pp. 150 for details). The fuzzy inference result is:”

page 87, ***figure 3.8***, where it reads “⊕ is the bounded sum t-conorm” should be read “⊕ is the probabilistic sum t-conorm ($a \oplus b = a + b - ab$)”.

page 88, ***line 5 counting from the end***, where it reads “considering as the t-conorm the bounded sum t-conorm and “ should be read “considering as the t-conorm the probabilistic sum t-conorm ($a \oplus b = a + b - ab$) and “.

page 92, ***line 5***, where it reads “but because $B_1^*(s) \geq 0$ and $B_2^*(s) \geq 0, \forall s$”, should be read “but because $B_1^*(s) + B_2^*(s) - 1 \geq 0, \forall s$”.

page 97, ***line 6***, where it reads “will be applied in chapter 6” should be read “will be applied in chapter 5”

page 97, ***line 9***, where it reads “Before that, chapter 5 presents ” should be read “Before that, chapter 4 presents”.

page 112, ***line 14***, where it reads “GA are different from normal optimisation techniques” should be read “GA are different from traditional optimisation techniques”.

page 120, ***line 5***, where it reads “ $\forall i \in \{1,...,\lambda\}: rank(a_i^t) = i \quad \Leftrightarrow \quad \forall j \in \{1,...,\lambda - 1\}: f(a_j^t) \lozenge f(a_{j+1}^t)$ ”, should be read “ $\forall i \in \{1,...,\lambda\}: rank(a_i^t) = i \quad \Leftrightarrow \quad \forall j \in \{1,...,\lambda - i\}: f(a_j^t) \lozenge f(a_{j+1}^t)$ ”.

page 129, ***line 5 counting from the end***, where it reads “assigning *ey* to this value” should be read “assigning *ey* to this value, for example if *y* is 10^{-5} then *ey* is assigned the value 5”.

page 149, ***line 5***, where it reads “The learning method based on GA presented in this chapter fixes the number of fuzzy variables” should be read “The learning method based on GA presented in this chapter fixes only the number of fuzzy variables”.

1 Introduction

1.1 ATM for Broadband Communications Networks

Asynchronous Transfer Mode (ATM) is the basis for future high-speed telecommunications networks, namely the Broadband Integrated Services Digital Network (B-ISDN), [ITU I121, 1991]. B-ISDN provides end-to-end digital connectivity to support a wide range of services, including voice and non-voice services ([ITU I121, 1991]). Hitherto, networks have been service specific (e.g. telephony over circuit switching networks and data services over packet networks). The principal argument in favour of using ATM as a transfer mode for B-ISDN is precisely its flexibility to integrate on the same physical network existing services and new services to be introduced in the future, given that:

- in ATM, all information to be transferred is packed into fixed-size slots called *cells*;
- the multiplexing and switching of cells is independent of the actual application.

In addition, the asynchronous nature of ATM allows B-ISDN to achieve a higher utilisation of the bandwidth than using a synchronous transfer mode (STM). This is because, in ATM, each service can use the capacity of the links, switches and buffers according to the actual demand. Hence, if temporarily one of the connections has no information to be transmitted, the network resources can be used by other connections. This is known as statistical multiplexing and is important because many of the B-ISDN services, particularly video services, are variable bit rate (VBR) services, where the activity level of the source varies over time.

In ATM, the competition for limited network resources (link capacity and output buffer size) among the multiplexed connections can cause, in overload conditions, a decrease in the quality of service (QoS), expressed in terms of cell loss and delay statistics. Therefore, a traffic control policy needs to be introduced in order to obtain a reasonable trade-off between the QoS required by the user and the maximisation of the utilisation of the network resources.

Among the traffic control functions, Connection Admission Control (CAC) plays an important role as it determines the admittance or refusal of a new connection into the network. In order to do this, the CAC algorithm takes into account the traffic characteristics and QoS requirements of the new connection and those of the connections already established. Because ATM-based networks can integrate any service (with fixed or variable bandwidth rates), the CAC task is a complex one.

The CAC decision-making process relies on an accurate knowledge about the traffic behaviour of the sources being multiplexed into an ATM link. Traffic source models help in the analysis of the traffic generated by a particular source and give a first estimation of the amount of network resources each source requires. A second and more accurate estimation is given by the traffic model of the superposed traffic, that is, the model that describes the aggregate traffic behaviour resulting from multiplexing several sources on the same ATM link. By relating the superposed traffic characteristics to the available network resources, a prediction can be obtained for the expected QoS. CAC decides to admit a new connection if the expected QoS conforms with the required QoS; otherwise, the new connection request is rejected.

The traffic models available in the literature for source characterisation have been derived by analysing on-line measurements of statistical traffic parameters for voice, image and data traffic; for example, Leland et. al ([Lel, 1993]) analysed traffic generated by Ethernet Local Area Networks (LANs), Maglaris et al. (see [Magl, 1988]) studied the traffic generated by digital packetised video in packet networks (see also [Rama, 1990]) and Brady ([Brad, 1969]) studied N-ISDN voice sources. The specific properties of ATM (fixed size cells and bandwidth on demand) require statistical models different from those used for traffic in existing circuit or packet switched networks. ATM source models need also to be accurate, simple, and generally applicable. A traffic model that satisfies the previously mentioned requirements and takes into account the characteristics of data, image and voice sources given on the published traffic studies represents a big challenge for research studies.

In addition, CAC has to look for an optimisation in the distribution of the bandwidth resources in order to achieve a maximum statistical multiplexing gain. The principle of statistical multiplexing is that not all the sources need to transmit at their peak rates at all times. This means that more sources than the number given from the sum of the peak bit rates and available bandwidth can be carried, provided that the QoS requirements can still be satisfied.

1.2 Research challenge

The theoretical traffic model of the traffic process resulting from multiplexing several traffic sources in the same ATM link (assuming that the traffic model of each of the sources is known) is too complex to be used in a real time CAC application. In order to overcome this, several authors (see [Kilk, 1994], [Saito, 1992], [Ivers, 1987], [Hui, 1988]) have proposed to divide the multiplexed traffic process into several layers, each corresponding to a different time scale and each having a different influence on the expected QoS. The CAC approaches based on this layered model of the multiplexed traffic take into consideration the effects of some (not all) of the traffic layers on the QoS and, thus, can be either pessimistic or optimistic if the effects of the traffic layer(s) not considered happen to be influential.

Other CAC approaches are based on approximations made in the traffic models in order to achieve numerical tractability, but these approximations have an effect on the accuracy of the QoS prediction. For example, the convolution algorithm is based on a bufferless model and, therefore, gives a pessimistic cell loss ratio prediction for sources with small burst sizes that can be partially queued in ATM buffers.

Considering the limitations of a theoretically based CAC approach referred in the previous paragraphs, the research presented in the following chapters has explored the use of *fuzzy techniques*, in preference to analytically based approaches, to obtain an *empirical* model for the relation between the QoS to be expected for each of the multiplexed connections and the statistical parameters that characterise the multiplexed traffic. The QoS parameter considered is the cell loss ratio per connection. The reason for choosing the CLR in preference to QoS parameters, such as the maximum cell delay per connection, is that the CLR is the simple, single parameter that applies to both short and large buffer sizes.

The application of fuzzy logic based techniques to CAC constitutes a new research topic ([Ram, 1994]). A parallel research by Fontaine et al ([Font, 1996]) has also proposed a fuzzy logic based approach to CAC but this approach uses is a hybrid fuzzy-neural system. The cell loss ratio predictions obtained using each of these fuzzy logic CAC approaches will be compared and commented in chapter 6.

According to Zadeh ([Zad, 1993]), “the principal aim of fuzzy logic based techniques is to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness and low solution cost”. In the same line of thought, the fuzzy-logic based CAC (FCAC) proposed in this thesis, will exploit the tolerance for imprecision in the characterisation of the multiplexed traffic scenario in order to achieve a numerically tractable solution that predicts the effects on cell loss when several connections are multiplexed on an ATM link. The traffic model embedded in the fuzzy rule base can also express the non-linearity between the load in the ATM link and the maximum cell loss ratio per connection, observed in non-homogeneous traffic scenarios.

The FCAC performs the decision-making of admitting or rejecting a new connection into an ATM network by predicting the maximum cell loss ratio per connection when a candidate connection is added to a background traffic scenario (made up of the connections already admitted into the network). If the cell loss ratio predicted by FCAC does not violate the cell loss ratio requirements of the existing connections and of the candidate connection, the new connection is admitted; otherwise it is rejected.

The FCAC uses the user declared statistical parameters for each of the connections (including the incoming connection) on a node-to-node basis. The empirical knowledge is derived from on-line measurements collected at each ATM link for different traffic patterns. The measurements comprise the mean and peak bit rates, mean burst length per connection as well as the monitored cell loss ratio for each of the multiplexed connections.

1.3 Research proposal

The most difficult task in developing fuzzy logic systems is the identification of the fuzzy logic system, that is, the determination of the number and structure of the fuzzy rules, as well as the definition of the fuzzy sets for each of the fuzzy variables. Particularly, the definition of the fuzzy sets that allow the fuzzy logic system to obtain the optimum performance, is the most time consuming phase. Changes in the shape used for the definition of the fuzzy sets affect the performance of the fuzzy logic system, because the contributions each one of the rules makes to the final decision also changes. Among others, Procyk and Mamdani [Mam, 1978] introduced an iterative procedure for modification of the fuzzy sets of the domain of each fuzzy variable and Tanscheit and Scharf ([Tans, 1988]) proposed a self-organising fuzzy logic controller for robotics application, but in general, the design of fuzzy logic systems has been a trial and error process in which most of the development time is devoted to the "quest" for efficient fuzzy sets.

More recently, several automatic design methods have been proposed. These methods have mainly been based on neural networks ([Ichik, 1992], [Jang, 1992], [Tak, 1991]), gradient methods ([Arak, 1992], [Katay, 1992]) or genetic algorithms ([Karr, 1992], [Nish, 1992], [Nom, 1992]). Automatic design methods offer an improvement to the development of fuzzy logic systems over the traditional "trial-and-error" process. This is because the definition of the fuzzy sets and the fuzzy rules becomes the outcome of an automated process. The same automated process can also be used after the design phase to adapt the fuzzy logic system to a dynamic environment, by tuning the fuzzy sets for each of the rules.

This research has explored genetic algorithms in the automatic design of the fuzzy rule base of FCAC; the fuzzy rules and the definition of the fuzzy sets for each of the rules are obtained using a learning algorithm composed of three phases (generation, simplification and tuning), each of these, based on genetic algorithms. The application of this learning algorithm to the automatic design a fuzzy decision support system to CAC is a new research work and has been based on the research developed by Herrera et al ([Herr, 1995]).

Taking into account that the traffic patterns observed in an ATM link are expected to vary over time (new services are introduced, existing services might become obsolete), the *adaptation* of FCAC to changes in the traffic patterns is achieved using a tuning process, based on a genetic algorithm. The tuning phase is invoked either periodically or when a certain network performance threshold is reached in order to improve the accuracy with which the FCAC predicts the effect of a new connection on the ATM traffic already in the link.

Once a decision has been made to use (1) a fuzzy rule base for representing knowledge about the ATM traffic and (2) genetic algorithms in order to automate the design of this rule base, the next challenge is how to *acquire knowledge* based on the relationship between the multiplexed traffic characteristics and the observed cell loss ratio. In traditional fuzzy logic systems, this heuristic knowledge would be obtained by interviewing experts in the area and would be used to set up the fuzzy rule base. In the CAC case, this was not possible because the only heuristic knowledge that exists on ATM traffic was obtained by monitoring the traffic statistical parameters (e.g. mean and peak bit rates) of the multiplexed connections and the corresponding cell loss ratio for each connection, using on-line measurements obtained from an ATM test-bed and results generated with ATM traffic simulators. This led to the adoption of a method of learning from examples, the examples being the above mentioned data measurements.

1.4 Organisation of thesis

This thesis describes a fuzzy logic system for the prediction of the maximum cell loss ratio expected per connection, when a new connection is added to an ATM traffic mix made up of existing connections. This chapter has introduced the main contribution of the thesis to the topic of traffic control, namely to the CAC function in an ATM network.

In chapter 2 a more detailed review of ATM related issues is presented. The focus will be made mainly on topics related to CAC, such as traffic control, source characterisation (traffic models) and statistical multiplexing gain. This will provide an understanding of the application area and of the CAC algorithms (analytical and heuristic) proposed in the literature.

Chapter 3 introduces the theoretical background for the development of fuzzy logic systems. Initially, the definitions and terminology used in this thesis are presented. After that, the method for identification of fuzzy logic systems from examples is presented, followed by the presentation of the fuzzy inference model for rules with an associated uncertainty interval. The uncertainty interval asserts the lower and upper bounds of uncertainty contained in the rule about the statement: "the rule is a *true* rule for the system", where the truthfulness of the rule is related to its ability to describe the knowledge implicitly expressed in a set of examples.

Chapter 4 presents the architecture of the fuzzy logic based CAC (FCAC) proposed in this thesis. The input and output fuzzy variables considered in the antecedent and consequent of the fuzzy rules are described, as well as the format of the fuzzy rules selected and the fuzzy sets chosen for the universe of discourse of the fuzzy variables.

In Chapter 5 the three-phase algorithm for the automatic design of the fuzzy rule set, based on genetic algorithms, is presented and the characteristics of the genetic algorithm used in each of the phases of the algorithm are explained. This is preceded by a brief introduction to genetic algorithms.

Chapter 6 shows the results obtained when comparing FCAC predictions with cell loss ratio predictions using (1) theoretically based CAC approaches and (2) a fuzzy logic based CAC approach that uses neural nets instead of genetic algorithms for the automatic setting up of the fuzzy rule base. The traffic mixes studied are composed of “On-Off” traffic sources of the same type (homogeneous traffic mixes) and different types (heterogeneous traffic mixes). A network simulator ([Pitts, 1993]) and on-line measurements obtained from an ATM test-bed ([EXPLOIT, 1994]) were used to validate the cell loss ratio predictions. The experimental scenarios were chosen in order to show both (1) the areas where the prediction of the FCAC has to be accurate and (2) the areas where the FCAC prediction has to be backed up by a more conservative approach such as peak rate allocation.

Chapter 7 discusses the results obtained in chapter 6 and suggests further research topics which are a result of the limitations due to lack of knowledge of real ATM traffic sources faced during the research and the effect that such limitations can have on the performance of the FCAC. Chapter 8 presents the main conclusions reached in the research and summarises the application of a fuzzy logic based approach to CAC, focusing on the advantages and limitations presented in chapter 7.

2 Connection Admission Control in ATM Networks

One of the major advantages of Broadband Integrated Services Digital Network (B-ISDN) based on the Asynchronous Transfer Mode (ATM) is its flexibility at high transmission rates. Such a network allows the integration of a large variety of services having different stochastic features (such as the mean and the variance of the number of arrivals in a time interval) and different Quality of Service (QoS) requirements defined in terms of cell loss, cell delay and cell delay variation ([Cuth, 1993]).

Traffic management of an ATM-based network requires the definition of a set of control algorithms in order to be able to achieve a high network utilisation, maintaining the network performance. Connection Admission Control (CAC) is one of the purposed ([ITU I371, 1995]) traffic control functions and it decides whether or not to admit a new connection in the network.

CAC admits a new connection if there are available network resources to provide the QoS required by a new connection and by the already established connections. In order to fulfil these requirements, CAC needs to obtain an estimation of how the network performance decreases, in terms of cell loss, cell delay and cell delay variation, when a new connection is added to a background traffic scenario, consisting of the multiplexed traffic from accepted connections. CAC second aim is to look for an optimisation in the allocation of network resources, to take full advantage of the facilities provided by the ATM network.

In the following, a review of concepts considered when defining CAC in an ATM context is presented:

- section 2.1 presents an overview on ATM comparing it with present transfer modes, such as circuit switching, message switching and packet switching. Furthermore, concepts such as statistical multiplexing and statistical multiplexing gain are revised and its relation with CAC explained. The network performance

(NP) parameters for ATM-based networks are presented and comments are given on the choice of cell loss for the NP parameter considered for CAC purposes.

- section 2.2 considers ATM traffic behaviour at several time scales and presents the set of parameters chosen, in this study, for describing the traffic behaviour of the source; this set of parameters is referred in the literature as the traffic descriptor. After that, a description of the On-Off model as the traffic model adopted in this study is presented, followed by the identification of the On-Off model parameters using the chosen traffic descriptor.
- section 2.3 defines traffic control in ATM networks and introduces the traffic control functions which are related to this thesis: CAC and Usage Parameter Control (UPC). The relationship between CAC and UPC is explained, focusing on how CAC relies on UPC to guarantee that the user declared values for the traffic descriptor parameters (presented to the network at connection set-up) are being respected.
- section 2.4 presents CAC algorithms proposed in the literature to solve the CAC task, more precisely the ones that are most relevant for comparisons with the FCAC approach (see [Hui, 1988], [Ivers, 1989], [Klein, 1990], [Louv, 1988], [Mur, 1991], [Wors, 1992]).

2.1 Asynchronous Transfer Mode Networks

The transfer mode is a specific way of transmitting and switching information in a telecommunications network. More precisely, it defines how the information supplied by network users is eventually mapped onto the physical network: this includes the technique for transmission, multiplexing, and switching aspects of communications networks.

ATM networks flexibility is achieved by providing services with a common transfer capability due to its independence of the bit rate and data structure of the services carried ([Cuth, 1993]). In ATM, all types of information (voice, data, video and still picture) are presented in the same form using equal-sized packets comprising a 5 byte header and 48 byte payload, named *cells*. The cell header contains a label, called a virtual channel identifier/virtual path identifier (VCI/VPI), denoting the routeing address. The cell labels are used in switching and multiplexing. The payload contains service specific header and tail, plus the actual information that the user wants to transmit across the network.

ATM has been recognised to become the solution for the implementation of B-ISDN by several authors in the literature (see [Hand, 1989], [Kawa, 1988], [Turn, 1986]). B-ISDN implies a multi-service environment where applications will have different service requirements (e.g. voice applications are delay sensitive and data applications are sensitive to cell losses). The flexibility of the ATM-based B-ISDN access, resulting from the cell transport concept, allows the coexistence on a same physical network of a wide variety of services and applications and the introduction of new services.

ATM is inherently an asynchronous mechanism which allows connections to send information whenever they are ready to do so, after an initial connection set-up negotiation has taken place. Given that B-ISDN will accommodate not only constant bit rate (CBR) connections (bit rate is constant) but also variable bit rate (VBR) connections (bit rate varies over time), ATM allows a more efficient utilisation of the network resources than a dedicated channel allocation policy such as circuit switching.

ATM provides virtual circuit switching as opposed to fixed circuit switching. This has the advantage of using connection oriented routeing techniques from circuit switched networks but without having to establish a dedicated circuit (with fixed bandwidth) between the two ends of the communication for the whole duration of the connection.

In message switching networks the wastage of resources for VBR traffic is overcome by treating each information unit as a logical entity (referred to as a message) that is transmitted in the network independently of other units. This can be achieved by adding a header to each message that defines the destination node. The main disadvantage of this transfer mode is that it is not well suited for real time or delay sensitive applications, since the delay caused by header decoding at each intermediate node is not predictable.

Another candidate transfer mode for B-ISDN is the packet switching mode. Packet switching combines the advantages of both circuit switching and message switching. According to Onvural ([Onvur, 1994]), the main disadvantage of packet switching over message switching is that transmitting a message requires more overhead (i.e. packet header) per message than the correspondent overhead in message switching, thereby reducing effective resource utilisation in the network.

The asynchronous transfer mode (ATM) is similar to virtual circuit packet switching if packets are assumed to have the same size. ATM has advantages over both the circuit mode and the packet mode. The self-routeing ATM switch uses the advantages of circuit-switched networks allowing to provide high-speed communication transfer after connection set-up. The simplification of functions within the core network reduces the number of software driven procedures. This overcomes the disadvantage of the packet mode and allows broadband communications whilst allowing an arbitrary bandwidth to be allocated [Saito, 1994].

The main challenges introduced by fast-packet switching with variable-length packets were the complexity of the switching fabrics and the buffer management scheme as the transmission speed increases. ATM answers this by using fixed sized packets and, therefore, is able to support real-time applications together with data applications in the same network environment.

2.1.1 Statistical Multiplexing

When several ATM connections share common resources through multiplexing, a more efficient usage of the resources is obtained compared with the fixed amount of resources allocated without multiplexing. This phenomenon is due to the statistical nature of the traffic streams and the resource sharing is called *statistical multiplexing*. The statistical multiplexing is quantified by means of the *multiplexing gain*, defined in [R1022 D.126, 1992] as the ratio of the link utilisation obtained when the highest possible number of connections is multiplexed without violation of their Quality of Service (QoS) requirements and the link rate utilisation obtained by peak allocation (deterministic multiplexing). The utilisation previously referred can be measured in terms of the number of connections admitted into the network (see figure 2.1).

Considering the above definition, the multiplexing gain is a function of :

- the characteristics of the multiplexed sources: the bandwidth requirements such as peak and mean bit rates and space requirements expressed in terms of the mean burst length;
- the network resources (e.g. link capacity, output buffer size);
- the user required QoS (expressed in terms of parameters such as cell loss, cell delay, cell delay variation).

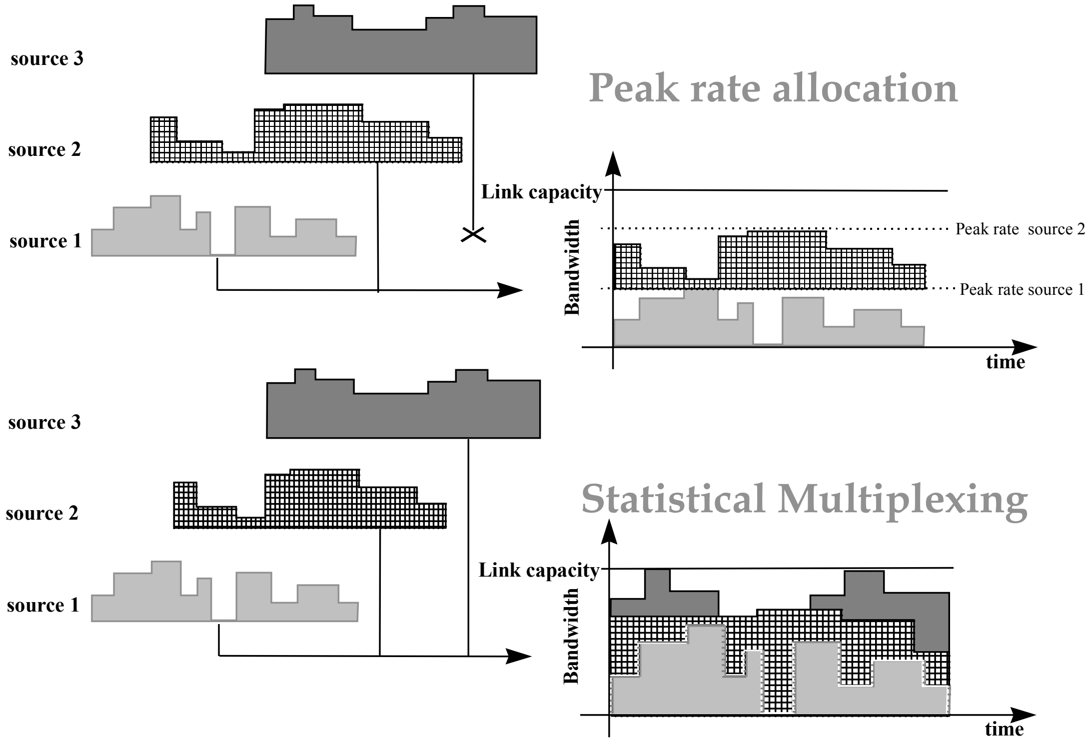

Figure 2.1 Statistical Multiplexing and Peak Rate bandwidth allocation.

Exploitation of statistical multiplexing allows a more efficient utilisation of the network resources under traffic loads that vary over time. On the other hand, statistical multiplexing poses challenging problems because it requires the introduction of appropriate traffic control procedures in order to support the QoS required by the services using the network. Therefore, traffic control in ATM networks requires a basic understanding of the statistical multiplexing effects on the QoS of the multiplexed traffic streams. This understanding is also necessary for the definition of the network performance objectives and traffic descriptors.

2.1.2 Network performance parameters

Network Performance (NP) is defined in the ITU Recommendation I350 ([ITU I350, 1993]) as "the ability of a network or network portion to provide the functions related to communications between users". This thesis is interested in the identification of the network performance parameters that have a significant influence on the user required Quality of Service (QoS). The term QoS is described in CFS D510 ([CFS, 1992]) in the following way: "QoS is the measure of how good a service is, as perceived by the user. It is expressed in user understandable language and manifests itself as a number of parameters, all of which have either subjective or objective values".

According to Saito ([Saito, 1992]):

> “Of the network performance deterioration factors, cell segmentation delay and propagation delay over transmission links are fixed, and are independent of traffic characteristics. Cell loss and misdelivery due to header field errors in transmission are also independent of the traffic. Cell loss and delay in a switch are negligible. Thus, the key factors in ATM network performance deterioration are cell loss and delay in the output queue to a transmission link in ATM nodes.”

Although the QoS requirements of a connection may contain several parameters, the cell loss is important as far as connection admission is concerned. The cell delay and cell delay variation depend mostly on the network properties, especially on the storage capacity of the output buffer and on the length of transmission links. Cell delay variation is a measure of the variation of the end-to-end delay introduced by network elements such as switches and access multiplexers. However, if the following buffer size dimensioning method is used (see also [Saito, 1991a]), the variable portion of the end-to-end delay remains at a level that is not relevant for the QoS of the B-ISDN services:

> “Consider a transmission link of C Mbit/s and its output buffer. The buffer size is assumed to be K. Let T ms be the maximum admissible delay in the output buffer. Assume that the buffer size K is dimensioned such that the maximum delay in the buffer is less that T under a First In First Out (FIFO) discipline. That is, the output buffer size K and the maximum admissible delay T are assumed to satisfy the relationship
>
> $$K = 1000TC / L$$
>
> where L is the cell length in bits.”([Saito, 1992])

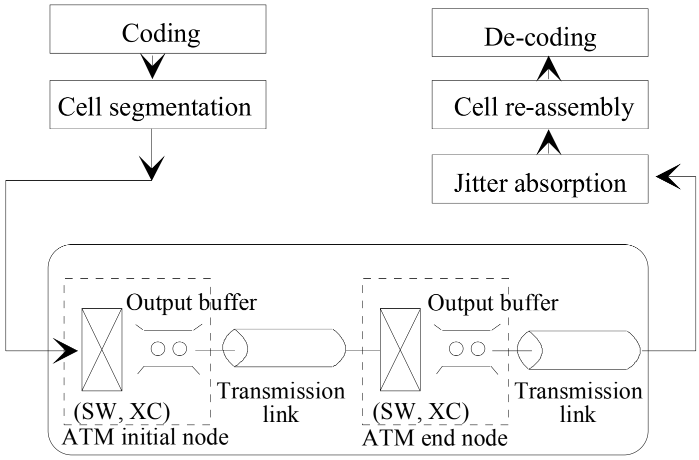

Figure 2.2 Simplified model of an ATM network focusing on the processing in the terminal nodes (from [Saito, 1991a]).

Studies presented by Blondia and Casals in ([Blond, 1992]) indicate that the probability but not the correlation of cell losses can be reduced by enlarging the network buffers. However, according to Jacobsen ([Jac, 1990]): "the increase in network efficiency that can be obtained by increasing the buffer size is small". Jacobsen also refers that this is because large buffers introduce more cell delay variation and changes in the characteristics of the cell streams.

Considering all the previously stated, this research has adopted the cell loss measured at the output buffer to the transmission link as the QoS parameter for connection admittance purposes.

2.2 ATM Traffic

The characterisation of the cell stream, generated by ATM traffic sources has been studied by many researchers over the past years. The proposals for source (traffic) models put the emphasis on simplicity, realism and accuracy. Traffic models are used to evaluate the performance of CAC and UPC algorithms.

Traffic in ATM networks is mainly of two types: Constant Bit Rate (CBR) and Variable Bit Rate (VBR). CBR connections are deterministic connections in the sense that the bit rate is fixed over time. VBR connections exhibit a variable bit rate over time and are one of the most predominant connection types in ATM Networks.

Table 2.1: ATM traffic classes as described in [Bon, 1995].

Traffic Class	**Main Characteristics**
CBR, VBR	Fixed (possibly re-negotiable) Traffic Descriptor; QoS guaranteed by CAC; Traffic Descriptor parameters policed by UPC.
ABR	The source cannot describe completely its traffic characteristics but provides some minimum traffic requirements (e.g., Minimum Cell Rate); Source “dynamically adapts” to feedback from congestion control mechanism; Requires low loss for cells “conforming” to correct dynamic behaviour.
UBR	No traffic description nor QoS guarantee; No call admission nor policing.

The classification adopted in this thesis for QoS guaranteed ATM traffic is resumed to two traffic types, VBR and CBR, aiming at a simplification of the description of the ATM traffic inherent characteristics and not at providing an exhaustive list of all the possible traffic classes identified in the literature (see also table 2.1). Recently, two other traffic classes have been specified, namely the Available Bit Rate (ABR) and Unspecified Bit Rate (UBR), see also [Bon, 1995], and [Berg, 1995] for details. The justification behind the existence of these traffic types is that, in order to guarantee QoS levels for traffic such as VBR and CBR, the network may have to be under utilised. Hence, the bandwidth unused by guaranteed traffic can be available to “best-effort” traffic such as ABR and UBR. A problem arises in how to share the bandwidth available to the best-effort service dynamically. A solution suggested by [Bon, 1995] queues QoS guaranteed traffic (CBR and VBR) and best-effort traffic (ABR and UBR) in different output buffers and best-effort traffic is served when there is no guaranteed QoS traffic ready to be served.

This thesis focus on the influence of multiplexing traffic comprising mainly of CBR and VBR traffic on the monitored cell loss for CAC purposes. This is because ABR and UBR do not require a guaranteed QoS, and, therefore, CAC methods do not apply. CBR traffic, due to its deterministic statistical behaviour, does not allow CAC methods to contribute for a high statistical multiplexing gain and, therefore, is considered in this thesis just to illustrate a bandwidth consumption service.

In the next section, models for analysing the traffic generated by VBR connections and the traffic parameters chosen to define VBR traffic characteristics are going to be identified and explained in detail.

2.2.1 Traffic Parameters

The traffic parameters are the parameters that identify the traffic behaviour of an ATM source. These parameters are used by traffic control mechanisms and should fulfil the following requirements ([ITU I371,1995]):

- the parameters should be *simple* such that they can be declared by the user-end or his terminal and be *enforceable* by the Usage Parameter Control (UPC) and Network Parameter Control (NPC) functions (source policing functions, see section 2.3.2 for more details);
- the parameters should be *relevant* for resource allocation schemes meeting network performance requirements.

The *traffic descriptor* of an ATM source is a vector specifying the actual values for the traffic parameters. The traffic descriptor plays a key role in the service provisioning, admission control, and UPC of the network, because it:

- constitutes the basis of the service contract between the user and the network;
- is the basis for the network decision on call connection attempts;
- is the basis for judging a user's conformance with the service contract.

There has been some discussion on traffic descriptors in the literature and standards' forums (see also [ITU I371, 1995], [ATM-F, 1993], [Eck, 1991]); the discussion is far from over and there is no consensus on the exact parameter set to be included in the traffic descriptor. The difficulties in the choice of a traffic descriptor set are due to the trade-off between the exploitation of multiplexing gain and the complexity of the traffic control schemes, not to mention the complexity of its enforcement by a policing function.

Vakil and Saito ([Vak, 1991]) proposed a traffic descriptor including the peak rate, mean rate, maximum burst length and mean burst length. These parameters are defined as follows:

- the *peak rate* is the inverse of the minimum time between the transmission of two successive cells; if this rate is measured in Mbit per second is referred as *peak bit rate*, if measured in cells per second as *peak cell rate*;
- the *mean rate* is the average number of cells that the source transmits during an interval T (see 2.3.2.2 for more details on the criteria for the choice of T) and similar comments as before apply regarding the measurement units;
- the *maximum burst length* is the maximum number of consecutive cells that the source could transmit at its peak rate during an active period;
- the *mean burst length* is the average number of consecutive cells that the source transmits at its peak rate during an active period.

Commenting on the previously presented traffic parameter set in terms of its relevance for CAC schemes, it can be pointed out that, since bandwidth allocation based on the peak bit rate excludes any multiplexing gain potential, then at least peak and mean rate should be taken into consideration. However, mean and peak rates allow an estimation of the traffic behaviour in terms of bandwidth consumption (time resource), but give no information in terms of the output buffer occupation (space resource). This is why, in this research, the traffic parameter set includes not only the mean and peak rates, but also the mean burst length of a traffic connection. This choice for the parameter set has also to do with the feasibility of the enforcement of the traffic descriptor values by the policing function (see on section 2.3.2 how this can be achieved).

Recently, the ATM-Forum ([ATM-F, 1993]) suggested the inclusion in the ATM traffic descriptor of the mean rate, peak rate and maximum burst length. In this research, the mean burst length is preferred to the maximum burst length because the latter is associated with a worst case buffer occupation. Only in case the user cannot specify a value for the mean burst length, but can do so for the maximum burst length value, this would be the value used for the quantisation of the requirements in terms of buffer space occupation.

Other authors ([Andr, 1994], [Andr, 1991], [Heff, 1986]) have suggested the inclusion in the traffic descriptor of the Index of Dispersion for Counts (IDC), defined for a time interval of length t, as the variance of the number of arrivals in an arbitrary time interval of length t divided by the mean number of arrivals in the same arbitrary interval. This thesis does not consider this traffic parameter, because it is quite difficult for the user to specify the source's IDC value. Nevertheless, if this parameter can be obtained via on-line traffic measurements, it will provide a more accurate correlation measure than an estimation based on the mean burst length.

Summarising, the chosen traffic parameters to describe a traffic source (connection) are inspired in Rasmussen's approach ([Rasm, 1991]) and comprise the following:

- peak rate (R_p) measured over a short period T_s of time;

- mean rate (R_m) measured over a longer period T_l of time;
- mean burst length (B_m).

The value for T_s is chosen such that the source is strictly peak rate policed; it is assumed that the minimum duration of a connection is T_l. The mean burst length can be obtained from R_p, R_m and T_l in case the user cannot specify this parameter (see section 2.3.2.4 for more details).

2.2.2 The On-Off Model

The On-Off model is a versatile source model and has been shown by several studies to be an adequate traffic model for ATM traffic sources; namely:

- Maglaris et al. ([Magl, 1988]) showed that a superposition of N (N>1) video sources can be modelled by M (M=20×N) independent On-Off sources, each alternating between transmitting 0 bits/pixel and A bits/pixel according to a Bernoulli distribution.
- Heffes and Lucantoni ([Heff, 1986]) showed that the statistical multiplexing of M voice sources with speech activity detection can be analysed with a superposition of M sources each modelled with an On-Off model, where the On state represents a talkspurt and the Off state the silence periods.
- Andrade et al ([Andr, 1994]) presented a study on the characterisation of Local Area Networks (LAN) traffic with the Index of Dispersion for Counts (IDC) parameter. LAN traffic measurements made at Bellcore (see [Fowl, 1991]) revealed special characteristics which are not present in traditional arrival processes studied in the literature; namely, the measurements show a long range dependence confirming the opinions on the self-similar nature of LAN traffic ([Lel, 1993]). Andrade et al. matched the parameters of the On-Off model to the IDC values calculated for two time intervals of the LAN generated traffic (a short time interval in order to capture the short term variations of the cell rate, and a longer time interval to capture the rate correlation).

Considering the previously stated, the On-Off model is the model chosen, in this thesis, for describing the traffic behaviour of an ATM source. In the following, the On-Off model will be presented in more detail, emphasising on the calculation of the model parameters using the traffic descriptor presented in section 2.1.1.

2.2.2.1 Model description

The On-Off model is a stochastic process with two possible states: “on”, active state and “off”, inactive state. While in the “on” state, the source generates cells at a constant rate. The transitions between the “on” and “off” states occur with exponential transition rates. The resultant rate $\lambda(t)$ is a continuous time process with discrete jumps at exponentially distributed sojourn times.

There are other types of On-Off source models described in the literature. For example, Fuhrman [Fuhr, 1991] used an On-Off model in which the source generates cells according to a deterministic arrival process instead of a deterministic arrival process. According to Yang ([Yang, 1995]) the model in [Fuhr, 1991] is appropriate for bursty sources whose cell arrivals are more random and less correlated, whereas the model presented above (and adopted in this thesis) is suitable for bursty sources that are highly correlated.

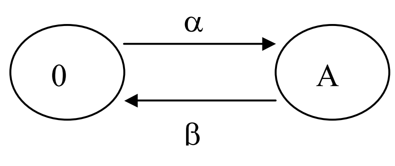

Figure 2.5 The On-Off Model

The On-Off model is identified by the following parameters (see figure below):
- the bit rate A in the active state (no information flow in the inactive state);
- the mean burst duration (also referred as mean on period) $1/\beta$;
- the mean silence duration (also referred as mean off period) $1/\alpha$;

where β and α are the mean transition rate out of the on state and mean arrival rate in the on state, respectively.

2.2.2.2 Parameter identification

The parameters A, α and β can be easily obtained from the traffic descriptor parameters defined in section 2.2.1, namely the mean, m, and peak, p, bit rates (Mbit/s) and the mean burst length, b (cells). If we denote by α‘ (=1/α) the mean off period (sec.), and by β‘ (=1/β) the mean on period (sec.), the percentage of time the source is on, ρ, is given by:

$\rho = \dfrac{m}{p}$ and also $\rho = \dfrac{\beta'}{\alpha' + \beta'}$,

The On-Off model parameters are obtained as follows:

- bit rate in the on state, $A = p$;
- mean on period, $\beta' = L_c \dfrac{b}{p}$,where L_c is the length of an ATM cell (53 octets);
- mean off period, $\alpha' = \beta'\left(\dfrac{p-m}{m}\right)$.

2.2.3 Time resolution for traffic models

The multiplexed traffic observed in ATM networks is very difficult to model stocastically due to its unpredictability over time. In order to simplify its analysis, the traffic process can be divided into several levels corresponding to a specific time scale and with typical traffic characteristics. Three, four or even more time scales have been applied (see [Hui, 1988], [Jacob, 1990], [Saito, 1992]) when dealing with performance analysis in an integrated broadband communication environment. In this thesis, Kilkki’s four time scale model is chosen ([Kilkki, 1994], pp.13). The description of each time scale is as follows:

- *Connection* scale: traffic variations result from variations on the number of established connections (see [Lutt, 1984], [Ivers, 1987] and [Delb, 1981] on how to calculate the blocking probability in the connection scale for different traffic mixes);

- *Rate-variation* scale: the number of established connections is assumed constant but traffic variations are induced by the changes in the demand in terms of cell rate, for example in a VBR video or audio source. This scale covers typically the region from 20 ms to minutes (see [Hui, 1988] for a study on the rate scale using the theory of large deviations and [Lind, 1989] for a similar study using an approximation based on a multi-server loss system);

- *Burst* scale: the inherent phenomenon is the arrival of bursts (arrival process where the inter-arrival time between successive cells is fixed but does not necessarily have to be the minimum declared inter-arrival time), for example the arrival of packets from Local Area Networks (LANs). Each of the connections has a constant inter-arrival time between successive bursts, and the variations occur in the burst duration. This time scale covers typically the region from 0.1 ms to 100 ms (see Yang et al. [Yang, 1995] for a study on the influence of burst level congestion on cell loss using the Markov Modulated Deterministic Process (MMDP) to approximate the arrival process);

- *Cell* scale: each connection is supposed to be deterministic (the inter-arrival time between successive cells is the minimum declared inter-arrival time) and thus the variations in the aggregated traffic process are due to the randomness of phases of different connections. The time scale of these variations is approximately from 1 μs to 1 ms. In cell scale the individual cells and the structures of the switches are taken into account and queuing theory is used to obtain performance data. (see [Hui, 1988], [Ekl, 1988], [Rob, 1991] for discussions on performance at the cell level).

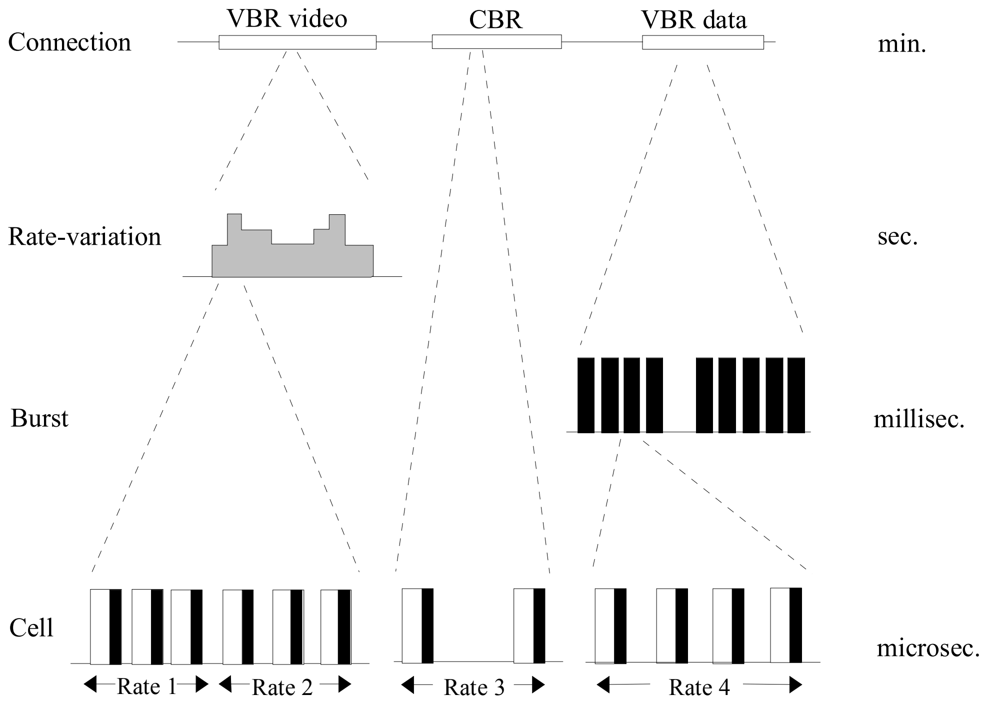

Figure 2.3 Time resolution of ATM traffic

As illustrated in Fig. 2.3, traffic offered from a source, can be observed on connection (or call), rate-variation, burst and cell layers (see also [Gihr, 1991]). In the high-speed ATM environment, the time resolution in these traffic layers differs in order of magnitude, that is, seconds, milliseconds or microseconds for rate, burst and cell layers, respectively. There are no problems to determine and name the connection and cell scales whereas the situation is much more difficult with the intermediate scales: burst and rate-variation scales. According to Kilkki ([Kilkki, 1994]):

> "The names used [...] try to depict the inherent characteristics of a traffic process at each scale. [...] A burst is interpreted as a block of information which has a certain size but not necessary a tight requirement for the peak rate during bursts (except that the transmission of a burst should end before the arrival of the next burst). In contrast, in rate-variation scale there is typically no definite amount of information but a required level for the average cell rate. [...] The time resolution proposed by Aagesen (1993) ([Aag, 1993]) is similar to the [one shown in the figure above]; the most important difference is that the frame scale (from 1 ms to 1 s) and average scale (from 1 s to 1000 s) of Aagesen are united in the rate-variation scale in this study. The reason for this is that if the buffer size is relatively small, as expected in this study, the same traffic models can be applied to the whole region from 10 ms to minutes."

From the CAC point of view, the connection admittance decision that must be made for the connection time scale should be able to guarantee the Quality of Service (QoS) on the rate, burst and cell layer. The CAC studies further reported focus on the traffic behaviour at the burst and rate variation layers. The choice was made in order to study the impact of the burst sizes and the cell rates on the cell loss of the connections being multiplexed in an ATM link.

2.2.4 Queueing influence on network performance

The queue length distribution, obtained for the multiplexed traffic at the output buffer to an ATM transmission link, is difficult to analyse mathematically when cell, burst and rate-variation scale fluctuations are considered. However, several studies ([Norros, 1991], [Rasm, 1991]) have shown that a regular behaviour can be found independently of the actual traffic parameters in each scale (see figure 2.4). Basically, when VBR connections are aggregated on a multiplexer, the queue length distribution is composed of three distinct components. According to Kilkki ([Kilkki, 1994]):

> "The additional rate-variation scale component [...] arises where the input rate is permanently greater than the output rate. The burst scale component [...] is due to the relatively short bursts which can be partly buffered even by the small buffers of ATM nodes. The cell scale component [...] reflects the small queues which occur due to the asynchronous arrival of cells from distinct connections".

There is an essential difference between cell and burst scale components: the cell scale depends on the stationary distribution of the arrival rate whereas the burst component varies with the time dependent stream characteristics ([Rob, 1992]). As stated in [Norros, 1991] and discussed in [Pitts, 1993]:

> "... the delay distribution depends significantly on the cell component, whereas the burst component is the dominant factor in dimensioning the buffer to low cell-loss probabilities. This latter observation arises from the low loss, low delay requirements of ATM (...). The requirement for low delay means that buffer sizes must be small. This in turn means that burst

scale behaviour cannot be adequately buffered and the probability of it occurring must be kept below the low loss requirement".

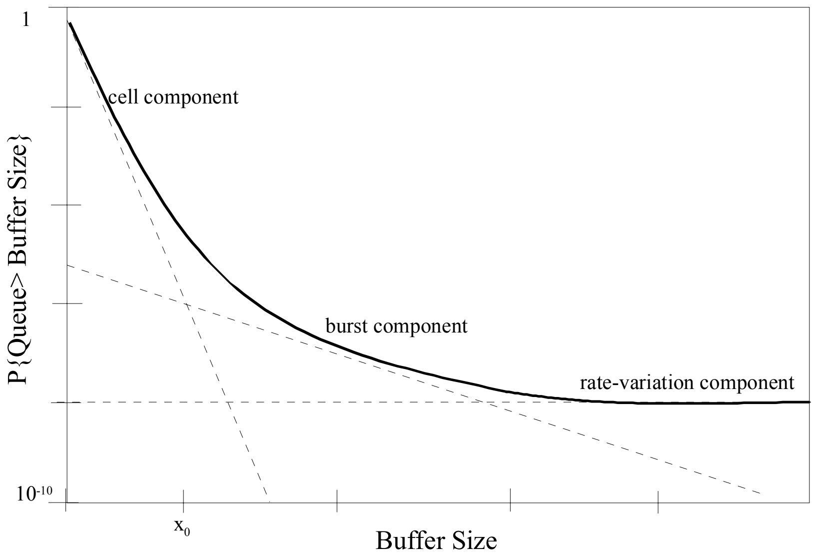

Figure 2.4 Cell, burst and rate-variation scale components of the queue length distribution.

From the previous paragraphs, it can be inferred that the buffer size must be chosen taking into account the contributions of each component on the network performance (cell loss and cell delay). In Rasmussen et al [Rasm, 1991], x_0 in the figure above denotes the buffer size at which the cell scale contribution (approximated by an $M/D/1.r$ queue) equals the burst scale (binomial tail) contribution. Norros et al. ([Norros, 1991]) obtained the same queueing pattern in which the cell component is shown to behave like a $\sum D_i / D / 1$ when the burst component is zero. The burst component was obtained using a fluid flow approximation.

2.3 Traffic Control in ATM Networks

When several traffic streams compete for a common network resource, the competition needs fair and efficient rules. These rules are part of the so-called *traffic control* and *congestion control* described in the ITU recommendations. In [ITU I371, 1995], traffic control is defined as the set of functions taken by the network in all the relevant network elements to *avoid* congestion conditions. Congestion control refers to the actions taken by the network to *minimise* the intensity of congestion effects and to avoid the congestion state *spreading* once congestion has occurred.

Also, in the same recommendation, *congestion* is defined as the state of network elements (e.g. switches, concentrators, transmission links) in which the network is not able to meet the negotiated network performance objectives for the already established connections and/or for the new connection request. Congestion is to be distinguished from the state where buffer overflow is causing cell losses, but negotiated network performance is still achieved.

The primary role of traffic control and resource management parameters and procedures, is to allow the network to achieve the network performance objectives. An additional role, is to optimise the utilisation of network resources. In ATM-based networks, the output buffers are dimensioned to allow the network to cope with simultaneous arrivals of cells from different connections that may require a temporary buffering of cells prior serving. The dimensioning is not meant to take into account the bit rate fluctuations at the burst level obtained when multiplexing variable bit rate (VBR) connections. This implies that if *momentarily* the sum of the bandwidth actually being used by each of the connections exceeds the link capacity and the capacity of the buffers is not enough to queue the excess cells, cell loss will inevitably occur. Therefore, a suitable traffic control strategy that can allow the network operator to obtain a high utilisation of the network resources and at the same time to satisfy the QoS required by the customers is necessary.

Under normal operation (i.e. when no network failures occur) the functions referred to as traffic control functions are intended to avoid network congestion conditions or to minimise congestion effects in order to avoid the congestion. This is also referred in the literature ([Onvur, 1994]) as *preventive* control. However, preventive control techniques are not sufficient to eliminate congestion. Congestion may occur because of malfunctioning of traffic control functions, caused by unpredictable statistical fluctuations of traffic flows or of network failures. Therefore, additional functions referred to as congestion control functions are intended to react to network congestion to minimise its intensity, spread and duration (*reactive* control).

In the following, only preventive control functions are considered, namely the CAC and UPC traffic control functions (see [ITU I371,1995] for more details on other traffic and congestion control functions). The UPC function will be considered because it enforces the values of the traffic parameters specified in the traffic descriptor and upon which the CAC decision is going to be made.

2.3.1 CAC and UPC control functions

Many problems must be overcome in order to use the facilities of ATM networks [Kawash, 1990]; one of these is *Connection Admission Control* (CAC) [Eck, 1989], [Wood, 1988]. In the synchronous transfer mode (STM), when the (peak) bandwidth of a new connection exceeds the remaining capacity of the links, the connection request is rejected. However, in ATM networks, the bandwidth of a connection is not clear, since all the information is segmented into fixed-size cells and just the necessary number of cells is generated and conveyed through the network. As the number of connected calls increases, the NP decreases (the cell loss probability increases). Thus, it is impossible to admit an unlimited number of calls under the service requirements of ATM networks. On the other hand, by limiting the number of connections to a network element and its characteristics (output buffer size, bandwidth capacity), the network load is limited and a suitable QoS is maintained.

To attain high utilisation under the QoS standards, CAC must decide whether to admit a new connection, based on the new connection anticipated traffic characteristics and the QoS requirements of the new and existing connections ([Wood, 1988]). The traffic behaviour of the new connection is estimated from the parameter values specified by the user at connection set-up. Thus, it is very important to ensure that the user declared values are being enforced for the whole duration of the connection. This is the aim of the *Usage Parameter Control* (UPC) function. UPC polices the parameters declared by the user and makes sure that the declared value is being respected. Therefore, CAC and UPC co-operate in the sense that, to make an admittance decision, the CAC function needs to rely on UPC to assume that the traffic parameters used in the decision making are being enforced.

2.3.1.1 What is CAC ?

The Connection Admission Control (CAC) definition given by the ITU in [ITU I371,1995] is "the set of actions taken by the network at the call set-up phase (or during call negotiation phase) to establish whether a Virtual Channel (VC) connection or a Virtual Path (VP) connection can be admitted or rejected". In other words, CAC in ATM networks, decides to admit a connection request only when sufficient resources are available to establish the connection end-to-end, at its required QoS, and maintaining the QoS agreed for the existing connections. Although the primary responsibility of CAC is to maintain the QoS, to operate effectively the CAC should also maintain a high network utilisation. In addition, a switched network requires the CAC to provide a rapid response to a connection request.

[ITU I371,1995] considers the routeing of a connection to be part of CAC. In this studies, however, CAC refers only to the question whether the new connection can be admitted or not on a node-to-node basis. Summarising, the purpose and requirements that CAC has to fulfil are as follows:

1. the network has to be protected from overload;
2. resources have to allocated such that the QoS requirements are met for all established connections;
3. maximal statistical multiplexing gains should be obtained;
4. the required real-time processing should be reasonable.

2.3.1.2 What is UPC?

The purpose of the Usage Parameter Control (UPC) is to detect violations of the traffic contract and take suitable actions at the user access. The traffic contract is specified at connection set-up and binds the user to transmit as declared in the traffic descriptor and the network to provide the user required QoS.

UPC mechanisms should protect the network resources from malicious as well as unintentional misbehaviours that can affect the QoS of other already established connections. In case of violations of negotiated parameters, possible actions may be to discard cells, mark cells as low priority, charge extra, disconnect or re-negotiate the connection. Rescheduling of compliant cells may be used in the UPC as an option. A large number of UPC-algorithms have been proposed: for more details see [Niest, 1990], [Hemm, 1991a], [Rath, 1991], [Race 1022, D122], [Race1022, 1990], [Yam, 1992] among others.

2.3.2 Relation between CAC and UPC

For a consistent traffic control framework in ATM networks it is necessary that CAC and UPC co-operate in a proper way. A careful selection of the traffic parameters used by both control schemes has to be made, because the CAC function can only rely on those parameters that are enforced by the UPC.

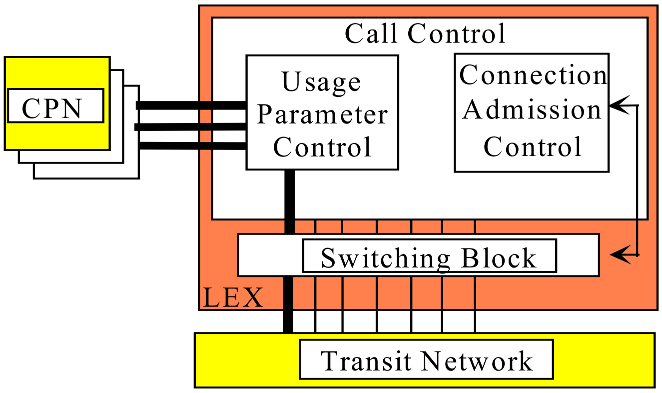

Figure 2.6 Relation between CAC and UPC.

The above figure localises the CAC and UPC as traffic control functions within the more general Call Control function. At connection set-up, a negotiation takes place between the Call Control module and the user that wants to have access to the network. This results in a contract describing the source traffic parameters at a certain reference point. The contract will contain user declared values for the traffic descriptor parameters referred in section 2.2.1 and also the required cell loss, delay and delay variation values. Call Control consults CAC (based on the contract) in order to allow or not the user to start the transmission. If the connection can be admitted, Call Control will then inform UPC with the parameter values required to be enforced and the user is allowed to start a new connection.

2.3.2.1 Control of the peak cell rate

Peak cell rate is of prime importance for the performance of a statistical multiplexer and therefore should be used for CAC schemes. The UPC enforcement of this parameter has been suggested by many authors in the literature (see [Boy, 1990], [Rath, 1991] among many others).

At the reference point (e.g. network access) where the source is controlled by the UPC function, the cells may have experienced a variable delay (e.g., due to waiting time in the buffers, cell multiplexing, etc.). This will alter the traffic characteristics of the source (namely its cell inter-arrival time) and it is referred as Cell Delay Variation (CDV).

The peak cell rate is defined as the inverse of the minimum inter-arrival time; but due to CDV, when the cells arrive at the UPC function, they may have experienced a variable delay. Therefore, some deviations from the negotiated inter-arrival time should be admitted by the UPC function.

The acceptable deviations from the negotiated minimum inter-arrival time is described by the maximum CDV. Therefore, a parameter describing the maximum allowed CDV should be included in the traffic contract. The user should then specify:

- peak cell rate (traffic descriptor parameter);
- maximum allowed CDV (QoS parameter).

A precise definition of what it means to respect/violate the contract is needed to evaluate the performance of UPC algorithms regarding peak rate policing ([Hemm, 1991b], [Hemm, 1992]). An approach that considers that if one of the above mentioned parameters is violated, the contract is regarded as violated, is not practical, as the two parameters are not independent. Therefore, the combination of the two parameters should be controlled. Bearing this in mind, an algorithm referred as "generic cell rate" algorithm is proposed in appendix 1 of ITU recommendation I371 ([ITU I371 A1,1995]) and explained in detail in ([Stall, 1995], pp.507).

2.3.2.2 Control of the mean cell rate

In order to take advantage of the statistical multiplexing gain, traffic parameters other than the peak cell rate have to be taken into account in CAC schemes. The *mean cell rate* is one of those parameters. Nevertheless, several studies (see [Rath, 1991], [Rath, 1989], [Hemm, 1989], [Kron, 1991]) have shown that a strict control of a general mean value is not possible. This is because in order to control the mean cell rate exactly, the necessary counter limits for the Leaky Bucket algorithm (e.g. maximum bucket level) may become very high for bursty (high peak to mean ratio) broadband sources. The source will then be allowed to transmit on a high bit rate for a long period of time, which is not acceptable, because it may not provide adequate protection for rather small buffers within the network.

The problem may be reduced by choosing parameters that allow a higher mean cell rate than the agreed contract (safety factor). This is referred in ITU's recommendation I.371 ([ITU I371,1995]) as Burst Tolerance (BT) and the upper bound on the average cell rate of a connection as sustainable cell rate (SCR). Thus, using the "generic cell rate" algorithm referred before and explained in detail in ([Stall, 1995], pp.511) the sustainable cell rate can be enforced over the duration of the connection.

2.3.2.3 Control of mean burst length

COST 224 Final Report ([COST, 1991]) gives some suggestions on burst length policing; it states that a UPC approach based on Leaky Bucket is "not very sensitive to the burst duration as long as the source increases the burst and silence duration in the same proportion, keeping the mean rate constant". It also refers that a window method (such as Moving Window, Jumping Window or Exponentially Weighted Moving Average) because is based on counting the number of cells sent in N consecutive bursts is appropriate to control the mean burst length as long as the parameter N (window size in number of bursts) is properly configured. The conclusion is that "in order to have small error probabilities, a large number of bursts should be controlled", i.e., the value suggested for the parameter N for an error probability of the order of 10^{-3} is shown to be 80 (see [COST, 1991], pp. 63 for more details).

2.3.2.4 Rasmussen's approach

In Rasmussen et al ([Rasm, 1991]) a different philosophy on the interpretation of the traffic parameters that compose the traffic descriptor is taken. A set of two parameters

- a peak cell rate (R_p) measured over a short period T_s of time;
- a mean cell rate (R_m) measured over a longer period T_l of time;

referred to as "virtual cell rate" parameters are proposed:

> "A virtual cell rate can be interpreted as the maximum of the average cell rate over all intervals of a chosen length. For short intervals, the limitations on the traffic are strict. The longer the intervals are, the more the average will smooth out variations in the cell rate. This means that especially for bursty sources, the virtual mean cell rate can be set much lower than the virtual peak cell rate. Note that the real mean cell rate will be the limit of virtual mean cell rate as T_l tends to infinity." ([Rasm, 1991]).

The T_s parameter is chosen such that the source is strictly peak rate policed; it is assumed that the minimum duration of a connection is T_l. These assumptions yield:

$p \leq R_p$, where p stands for the real peak cell rate;

$m \leq R_m$, where m stands for the mean cell rate.

These two inequalities ensure that the traffic load is not underestimated if π and μ are to be replaced by R_p and R_m, respectively. The introduction of these virtual cell rate parameters envisages their direct translation into a policing function mechanism based on two running windows, one for the virtual peak cell rate and one for the mean cell rate ([Rasm, 1991]).

Rasmussen's approach meets the requirements proposed in this study because from the parameter set (R_p, R_m, T_l) we can easily obtained an estimation of the mean burst length value. Quoting ([Rasm, 1991]):

> "Assume that all peak rate bursts have maximal allowed duration and that the virtual mean is met. Then the traffic process is completely determined as a periodic on-off process with peak rate R_p, mean rate R_m, burst duration $\beta' = T_l R_m / R_p$, and silence duration $\alpha' = \left(1 - R_m / R_p\right) T_l$.
> "

Hence, the mean burst length, l, can be obtained using equation $l = R_p \cdot \beta'$.

From what has been written before, a final comment can be made: if the mean burst length value is to be obtained from the formula stated above,

1) the user will have to specify only the connection's mean and peak bit rates,
2) a UPC mechanism to enforce the value for the mean burst length is no longer needed.

On the other hand, the reasoning presented on the last paragraph comes from the assumption that all peak rate bursts have maximal allowed duration. If the traffic generated by the source does not follow this "worse case" behaviour, than the estimated value for the mean burst length is not accurate and the QoS prediction could be too pessimist. This study suggests to follow Rasmussen's assumption if a user declared value for the mean burst length cannot be obtained.

2.4 CAC methods

CAC methods vary in terms of the traffic scale (cell, burst or rate-variation scale) that is addressed for the traffic model ([Kilk, 1994]). Given this, three main limitations that have an effect on the CAC response can be pointed out:

i. limitation due to the *traffic models* being used: some CAC methods are tailored for cell scale traffic models (linear CAC) while others are tailored for rate scale traffic models (convolution and normal approximation methods);
ii. limitation to a specific technique for the determination of *traffic parameters* (such as measured flow techniques, neural networks)
iii. limitation to a certain method to approximate *heterogeneous traffic* cases (effective bandwidth, effective variance models).

A number of possible algorithms have been studied for CAC purposes (see among others [Ivers, 1990], [Hui, 1988], [Wors, 1992]). In this section the CAC algorithms mostly referred in the literature are summarised and subsequently described in detail; the CAC algorithms presented are based on resource allocation schemes applied to each link and switching unit.

The convolution CAC algorithm derives a full description of the multiplexed traffic by convoluting the traffic distributions of all connections ([Ivers, 1990]). The congestion probability and cell loss probability can then be evaluated by comparing the traffic distribution with the available capacity. The accuracy of this method is only limited by the choice of parameters used to describe the component traffic streams. The main drawback is that the convolution calculus is very lengthy and therefore time-consuming.

Another CAC approach, the two-moment allocation scheme (also referred as normal approximation), assumes the independence of the connections traffic behaviour and characterises the multiplexed traffic by a normal distribution with parameters given by the sum of the means and the sum of the variances of each of the multiplexed connections. A connection is only admitted if the congestion probability derived from the tail of the normal distribution is less than a pre-specified threshold ([Hui, 1988]).

The notion of "effective" bandwidth for each connection aims to summarise in a single figure the bandwidth and QoS requirements of a connection ([Guer, 1991], [Mur, 1991a]). The CAC method based on this parameter is the Linear CAC which reduces the CAC task to the simple problem of determining whether the sum of the effective bandwidth of each of the connections is greater then the resource capacity; if that is the case the connection is rejected, else it is admitted.

The other group of CAC approaches is based on heuristics and data modelling techniques. The neural network approach is an example of this kind of approaches ([Wors, 1992], [Hiram, 1990]). It's main advantage is the clustering of data obtained from ATM traffic measurements in a net structure that constitutes the traffic model.

2.4.1 Basic definitions

In the following, the term *source* is used for a connection that has certain traffic parameters (e.g. peak and mean rates). These parameters are valid for the input link at the ATM node under study. All sources with the same values of parameters belong to a *source type*.

The other notations used in this thesis are:

C = link capacity;
K = output buffer size;
p = peak rate;
m = mean rate;
l = mean burst size;
α'= mean off (silence) duration;
β' = mean on (active) duration;

ρ = maximum load factor;

N = number of sources;

The CAC functions described in the following are based on the connections statistical traffic parameters values (such as peak, mean and variance of the bit rate) and on the network dimensioning parameters (link capacity and output buffer size).

The bandwidth requirement of a connection of type i is described by means of a discrete random variable X_i, $i \in \{1,...,S\}$, where S denotes the number of different connection types. The distribution $P(X_i = b_i)$ defines the (steady-state) probability that a connection of type i requires the bandwidth b_i.

Let the vector $(N_1, ...,N_S)$ denote the system state at the connection level, i.e., N_i connections of type i, are established on a link with available capacity C. It is assumed that the sources are independent of each other. The random variable Y for the bandwidth required by the total traffic stream is given by the sum of the random variables of the individual traffic streams:

$$Y = \sum_{i=1}^{S} N_i \cdot X_i \,.$$

The performance parameters that can be calculated by means of the distribution of Y are ([R1022, D120, 1990]):

- the *congestion probability*, *PC*, defined by $PC(Y) = P(Y>C)$;
- the rate (for the total traffic stream) of cells that exceeds the link capacity, *PLT*, given by

$$PLT(Y) = E(Y)^{-1} \cdot \sum_{L>C} (L - C) \cdot P(Y = L)$$

  where $E(Y)$ denotes the mean value of the bandwidth required by the multiplexed traffic.
- the cell rates, *PLI*, for the individual traffic streams X_i, (assuming that the cells exceeding the capacity are randomly chosen) given by

$$PLI(X_i, Y) = E(X_i)^{-1} \cdot \sum_{L>C} \sum_{i} \frac{b_i}{L} \cdot (L - C) \cdot P(X_i = b_i, Y = L).$$

PLT and *PLI* provide estimators for the total cell loss rate and the individual cell loss rate, respectively.

2.4.2 Convolution approach

The convolution CAC assumes that the bandwidth distribution of each connection is known and calculates the bandwidth distribution of the superposition of several connections by convolving the distributions of each of the connections (see, e.g. [Ivers, 1987], [Ivers, 1990]). The convolution algorithm is based on a server bufferless model, that is, cells generated by several sources are served at maximum speed (limited by the link bandwidth capacity) and if momentarily the multiplexed traffic rate exceeds the link capacity cells are lost.

Let Y denote the random variable for the bandwidth required by the already established connections on a link. A connection of traffic type i will be admitted if the following conditions are fulfilled ([R1022, D120, 1990]):

$$E(Y + X_i) < \rho \cdot C,$$

$$PC(Y + X_i) < \varepsilon_1,$$

$$PLT(Y + X_i) < \varepsilon_2$$

$$PLI(X_j, Y + X_i) < \varepsilon_3,$$

where $j \in \{1,...,S\}$ and possible parameter values are $\rho=0.8$, $\varepsilon_1=\varepsilon_2=\varepsilon_3=10^{-10}$.

Depending on the number of different traffic types to be considered and the number of states of each type, the computations may be very time consuming. This is an obstacle if a fast real-time processing is required. Considering this and the fact that the users might not be able to give at call set-up a lot of information on statistical parameters, a convolution algorithm using peak and sustainable cell rates has been proposed by Marzo et al. ([Marzo, 1993]). The new algorithm is referred as "enhanced" convolution approach (ECA) by the authors. ECA allows real-time calculations by using a set of mechanisms designed to improve the running time performance (see [Marzo, 1993] for details). ECA uses the multi-nomial distribution function to model the aggregated traffic and groups the connections in classes, where each class characterises a different traffic type (see chapter 5 for more comments on this convolution algorithm).

2.4.3 Normal approximation

This method is based on the assumption that the distribution of the required bandwidth of all existing calls can be approximated by a Normal distribution described by the first and second moments (mean and variance) of the rate distribution of the multiplexed traffic. However, from the mathematical point of view it is known that this approximation provides accurate results only in the neighbourhood of the mean value, when many sources of a traffic type are superposed. Comparisons with the convolution approach show (see [Ivers, 1990]) that for various traffic mixes the Normal distribution may accept too many connections. An improved approximation based on the large deviation approximation is described in [Hui, 1988].

The reason for investigating this approach is that the probability of exceeding a certain capacity, can be easily expressed by the ε-fractile, f_ε, of the standard normal distribution described by mean, $E(Y)$, and variance, $Var(Y)$, given by the sum of the mean and variance of each of the multiplexed connections, respectively. In these conditions, the congestion probability is given by:

$$PC(Y) = P(Y > \rho \cdot C) < \varepsilon \quad \Leftrightarrow \quad E(Y) + f_\varepsilon \cdot Var(Y)^{0.5} \le \rho \cdot C$$

The inequality on the right is called the "two-moment allocation scheme". If the connection admittance procedure is based only on the congestion probability, then with the assumption of a Normal distribution the CAC algorithm is very simple. Given the mean and variance of each traffic type, only the ε-fractile of the normal distribution of the aggregated traffic must be known and no other distribution function needs to be identified (see also [Ivers, 1987] for more details on the CAC algorithm based on this method).

2.4.4 “Effective” rate CAC procedures

The main idea behind the notion of *effective rate* (also referred as *effective bandwidth* or *equivalent bandwidth*) is the representation of the bandwidth requirements of a source in a single figure. This single-valued descriptor is a value between the mean and the peak bit rates of a connection (defined at call set-up) and represents the multiplexing attitude of a connection within a certain environment (i.e. given multiplexer dimensions and background traffic). It expresses the bit rate that is effectively consumed by the connection in that environment when the constraint that the QoS for all connections multiplexed is respected. Once this goal is reached, simple linear CAC schemes arise.

2.4.4.1 Linear CAC function

The linear approach is defined by a linear function F

$$F(N_1, N_2, \ldots, N_S) = \sum_{i=1}^{S} B_i \cdot N_i$$

where B_i is the *effective rate* of traffic type *i*. A connection of traffic type *i* will be admitted if $F(N_1, N_2, \ldots, N_{i+1}, \ldots, N_S) \leq \rho \cdot C$, where ρ is the maximum load factor. The load factor is a pre-defined value given by results for the G/D/1/K queue for specific input streams (see [Louv, 1988]). For example with a cell loss probability equal to 10^{-9} and buffer size equal to 64, ρ is 0.85.

Numerical results ([Hui, 1988]) show that for many traffic mixes the boundary of the admissible load region can be approximated by a linear function, but depending on the heterogeneity of the traffic types multiplexed, the results may be close to the peak allocation scheme.

2.4.4.2 Determination of the effective rate

Despite the simple outline of the effective rate mechanism, the full development of the idea presents some difficult problems. Firstly, a method of associating a particular source with a certain effective rate value must be devised. Secondly, these effective rates must enjoy an additive property, if they are to be employed for a linear scheme. The latter property means that whenever a combination of connected sources features a sum of effective rates below the link capacity, then it should be guaranteed that the QoS will be respected for all admitted connections.

Considering the simplicity requirement of the linear schemes, several definitions for the effective rate have been introduced in the literature (see [Mase, 1991] for a review on these). There are basically two variants:

- Guerin et. al ([Guer, 1991]) obtains a conservative definition for the minimum output buffer size and link capacity that the multiplexer should provide, so that feeding it with a single on/off source, the required QoS could be respected. The determination of this rate is computed by assuming each source in isolation via the usage of the asymptotic expression for the solution of a fluid flow model. Hence, using the sum of the effective rates as the criterion for CAC is a conservative strategy that ignores multiplexing gain. Guerin et. al ([Guer, 1991]) also propose a joint strategy using the stationary rate distribution to partially compensate for this.

- Murase et. al ([Mur, 1991a]) propose an alternative definition using a bufferless fluid-flow model to determine the effective rate of a stream, which is defined as the ratio of the output link capacity to the maximum number of sources that can be superposed without violating the QoS. According to [R1022 D126, 1992] this definition of the effective rates allows a linear CAC to achieve higher multiplexing gain than a similar CAC based on Guerin's method but, unfortunately, is not always conservative, that is, the QoS may not be respected even if the sum of the effective rates of the superposed streams remains lower than the link capacity. In [Mur, 1991b] the notion of “interfering” class is introduced to describe this sub-linear effect which is attributed primarily to large differences in burstiness or peak rates among different traffic classes.

An approach closer to the one given in [Mur, 1991a] is presented in [Race1022 D126, 1992]; but this time a buffered fluid model is used, that is, a multiplexer with a specific output buffer size and link capacity is assumed. The method for the determination of the effective rate and the relevant properties for CAC schemes outlined. The basic source model assumed is the On-Off type.

A generalisation of the effective rate definition to sources represented as a superposition of identical On-Off sources or sources that feature a CBR component has also been presented in [Race1022 D126, 1992]. The QoS requirements refer only to cell loss and are expressed as a probability denoting the maximum allowable overflow probability.

Experiments performed by the RACE Consortium R1022 ([Race1022 D126, 1992]) for heterogeneous traffic mixes revealed that the burstiness (ratio between the peak rate and the mean rate) and the mean volume of the bursts are the two parameters that primarily affect the linear behaviour. Large differences in burstiness and mean volume of the bursts presented by the multiplexed traffic classes causes deviation from linearity. In some cases the influences of the two are counterbalancing each other. It is still an open problem, however, to determine regions in the traffic parameter space preserving a desired degree of linearity.

Therefore, when the traffic streams multiplexed on a link are statistically identical, a simple CAC procedure can be applied based directly on the effective rate definition given in [Race1022, 1992]. However, since ATM is introduced to support a variety of services with quite divergent characteristics, the homogeneous case will only be an exception. The challenge is to extend the simple effective rate-based linear CAC scheme to work accurately in a heterogeneous traffic environment, without loosing the possibility of achieving a statistical multiplexing gain.

In order to capture the non-linearity shown for the boundary of the admissible load region, in heterogeneous traffic mixes, whilst using a linear CAC approach, a modified effective rate definition has been proposed in [Race1022 D126, 1992]. The modified effective rate approach reduces the overestimation of the admissible-load region. However, for very strong heterogeneity, even using the modified rates there may be some overestimation. In such cases, a rate partitioning scheme between the different classes may be applied (see [R1022 D126, 1992] for more details).

2.4.5 Neural networks approach

The usual approach to CAC is to find a solution that relates a queueing model with a source model in order to relate offered traffic (in terms of offered connections and respective source model parameters) to a performance threshold under which the QoS requirements are guaranteed. The traffic descriptor chosen for offered connections is mapped in the source model parameters. This approach is problematic if the source (traffic) model does not fit the real sources well, that is, the source model does not reflect differences between a low load source and a high load source because these characteristics cannot be visualised from the traffic descriptor parameters.

In attempting to solve the uncertainties faced in the choice of a source model, CAC algorithms based on automatic learning principles have been proposed in the literature: the main idea is to learn the relation between the offered traffic and the expected service quality during on-line operation. The allocation of resources to virtual connections would be based on past experience and would be adaptive to changes in traffic characteristics and possibly also system performance.

Considering the previously stated, Worster ([Wors, 1992]) and Hiramatsu ([Hiram, 1990], [Taka, 1990]) presented CAC approaches based on neural networks (NN).

The network state seen by the neural network is $X = \{n_1, n_2, ..., n_M\}$, where n_i denotes the number of active connections of type i in the system. The CAC problem is then formulated as a pattern recognition problem: upon recognition of the load pattern X, a yes/no decision is made to admit/reject the connection request.

NN have also been used to convert the source's traffic descriptor into parameters that can be used by a suitable CAC procedure (see [Taka, 1990], section 5.2). Kilkki ([Kilk, 1994]) comments on this approach referring that "the same method that is required to train the neural network to recognise acceptable [traffic] patterns can be used for the determination of effective bandwidth or effective variance of a conventional CAC. If pre-defined source types are used, there is probably no reason to use a neural network for on-line calculation" between traffic descriptor and CAC parameters". However, it must not be forgotten that NN response is not just a reflection of the training data but can also be a generalisation of the same data for similar traffic patterns. The conventional method of determination of effective rates does not possess such characteristic.

2.4.6 Other approaches

Simulated and analytical results have proved that performance parameters like cell loss are strongly dependent on variances in the traffic statistic characteristics. Hence, to be able to control cell loss, the CAC function needs a traffic model which represents accurately the sources' traffic behaviour. Because a traffic model which can be used for any type of ATM traffic has not yet been identified, at call set-up, the user is expected to provide the network with a set of parameter values that are able to characterise the traffic to be generated.

Very often a user will not be able to determine the mean or the variance of the traffic bit rate, only the peak bit rate can be given. A possible solution to overcome this, is to monitor the behaviour of the traffic sources and perform a CAC based on the measurements made.

Several approaches have been introduced that use this solution:

- the congestion control proposed in [Boy, 1990] is based on the classification of the different sources in four types (CBR, VBR with continuous bit rate range, VBR with discrete bit rate range and others). The congestion control is performed at different levels: call, burst and cell levels. At call set-up a load control is performed by an expert system based on load measurements and prediction.
- Saito's method ([Saito, 1992]) is based on an estimated distribution on the number of cells arriving during a renewal period, $\hat{p}(t)=(\hat{p}(0;t),\hat{p}(1,t),...)$ and on the measured distribution of cells during N periods, $q(t)=(q(0,t),q(1,t),...)$. The estimated distribution for the period $(t+1)$ is: $\hat{p}(t+1)= \alpha \mathrm{q}(t) + (1-\alpha)\hat{p}(t)$. Thus, the probability that eactly n cells arrive from the connected sources can be estimated by convoluting the distribution of each source. The source traffic behaviour is determined by two parameters: mean and maximum cell rate values during the observation interval. The interval proposed is equal to the time at which $K/2$ cells are transmitted, where K is the buffer size in cells.

Parameters describing the traffic process, such as the mean and the intensity of variations, can be estimated with the aid of on-line measurements instead of the theoretical values which have been calculated from declared traffic parameters. The drawbacks associated with this approach have to do with the difficulties in the choice of a suitable time scale for the measurements: "If there are long range fluctuations, for example because of scene changes in video sources, the measuring period should be very long in order to capture all fluctuations. On the other hand, during a long measuring period connections will be established and released, and we cannot suppose that the behaviour of sources with rapid fluctuations remains unchanged." ([Kilk, 1994]).

2.4.6 Implementation aspects

From the computation point of view only the two-moment allocation scheme (given the ε-fractile) and the linear approach (given the effective bandwidth for each traffic type) fulfil the requirements given in section 2.3.1.1 for a CAC procedure. But, as discussed in the previous sections, these mechanisms could be too optimistic (network overload) or too pessimistic (low multiplexing gain).

To overcome this problem a CAC implementation in two levels has been proposed ([Kron, 1991]). The idea behind this "two-level" model is to combine the advantages of a fast real-time processing first level algorithm with more accurate (but slower) algorithms at the second level. Hence, at the first level, a CAC algorithm with a fast real time processing (e.g. linear approach) will give a quick answer whether a connection can be admitted or not; whereas a slower but more accurate CAC algorithm (e.g. convolution) should be implemented at the second level to calculate the bandwidth values actually being used by each of the connections. The admittance or rejection of the connection is based purely on the decision made on the first level CAC. The second level module would be activated periodically to calculate and update the required bandwidth values.

2.6 Conclusions

CAC in ATM Networks addresses two fundamental problems related with the unpredictability of the traffic characteristics and the multiplexing effects on the QoS of the multiplexed traffic:

- *source modelling*: ATM sources' traffic behaviour needs to be accurately described either by a set of parameters or a traffic model that relates these parameters in a statistics framework;
- *resource allocation*: knowing the characteristics of the traffic streams, a relation between available network resources (link capacity and buffer storage space) and multiplexed traffic demands needs to be identified and categorised in terms of the associated QoS.

As stated in previous sections, the majority of CAC approaches proposed in the literature assumes a specific source model (e.g. On-Off model, normal distribution) and solves the resource allocation problem considering the multiplexed effects inherent to the sources' traffic model (for example, convolution and normal approximation). When applied to heterogeneous scenarios or to scenarios where the sources present large differences in burstiness and/or peak rates, the before mentioned CAC approaches can be either too pessimistic or too optimistic ([Ivers, 1990], [R1022 D126, 1992]).

Other approaches, such as the linear CAC approach, are based on a specific parameter: effective bandwidth, to relate bandwidth demands to QoS requirements. Linear approaches for which the effective bandwidth values are calculated (for each traffic type) using approximations based on homogeneous traffic mixes tend to be too optimistic (i.e. too many connections are admitted for a specific QoS) when applied to heterogeneous traffic mixes ([R1022 D126, 1992]).

The "two-level" approach satisfies the real time processing requirements as long as used with a linear or two-moment approach as the first level algorithm. If the convolution approach is the algorithm chosen for the second level algorithm, a conservative approach to CAC is obtained. This is because the convolution approach does not consider the burst phase duration nor the size of the output buffer; in other words, the convolution approach ignores the multiplexing gain obtained by queueing the whole burst or part of it. Hence, transmission capacity will be wasted using the convolution approach if the burst size is long.

Kilkki ([Kilk, 1994]) suggests that a neural network (NN) approach can be used "to refine on-line traffic measurements regarding fluctuations of incoming traffic process, queue length, etc." These fluctuations occur due to delays due to buffering before service, which may lead to a higher peak rate than the user declared peak rate value. The mean bit rate measured might also differ from the user declared mean bit rate value, because the user might not be able to define this value accurately. A CAC approach which uses the NN estimated statistical parameters instead of the user declared ones would be able to allocate resources more efficiently. Independently of the purpose NN are used for, considerations on the fact that in order to provide an accurate answer NN require successive training periods and that can be a drawback if CAC needs to adapt to changes in a reasonably short period.

The fuzzy logic approach to CAC further described shares the same aims as the NN learning approach: to support the CAC decision making by modelling the relationship between the traffic statistical behaviour and the monitored cell loss ratio via a model which is embedded in the architecture used to represent the knowledge (as opposed to a stochastic based estimation model), more precisely:

- a set of neurones and respective connections in the case of NN,
- a set of "if-then" rules relating the input and output fuzzy variables in the case of fuzzy logic systems.

The main advantage of neural and fuzzy logic based CAC approaches over conventional mathematical approaches (such as the convolution, two-moment based approach and linear) is that they estimate sample functions from samples collected on-line from the system under study. Also, neural and fuzzy systems are numerical knowledge based approaches as opposed to symbolic knowledge based approaches (e.g. expert systems), which allow for a greater accuracy in systems described by numerical values.

The main differences between fuzzy and neural systems is in how they estimate sampled functions, more precisely: how the samples are represented and stored and how the inference mechanisms work (for more details see [Kosko, 1992]). The choice between fuzzy or neural systems has to do with the nature of the problem and the availability of numerical and structured data. NN so far seem best applied to ill-defined two-class pattern-recognition problems (defective or non-defective, accept/reject, etc.). Fuzzy-based systems have been applied to problems where a comparison with theoretic models is possible and problems where there is heuristic knowledge (obtained by consulting experts in the area or monitoring the system) which can be formulated in linguistic rules of the type: “If traffic-load is HEAVY then cell-loss-ratio is HIGH”.

Other differences between fuzzy and neural systems appear at design stage. The neural approach requires the specification of a non-linear dynamical system: the acquisition of a representative set of numerical training samples, and the encoding of those training samples in the dynamical system is done through repeated learning cycles. The fuzzy system requires only that we partially fill in a linguistic “rule matrix”. According to Kosko ([Kosko, 1992]):

> “The neural network suffers a deeper problem than just the computational burden of training. What does it encode? How do we know the network encodes the original structure? What does it recall? (...) Unlike an expert system, we do not know which inferential paths the network uses to reach a given output or even which inferential paths exist. (...) We are left with an unstructured computational black box. We do not know what the neural network encoded during training or what it will encode or forget in further training.”

Considering all the expressed above, a fuzzy logic decision support tool to CAC is proposed in the following chapters (also referred in the following as FCAC). The aim is not to propose the FCAC as the best approach for the CAC problem but certainly to show that such kind of approach can be of great usefulness where other approaches fail, namely:

- to capture the non-linearities in the bandwidth actually being used by an heterogeneous traffic scenario that are not so easy to capture using a Linear CAC approach based on effective bandwidth values;
- to obtain a faster, although accurate response for traffic scenarios previously studied overcoming the computational efforts required by the convolution approach;
- to construct a system easier to understand and update than a neural network approach, but still keeping the philosophy of a tool that learns from data samples obtained from on-line measurements and is adaptable to changes in the traffic patterns introduced by the addition of new services in the network.

The studies further presented do not address the source characterisation problem, delegating the task to a traffic analyser (e.g. based on a NN as referred in Kilkki) which refines the traffic parameters declared at connection set-up. Instead, and for the time being, the UPC enforced parameters referred in section 2.3.2 are adopted to characterise the connection’s traffic behaviour.

It is the aim of the next chapters to address the resource allocation problem, by identifying a relation between available network resources (link capacity and buffer storage space) and multiplexed traffic demands using a fuzzy-based tool, in the following referred as FCAC. FCAC predicts the worse (in the sense of maximum) cell loss ratio that can be expected for the multiplexed connections. The predicted cell loss value will be a CAC decision support parameter, in the sense that, if the cell loss requirement of any of the existing connections or the new connection is not possible to be satisfied, the new connection will be rejected.

3 Fuzzy logic systems

Fuzzy logic systems represent the knowledge on the relation between inputs and outputs, for a specific problem area, without a mathematical model of how outputs functionally depend on inputs; fuzzy logic systems are model free estimators ([Kosko, 1992]).

Fuzzy systems can be usefully applied in areas where:

- expert knowledge is required (e.g. control systems);
- the characteristics of the monitored data change over time (e.g. dynamic systems modelling);
- an algorithm based solution, although possible, is very complex and, therefore, difficult to implement.

Fuzzy logic systems acquire knowledge using sampled numerical data obtained from monitoring the input and output variables of the process under study. Fuzzy logic systems can also acquire knowledge by consulting experts to provide linguistic associations such as (*Heavy*, *Longer*); or equivalently, "If X is *Heavy* then Y is *Longer*", where X and Y refer to input and output linguistic variables (such as *Weight* and *Width*), respectively. These associations are also referred in the literature as *fuzzy rules*.

The theoretical support behind fuzzy systems ability to represent the mathematics of the system (fuzzy logic) is, quoting Zadeh, a means of "computing with words" and is described very clearly in [Zad, 1993]:

> "There are two concepts within fuzzy logic which play a central role in its applications. The first is that of a *linguistic* variable, that is a variable whose values are words or sentences in a natural or synthetic language. The other is that of a *fuzzy if-then* rule in which the antecedent and consequent are propositions containing linguistic variables. The essential function served by linguistic variables is that of granulation of variables and their dependencies. […] In this respect,

> fuzzy logic mimics the crucial ability of human mind to summarise data and focus on decision-relevant information."

Another very important concept in fuzzy logic, is that of a *fuzzy set*, that is, a set to which members belong with a degree of membership. This extends the notion of a crisp set in set theory, given that, an element of the universe where the fuzzy set is defined, is said to belong to the fuzzy set with a membership degree, say 0.75, whereas for traditional crisp sets, the element either belongs or not to the crisp set.

The inference process in fuzzy logic systems differs from the one used in other knowledge based systems, in that fuzzy logic systems fire all the rules in the rule base (parallel inference) and the weight associated with each rule in the inferred output is determined by the degree of matching between the input and the rules premises. Regarding this, Kosco ([Kosco, 1992]) states:

> "Fuzzy based systems include rules to direct the decision process and membership functions to convert linguistic (fuzzy) variables into the precise numeric values required by the application. [...] When asked a question or given an input, a fuzzy system fires each [fuzzy] rule in parallel, but to different degree, to infer a conclusion or output."

Although the first applications of FBS have been in the control area ([Mamd, 1973]), FBS have also been applied in computer vision ([Kell, 1992]), preference modelling and decision analysis ([Roub, 1985]), operations research and optimisation ([Verd, 1984]) and pattern recognition ([Pedr, 1984]).

Recently, automatic methods of learning from examples (system inputs and outputs) have been studied and used in conjunction with a fuzzy-based tool,

- Delgado and Gonzalez in [Delg, 1993] have proposed a method whereby measures of uncertainty derived from the evidence of numerical samples are associated with fuzzy rules;
- Takagi and Sugeno in [Tak, 1985] proposed a method of identification of fuzzy systems by modelling directly the control actions of an operator using fuzzy rules where the consequent is a linear input-output relation;

- Pedrycz ([Pedr, 1984], [Pedr, 1993]) proposed the use of training techniques associated with neural networks to derive the fuzzy model for a particular problem. These hybrid methods allow the conjunction of the flexibility of an adaptive system with the fuzzy ability to represent vague knowledge; successive experience refines the rules improving gradually the performance of the system.

The next sections will describe an automatic learning method to identify fuzzy systems, following that, the fuzzy inference model adopted in this studies is presented and, finally, the definition of the input and output fuzzy variables as well as the fuzzy rule format considered in the fuzzy logic tool proposed in this thesis is given. Initially, the basic concepts of fuzzy set theory and fuzzy logic are reviewed.

3.1 Basic notions

3.1.1 Fuzzy sets

The concept of a fuzzy set can be defined by changing the usual definition of the characteristic function of a crisp set, so as to introduce *degrees* of membership. Crisp sets either allow full membership or no membership at all. Given for example the crisp set *T*, of equilateral triangles, and its characteristic function $T(\cdot): T \to \{0,1\}$, there are only two possible choices:

$T(x) = 1,$ if *x* is an equilateral triangle or

$T(x) = 0,$ if *x* is not an equilateral triangle.

Fuzzy sets allow partial membership. A fuzzy set *F* is defined by giving a reference set *U*, called the *universe*, and a mapping $F(\cdot): U \to [0, 1]$, called the *membership* function of fuzzy set *F* ([Zad, 1965]. *F*(*x*), for $x \in U$, is interpreted as the degree of membership of *x* in the fuzzy set *F*. Suppose, for example, that *U* is the set of all individuals and *F* is the fuzzy set of *Tall* individuals, the membership degree of an individual *x* in *F* is the degree (between 0 and 1) to which *x* can be categorised as a tall individual.

This gives a viable model for the vague categories of natural language (*Heavy*, is another example), defined on a universe which could be defined using a numerical scale (*U* = set of weights) or the set of objects qualified by these categories (*U* = set of individuals). *F*(*x*) then expresses how much the value (or the object) *x* is compatible with the concept *F*.

Figure 3.1 shows some of the most commonly used shapes of membership functions that can be found in the literature.

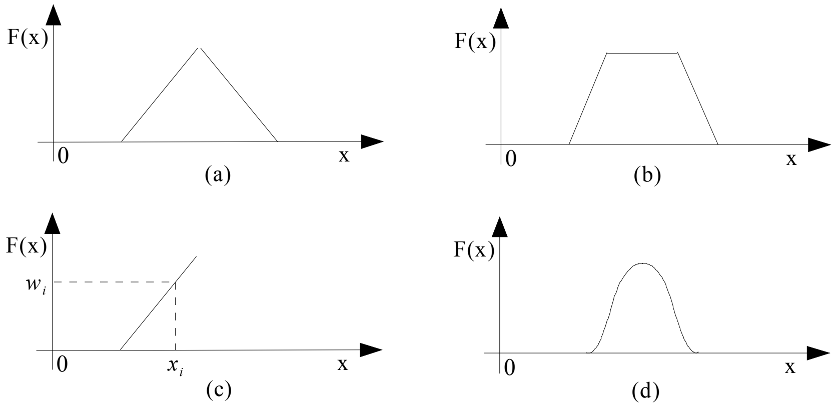

Figure 3.1 (a) Triangular, (b) Trapezoidal, (c) Monotonic, (d) Bell-shaped membership functions

The *support* of a fuzzy set *F*, defined in universe *U*, is a crisp set that contains all the elements $x \in U$ which have degree of membership greater then zero, that is, $\{x \in U: F(x) \geq 0\}$.

The *α-cut* of a fuzzy set F, F_α, is defined as the crisp set of all the elements of the universe *U* which have memberships in *F* greater than or equal to *α*, that is, $F_\alpha = \{x \in U: F(x) \geq \alpha\}$.

The *height* of a fuzzy set is defined as the value of the maximum degree of membership among all the elements of the support of *F*. If *height* (*F*) = 1, that is, $\exists\, x \in U: F(x) = 1$, then *F* is called a *normalised* fuzzy set.

Fuzzy or *linguistic* variables are variables whose values are expressed as fuzzy sets. The *fuzzy domain* or *domain of discourse* of a fuzzy variable *X* is the set of fuzzy sets that the fuzzy variable can take as values. As an example, the set {*Light*, *Medium, Heavy*} is the fuzzy domain of fuzzy variable *Weight*, where *Light*, *Medium* and *Heavy* are fuzzy sets defined in the same universe *U*.

3.1.2 Architecture of a fuzzy logic system

The architecture of fuzzy logic systems (see figure 3.2) is composed of:

- the *knowledge base* or *rule base*, expressed in the form of a set of fuzzy rules;
- the *data base* containing the definitions regarding the discretisation and normalisation of the universes, the definitions of the fuzzy domains of each fuzzy variable and the definition of the membership function for each fuzzy set;
- the *coding* (fuzzifier) algorithm to match numerical input values with the fuzzy sets of the domain of the correspondent fuzzy variable;
- the *fuzzy inference* mechanism, that is, the algorithm that by firing each of the rules of the knowledge base in parallel enables to obtain an inferred fuzzy set given a set of input values;
- the *decoding* (defuzzifier) algorithm to calculate the crisp output from the inferred fuzzy set;
- the *application area*, for example, the process under control in a fuzzy logic system control application.

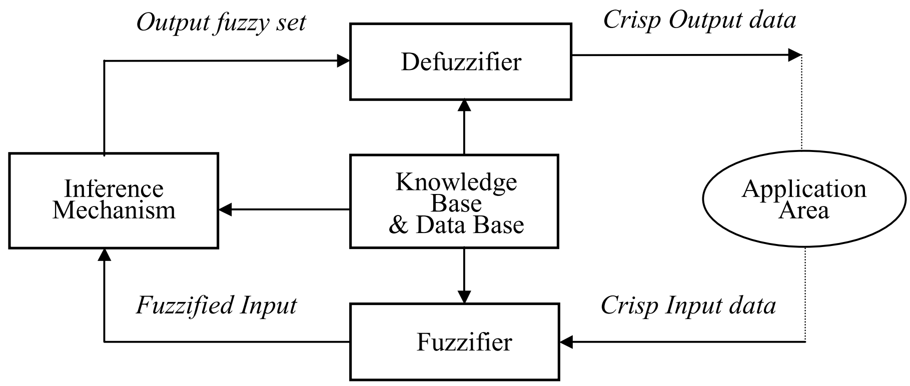

Figure 3.2 A simplified fuzzy system architecture

The figure above illustrates the data flows between the constituents of a FBS during a fuzzy query:

1. a set of input crisp values relative to variables represented in the antecedent of the fuzzy rules is introduced;

2. the values are fuzzified, that is, transformed in values of the universe of definition of the fuzzy variables by normalising and scaling (see for details 3.1.2.2);
3. the fuzzy inference mechanism is activated and an output fuzzy set is obtained by matching the fuzzified values with the antecedent of all fuzzy rules in the knowledge base;
4. finally, the output fuzzy set is defuzzified, that is, transformed in a crisp output value.

3.1.2.1 Fuzzy knowledge base and data base

The *fuzzy knowledge-base* contains is a set of fuzzy rules that express the relation between the input and output fuzzy variables. The rule

R1: If *cornering_angle* is *Moderate* and *loading_weight* is *Heavy*
then *moving_velocity* is *Slow*

from ([Tak, 1991]) where:

- *Moderate* is a fuzzy set of the domain of input fuzzy variable *cornering_angle* defined in universe *U* (e.g. set of allowed angle values);
- *Heavy* is a fuzzy set of the domain of input fuzzy variable *loading_weight* defined in universe *V* (e.g. set of possible weight values);
- *Slow* is a fuzzy set of the domain of output fuzzy variable *moving_velocity* defined in universe *T* (e.g. set of allowed velocity values).

is an example of a rule format that models human concepts expressed in human words. It can be obtained by querying the operator while he drives the mobile vehicle. Apart from this type of fuzzy rules, rules of the following type have also been proposed:

R2: If *cornering_angle* is *Moderate* and *loading_weight* is *Heavy*
then *moving_velocity* = g (*cornering_angle*, *loading_weight*)

where

- fuzzy sets *Moderate* and *Heavy* are now restricted to monotonic membership functions (see figure 3.1 (c))
- *g* is a function that implies the value of *moving_velocity* when *cornering_angle* and *loading_weight* satisfy the rule's premise.

Rule *R2* intends to model directly the control actions of a car driver (for instance, when the driver is parking the car) and have been proposed by Takagi and Sugeno in ([Tak, 1985]). Rules of the format of rule *R2* above, enable to incorporate in the fuzzy rule more knowledge about how to control a specific process. On the other hand, a deeper knowledge of the characteristics of the process is necessary if the relation between the variables in the antecedent of the rules is not polynomial.

The number of input and output fuzzy variables in the antecedent and consequent of the fuzzy rules, respectively, varies according to the application problem under study.

A *fuzzy data-base* contains all the information regarding:

- normalisation and scaling of the actual input values in values of the universe of definition of the fuzzy sets for each fuzzy variable;
- definition of the fuzzy domains of each fuzzy variable, that is, the set of fuzzy sets (linguistic terms) that each variable can assume;
- the definition of the membership function for each fuzzy set.

3.1.2.2 Fuzzification

The process of matching the input value for a specific fuzzy variable with the fuzzy set that the variable assumes as its value is called *fuzzification*. For a crisp input value, *x*, the fuzzification involves:

- normalising the value x to a value, x_n, in the interval [0, 1],
- scaling the normalised value x_n to a value x_s in the universe of the correspondent fuzzy variable, U_X. For a continuous universe, say $U_X = [0, u_{max}]$ this involves multiplying x_n by u_{max}. For a discrete universe, say $U_X = \{0, 0.2, 0.4, 0.6, 0.8, 1.\}$ a scaling function, $\varphi: \Re \rightarrow U_X$ needs to be defined and x_s is calculated as
  $x_s = \varphi(x_n)$,
- to match x_s with the membership function of the fuzzy set, F, that is, by obtaining the membership value of x_s for fuzzy set F (see also figure 3.3).

For more details on fuzzification methods see [Yag, 1991] and [Foul, 1993].

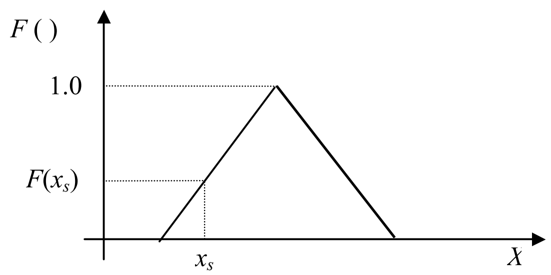

Figure 3.3 Fuzzification of the value x_s, by matching it with the membership function of F.

3.1.2.3 Fuzzy Inference

The *fuzzy inference* mechanism is a process by which the input values for each of the fuzzy variables in the antecedent of the rules (x_0 and y_0 in figure above) are matched with all rules in the fuzzy rule base and an inferred fuzzy set is obtained. The fuzzy inference in a parallel inference in the sense that all rules contribute in a large or small extent to the inferred result. The weight that a specific rule has on the final output is determined by the degree of matching between the input (fuzzified) values and the rule's antecedent.

Figure 3.4 illustrates a simple fuzzy inference mechanism using the max-min inference introduced by Mamdani et al. ([Mamd, 1973]). For more details on fuzzy inference methods see also [Fodor, 1993], [Nov, 1993], [Nguy, 1993], [Redd, 1992], [Hell, 1992], [Park, 1992], [Zimm, 1991], [Dub, 1991], [Dub, 1988], among others.

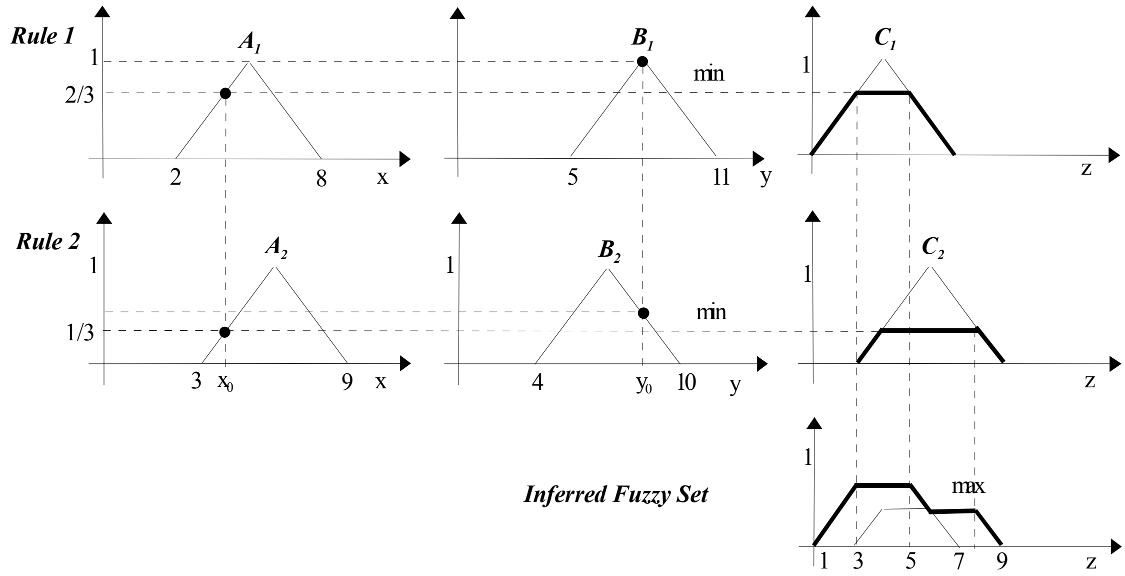

Figure 3.4 Fuzzy Inference using the max-min method (modified from [Yag, 1991], pp.79)
Rule1: IF X is A_1 and Y is B_1 THEN Z is C_1
Rule2: IF X is A_2 and Y is B_2 THEN Z is C_2

3.1.2.4 Defuzzification

The process whereby a non-fuzzy (crisp) output is obtained from the fuzzy set, B^* resulting from the fuzzy inference process is called *defuzzification*. The two most commonly used defuzzification methods are:

- *Centre of area* (COA) method (used for point-wise membership function such as triangular and trapezoidal) calculates the centre of gravity of the distribution of the degrees of membership of B^*. If B^* is defined in a discrete universe U, the crisp output value is obtained using

$$z^* = \frac{\sum_{i=1}^{q} B^*(x_i) \cdot x_i}{\sum_{i=1}^{q} B^*(x_i)}, \tag{3.1}$$

  where q is the number of quantisation levels of universe U, x_i is the crisp value for quantisation level i and B^* (x_i) is its membership value in the inferred fuzzy set, B^*.

When used for monotonic membership functions, CAC is also referred as *Tsukamoto* method. The crisp output value is, now, obtained using the formula

$$z^* = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}, \tag{3.2}$$

where n is the number of rules with firing strength, w_i (in figure 3.4 $w_i = \min(A_i(x_0), B_i(x_0))$) greater than zero and x_i is the crisp value of the support of fuzzy set in the consequent of rule i, with membership value w_i.(see also figure 3.1 (c)).

- *Mean of Maximum* (MOM) method (used for any membership function shape) calculates a crisp output value by averaging the support values of the inferred fuzzy set B^*, with membership value equal to the "height" of fuzzy set B^*. If B^* is defined in a discrete universe U, this is calculated using the formula

$$z^* = \sum_{i=1}^{l} \frac{x_i}{l}, \tag{3.3}$$

where l is the number of elements, x_i, with membership equal to the "height" of fuzzy set B^*.

The most commonly used defuzzification methods: COA and MOM methods, (*Tsukamoto*'s method is a special case of COA method) differ in that:

- MOM selects only the rules which have a strong influence on the inferred result (in this method only the quantised values that reach the maximum membership value for the inferred fuzzy set are used to compute the final result) whereas COA considers the weight of every rule to compute the final result;
- the defuzzified result obtained by MOM is independent of the shape of the membership functions used for the fuzzy sets in the consequent of the rules, as long as the fuzzy sets have a symmetric shape (only one element of the support reaches the maximum membership value). The same is not valid for COA. the shape of the consequent fuzzy sets in COA case is important, because the membership value of all the elements of the support of the inferred fuzzy set is

taken in consideration and not only those for which the membership function achieves its maximum.

3.1.2.5 A fuzzy inference example

In the following, an example from ([Yag, 1991], pp.79-80) is used to demonstrate the max-min fuzzy inference with a fuzzy rule base composed of the rules:

Rule1: IF X is A_1 and Y is B_1 THEN Z is C_1

Rule2: IF X is A_2 and Y is B_2 THEN Z is C_2

and how to calculate the crisp defuzzified value using the COA and MOM methods.

Suppose $x_0 = 4$ and $y_0 = 8$ (see figure 3.4) are the input values for fuzzy variables X and Y, and the following are the membership functions for fuzzy sets A_1, A_2, B_1, B_2, C_1 and C_2:

$$A_1(x) = \begin{cases} \dfrac{x-2}{3}, & 2 \le x \le 5 \\ \dfrac{8-x}{3}, & 5 \le x \le 8 \end{cases} \qquad A_2(x) = \begin{cases} \dfrac{x-3}{3}, & 3 \le x \le 6 \\ \dfrac{9-x}{3}, & 6 \le x \le 9 \end{cases}$$

$$B_1(y) = \begin{cases} \dfrac{y-5}{3}, & 5 \le y \le 8 \\ \dfrac{11-y}{3}, & 8 \le y \le 11 \end{cases} \qquad B_2(y) = \begin{cases} \dfrac{y-4}{3}, & 4 \le y \le 7 \\ \dfrac{10-y}{3}, & 7 \le y \le 10 \end{cases}$$

$$C_1(z) = \begin{cases} \dfrac{z-1}{3}, & 1 \le z \le 4 \\ \dfrac{7-z}{3}, & 4 \le z \le 7 \end{cases} \qquad C_2(z) = \begin{cases} \dfrac{z-3}{3}, & 3 \le z \le 6 \\ \dfrac{9-z}{3}, & 6 \le z \le 9 \end{cases}$$

First, the inputs x_0 and y_0 have to be matched against fuzzy sets A_1, B_1, respectively. This will produce $A_1(x_0)$=2 /3 and $B_1(y_0)$=1. Similarly, for rule 2, we have $A_2(x_0)$=1 /3 and $B_2(y_0)$=2 /3. The strength of rule 1 is calculated by

$$a_1 = Min\ (A_1(x_0), B_1(y_0)\ ) = Min\ (2/3, 1) = 2/3$$

similarly, for rule 2:

$$a_2 = Min\ (A_2(x_0), B_2(y_0)\ ) = Min\ (1/3, 2/3) = 1/3.$$

Applying a_1 to the conclusion of rule 1 results in the highlighted trapezoid fuzzy set of figure 3.4 for C_1. Similarly, applying a_2 to the conclusion of rule 2 results in the highlighted trapezoid fuzzy set C_2. The membership function for the fuzzy set inferred from rules 1 and 2 is obtained by operating fuzzy sets C_1 and C_2 with the *Max* operator (see graphic at bottom of figure 3.4).

Finally and using the COA method, the defuzzified output value is:

$$Z_{COA} = \frac{2\cdot\frac{1}{3}+3\cdot\frac{2}{3}+4\cdot\frac{2}{3}+5\cdot\frac{2}{3}+6\cdot\frac{1}{3}+7\cdot\frac{1}{3}+8\cdot\frac{1}{3}}{\frac{1}{3}+\frac{2}{3}+\frac{2}{3}+\frac{2}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{3}} = 4.7$$

Using MOM, three support values (3, 4 and 5) reach the maximum membership value, 2 /3, in the inferred fuzzy set. Hence, we have

$$Z_{MOM} = \frac{3+4+5}{3} = 4.0$$

As it can be shown with the example, the output obtained using MOM concentrates on the support values with membership values that reach the “height” of the output fuzzy set and therefore, ignores the contribution of support values with lower membership values. The COA output considers all support values and therefore provides an answer with a broad range. Depending on the fuzzy logic system application, the need to focus on broad or concentrated values of the output fuzzy set support will arise and, therefore, the defuzzification method chosen will be either COA or MOM, respectively.

3.2 Learning to identify fuzzy systems from examples

Process dynamics can be described by the relation between its input and output variables. For many complex processes the identification of a functional input-output relation is not straightforward and a possible way to overcome this is by describing the process behaviour through a finite set of fuzzy rules ([Delg, 1993]).

Rules for mobile vehicle breaking control, such as "if Acceleration is *Big* and Distance-to-Target is *Short*, then Breaking-Force is *High*", are just an example of a process described using linguistic (fuzzy) expressions. Even in cases where an exact theoretical model can be identified, a description using fuzzy rules may be useful for ease of understanding the dynamics of the problem at hand and to identify possible input and output variables.

Knowledge acquisition, that is, the transfer of knowledge from some source into a knowledge base, is the first task to face when developing a fuzzy logic system. Hitherto, knowledge acquisition has consist of the application of techniques for eliciting data from experts. Nowadays, new knowledge acquisition techniques are being proposed in the literature using automatic methods, namely:

- Pedrycz [Pedr, 1984] used fuzzy neural networks (distributed and parallel computing structures employing fuzzy connectives as logic operators) to identify fuzzy relations,
- Takagi ([Tak, 1985]) and Sugeno et al. ([Sug, 1991], [Sug, 1993]) investigated the identification of fuzzy systems where the consequent of the fuzzy rule is a linear input-output relation that is intended to model the operators' control actions when operating a system (e.g. a driver when parking a car),
- Delgado and Gonzalez ([Delg, 1993]) introduced a method of calculating frequencies in fuzzy domains in order to capture the frequency of appearance of some kind of patterns in raw data obtained from on-line measurements on the system input and output variables

(see also [Sest, 1991], [Lee, 1991], and [Nom, 1992a], among others, for studies on automatic knowledge acquisition techniques).

In classical system identification theory [Ljung, 1987], a distinction between modelling and system identification is made. *Modelling* is the process of splitting up the system into several subsystems whose properties are well understood. *System identification* is the process where the model is derived from a given input-output data set. Thus, *fuzzy modelling* is the process of building up a fuzzy model based on user knowledge only, while *fuzzy system identification* is the process of identifying the structure and parameters of the fuzzy model based on a given input-output data set. Bastians in [Bast, 1995] defines structure and parameter identification as:

> "The *structure identification* of a fuzzy model consists of the input variable identification and the rule identification. The input variable identification consists of a) the identification of the rule type used by the fuzzy model to represent a given input-output data relation and b) the identification of the rule structure. The latter task is the process of selecting the input and output variables involved in each individual rule, and determining the rule base size, that is, the number of rules.
>
> The *parameter identification* of a fuzzy model consists of a) the rule parameter identification and b) the mapping parameter identification. The rule parameters of a fuzzy model are all the parameters related directly to the interpretation of a fuzzy rule, as there are the membership functions, the aggregation operator and the implication function. The mapping parameters are those parameters which are related to the mapping of a crisp set to a fuzzy set, and vice-versa, namely the fuzzification and the defuzzification."

When knowledge on the process under study is not available from experts, fuzzy logic systems identification is achieved using on-line measurements on the input and output variables of the fuzzy system. These measurements are referred also as "input-output data" or simply "examples".

At this point a remark must also be made on the terminology used in this thesis. The term system is used as an abbreviation for fuzzy system and not to refer to the actual process whose dynamics are aimed to be captured by performing on-line measurements on monitored variables.

In the following section, a method for the identification of patterns in input-output data using the notion of “frequencies” in fuzzy domains developed by Delgado and Gonzalez ([Delg, 1993]) is presented. This method will be used to associate to a fuzzy rule an uncertainty measure expressing how the rule describes a set of examples. After that, a learning method used to identify fuzzy logic systems from raw data is described.

3.2.1 Frequencies on fuzzy domains

Let X be a fuzzy variable taking values in fuzzy domain D and A_i, $1 \le i \le r$, the fuzzy sets of D defined in universe U. Each of the fuzzy sets of D has its meaning given by membership functions $A_i(.) : U \to [0,1],\ 1 \le i \le r$. It is further assumed that the fuzzy sets of D are normal ($\exists u \in U,\ A_i(u) = 1,\ 1 \le i \le r$), and satisfy the following completeness condition (convexity):

$$\forall u \in U \quad \exists A_i \in D, \quad A_i(u) \ge \delta, \quad \delta > 0, \quad 1 \le i \le r \tag{3.4}$$

with $\delta \in (0,1]$ a threshold value representing the *completeness degree* of the referential fuzzy domain D.

Under these assumptions, Delgado and Gonzalez ([Delg, 1993]) define frequency in a fuzzy domain as:

Definition 3.1 (*Frequency*): “Given a set of examples $E_h = \{e_1, e_2, ..., e_h\}$, the frequency of any subset A of D through the set E_h is given by:

$$f_h(A) = \begin{cases} 0 & , \ \ if \ \ A = \varnothing \\ \sum_{k=1}^{h} \dfrac{\Pi_A(e_k)}{h}, & otherwise \end{cases}, \tag{3.5}$$

where:

- $\Pi_A(e_k) = \sup_{F \in A} \Pi_F(e_k)$ and

- $\Pi_F(e_k)$ is the normalised non-negative compatibility degree between fuzzy set *a* and the example e_k given by:

$$\Pi_F(e_j) = \frac{\mu_F(e_j)}{\sup_{F\in D}\mu_F(e_j)}, \tag{3.6}$$

It is also assumed that $\sup_{F\in D}\mu_F(e_j) > 0,\ j = 1,\ldots,h$, i.e., all examples are covered by domain *D*." ([Delg, 1993])

The compatibility degree, as defined in (3.6), can be interpreted as the *possibility* measure of *A* given the evidence e_j ([Delg, 1993]).

It can be shown (see [Delg, 1992] for proofs) that the frequency $f_h(\cdot)$, defined in (3.5), is a *plausibility* measure. Its dual measure, $g_h(\cdot)$, defined as

$$g_h(A) = 1 - f_h(\overline{A}) = 1 - \sum_{j=1}^{h}\frac{\sup_{F\notin A}\Pi_F(e_j)}{h} \tag{3.7}$$

is a *belief* measure[1].

---

[1] Given a function $Bel\colon D \to [0,1]$ which assigns to each subset A of D a number in the unit interval [0, 1], the function *Bel* is said to be a *belief measure* if satisfies:

Axiom 1 (boundary conditions): $Bel(A) \geq 0,\ \forall A \subseteq D$ and $Bel(D) = 1$.

Axiom 2 (montonicity): If $A \subseteq B$, then $Bel(A) \leq Bel(B)$.

Axiom 3 (continuity): For every monotonic sequence A_i, $i \in N$ of subsets of *D*, then

$$\lim_{i\to\infty} Bel(A_i) = Bel(\lim_{i\to\infty} A_i)$$

Axiom 4 (superadditivity to every positive integer order *n*): To the second order, this property is expressed as:

$$Bel(A_1 \cup A_2) \geq Bel(A_1) + Bel(A_2) - Bel(A_1 \cap A_2).$$

A *plausibility measure*, *Pl*, is a function that satisfies axioms 1 to 3 and

Axiom 4' (subadditivity to every positive integer order *n*): To the second order, this property is expressed as:

$$Pl(A_1 \cap A_2) \leq Pl(A_1) + Pl(A_2) - Pl(A_1 \cup A_2).$$

It then follows the inequalities:

$$Bel(A) + Bel(\overline{A}) \leq 1, \quad Pl(A) + Pl(\overline{A}) \geq 1,\ \forall A \in \Im(X)$$

The pair of dual measures (f_h, g_h) defined through function $\Pi_F(\cdot)$ constitutes a *belief-plausibility* pair in the sense of Dempster-Shafer's theory ([Delg, 1993], see also [Shaf, 1976], [Demp, 1967] for more details).

Hence, $g_h(A) \leq f_h(A), \quad \forall A \subseteq D$, and the interval $[\alpha, \beta] = [g_h(A), f_h(A)]$ can be interpreted as a measure of the uncertainty about A that lies in the raw data set ([Delg, 1993]). In the next section, a more detailed interpretation of this interval of uncertainty is given.

Once defined the concept of frequency on fuzzy domains, the generalisation to an M-dimensional fuzzy domain is as follows ([Delg, 1993]). Let $D = D_1 \times D_2 \times \ldots \times D_M$ be the cartesian product of domains $D_j = \{a_{1j}, a_{2j}, \ldots, a_{N_j j}\}$, where each domain D_j corresponds to a variable X_j and a_{ij} (i=1, ...,N_j, j=1, ..., M), is a fuzzy set in the universe U_j. Let $E_h = \{e_1, e_2, \ldots, e_h\}$, $e_k = (e_{1k}, e_{2k}, \ldots, e_{Mk})$, k=1,...,h, be a sequence of examples of the multidimensional variable $(X_1, X_2, \ldots, X_M)$ obtained through some sampling or trial process and taking values on $D = D_1 \times D_2 \times \ldots \times D_M$.

Definition 3.2 (*Frequency in M-dimensional domain*). "The frequency of any $B = (B_1, B_2, \ldots, B_M)$, with $B_j \subseteq D_j$, through the set E_h is defined as

$$f_h(B_1, B_2, \ldots, B_M) = \frac{\sum_{k=1}^{h} \Pi_B(e_k)}{h} \tag{3.8}$$

where $$\Pi_B(e_k) = \Pi_{B_1}(e_{1k}) * \ldots * \Pi_{B_M}(e_{Mk}) \tag{3.9}$$

$e_k = (e_{1k}, e_{2k}, \ldots, e_{Mk})$, k=1,...,h and * is a t-norm[2]." ([Delg, 1993]).

and, $$Bel(A) \leq Pl(\overline{A}), \ \forall A \in \Im(X)$$

For more details on belief and plausibility measures see [Klir, 1988] and [Dub, 1988].)

[2] Let I=[0, 1]. A triangular norm, or *t-norm*, is a mapping $* : I \times I \rightarrow I$ that satisfies the following conditions ([Schw, 199]):

A learning procedure needs to update information when new evidences are available; the conditional operator is used to achieve this. Given that f_h (g_h) is a measure (satisfies Axioms 1 to 3 in footnote 1), the conditional definitions in probability theory are going to be applied in the next section to define conditioned plausibility (belief).

A generalisation of the conditioning in probability theory given by Dempster ([Demp, 1967]) can be found in the theory of evidence, where conditional plausibility and belief are defined as:

Definition 3.3 (*Conditional plausibility and belief*):

$$Pl_D(B|A) = \frac{Pl(B,A)}{Pl(A)}, \quad Bel_D(B|A) = \frac{Pl(A) - Pl(\neg B, A)}{Pl(A)}. \tag{3.10}$$

Other authors, have proposed alternative definitions (see also [Moral, 1991], [Nom, 1992a]). In this study, the above conditional definition is chosen. The application of any of the definitions produces similar results ([Delg, 1993]).

3.2.2 Uncertainty of a fuzzy rule

Let us assume, to simplify the description, that the rules have a M-dimensional antecedent and a one-dimensional consequent

---

1. $a * 1 = a$;
2. $a * b = b * a$;
3. $(a * b) * c = a\ (b * c)$;
4. $a * c \le b * c$, whenever $a \le b$,

for any a, b, $c \in I$.

A *t-conorm* (or s-norm) is a mapping $*: I \times I \to I$, that satisfies the conditions 1 to 3 and

5. $a * 0 = a, \quad \forall a \in I$

$$R \equiv if \;\; X \; is \; A_i \; then \; Y \; is \; B_j \tag{3.11}$$

with $X = (X_1, X_2, ..., X_M)$, $A = (A_1, A_2, ..., A_M)$.

We want to associate to each rule R an interval $[\alpha, \beta]$ expressing the information contained in the examples about the proposition "the rule R is a true rule for the system". A direct interpretation of this interval is the usual one for a lower-upper measure model: α and β are, respectively, the lower and upper bound of the evidence contained in the set of examples about the aforementioned proposition ([Delg, 1993]).

Suppose that $E_h = \{e_1, e_2, ..., e_h\}$ is a set of examples where:

- $e_k = (ex_{mk}, ey_k)$, k=1,...,h;
- $ex_{mk} = (ex_{1k}, ex_{2k}, ..., ex_{Mk})$;
- ex_{mk} and ey_k are examples for variables X and Y, respectively;
- ex_{mk} and ey_k belong to domains D_j (j=1, ..., M) and S, respectively.

The definition of the frequency of a fuzzy rule is ([Herr, 1995]):

> **Definition 3.4** (*Frequency of a fuzzy rule*) The *frequency* of any fuzzy rule R through the set of examples E_h can then be defined as
>
> $$\Psi_E(R) = \frac{\sum_{k=1}^{h} R(e_k)}{h} \tag{3.12}$$

with:

- h is the total number of examples in E_h;
- the compatibility degree between rule R and example e_k is
$$R(e_k) = \Pi_A(ex_{mk}) * \Pi_B(ey_k) \tag{3.13}$$
- $\Pi_A(ex_{mk})$ is the compatibility degree between the rule's antecedent and example e_k, given by (3.9);
- $\Pi_B(ey_k)$ is the compatibility degree between the rule's consequent and example e_k, given by 3.6.

Applying the conditional operator (definition 3.3) and the notion of frequency in M-dimensional domains (definition 3.2), the interval of uncertainty $[\alpha, \beta]$, is calculated for each rule R using ([Delg, 1993]):

$$\alpha = g_h(B_j|A_i) = 1 - \frac{\sum_k \Pi_{A_i}(ex_{mk}) * \Pi_{\overline{B_j}}(ey_k)}{\sum_k \Pi_{A_i}(ex_{mk})}, \tag{3.14}$$

$$\beta = f_h(B_j|A_i) = \frac{\sum_k \Pi_{A_i}(ex_{mk}) * \Pi_{B_j}(ey_k)}{\sum_k \Pi_{A_i}(exm_k)}. \tag{3.15}$$

The property "to be a true rule" for a system may be measured by a crisp (Boolean) value. In a fuzzy domain it is more accurate to say that each possible rule in R will fulfil such property with a degree (degree of truth) between 0 and 1. This degree can be interpreted as the strength of the rule's implication and behaves as the "weight" associated with the fuzzy rule. Delgado and Gonzalez ([Delg, 1993]) refer to it as a measure of the "adequateness" between a certain rule and a set of examples and formulate this by associating with each rule, R, an interval $[\alpha, \beta]$, representing a *measure of uncertainty* about the statement: "rule R is a true rule".

The compatibility degree between an example, e, and a rule, R, or, equivalently, the ability of a rule R to describe example e, is given by the measure $\Pi_.(.)$ defined in (3.6): $\Pi_A(e)$ and $\Pi_B(e)$ measure the compatibility degree between example e and the antecedent, A, and consequent, B, of rule R, respectively. $\Pi_{\overline{B}}(e)$, measures the compatibility degree between example e and the complement of the consequent of rule R (best matching value between example e and any other consequent of domain S different from B) and is given by:

$$\Pi_{\overline{B}}(e_k) = \frac{\sup_{B' \neq B} \mathrm{B}'\left(ey_k\right)}{\sup_{F \in S} F\left(ey_k\right)} \tag{3.16}$$

Summarising, the application of the concept of frequency in fuzzy domains and its inclusion in the theory of evidence (conditional belief and plausibility) has been described in previous paragraphs. The definitions have been used to obtain a formulation for the calculus of the lower and upper bounds of the uncertainty interval associated with the rule. In the next section, these concepts are going to be applied on a:

- methodology of "learning from examples" in fuzzy domains;
- fuzzy inference model where the uncertainty contained in the rule is "propagated" from premises to conclusions.

3.2.3 Learning from examples

Learning from examples is the most thoroughly studied discipline of Machine Learning. The idea is to teach a computer a concept by means of its positive and negative examples. Concepts can be disjoint or overlapping, logical or probabilistic, crisp or fuzzy, stable or time-varying. The positive examples are representative and provide a good description of the concept. The negative examples allow to distinguish between instances and non-instances of the concept; they represent "near-miss" of the concept.

Before introducing the method of learning from examples, some basic definitions are given:

Definition 3.6 An example e_k is said to have an *effect* on rule R (in the sense of contributing for the corresponding interval of uncertainty $[\alpha,\beta]$), if $\Pi_A(e_k) > 0$, i.e., e_k matches to some extent the antecedent of R.

Using $\Pi_B(e_k)$ and $\Pi_{\overline{B}}(e_k)$, the examples having effect on R can be split in two categories:

Definition 3.7 (*Positive examples*) Examples such that $\Pi_A(e_k) > 0$ and $\Pi_B(e_k) > 0$. In other words, positive examples match both the antecedent and consequent of R.

Definition 3.8 (*Negative examples*) Examples such that $\Pi_A(e_k) > 0$ and $\Pi_B(e_k) = 0$. In other words, negative examples match the antecedent of *R* but do not match its consequent.

A negative example e_k for a rule *R* matches better with some other rule having the same antecedent but having a consequent value different from B. Also, it can be easily proved that if $\Pi_B(e_k) = 0$ then $\Pi_{\overline{B}}(e_k) = 1$ ([Delg, 1993]).

Although this classification could help to decide which rules represent the examples, the above definitions are over simplified, an example e_k is either positive or negative and, thus, it is not possible to distinguish between strongly positive examples to the rule (e.g. $\Pi_A(e_k) = 1$ and $\Pi_B(e_k) = 1$) and weakly positive examples (e.g. $\Pi_A(e_k) = 0.2$ and $\Pi_B(e_k) = 0.001$). The alternative is to consider "positively" and "negatively" to be vague properties and define the fuzzy sets for positive and negative examples.

Hence, the set of examples that have an effect on rule *R* is defined as the fuzzy set *E*(*R*), defined in universe E_h, as:

$$E(R) = \left\{ \left(e_k,\ \Pi_A(e_k)\right) \mid e_k \in E_h \right\}. \tag{3.17}$$

that is, example e_k has a membership value of $\Pi_A(e_k)$ in fuzzy set *E*(*R*).

Similarly, the fuzzy set of positive examples of *R*, defined in universe E_h, is given by:

$$E^{+}(R) = \left\{ \left(e_k, \Pi_A(e_k) * \Pi_B(e_k)\right) \mid e_k \in E_h \right\}. \tag{3.18}$$

and the fuzzy set of negative examples of set of *R* is defined as:

$$E^{-}(R) = \left\{ \left(e_k, \Pi_A(e_k) * \Pi_{\overline{B}}(e_k)\right) \mid e_k \in E_h \right\} \tag{3.19}$$

where symbol * stands for a t-norm operator. In the following, the t-norm chosen is the *Lukasiewicz* t-norm defined by $a*b = \max\,(a+b-1, 0)$, because it allows to concentrate the evidence in the most representative rules (for other definitions of t-norms see for instance [Fodor, 1993]).

An example, e_k, such that $\Pi_A(e_k)$=1 and $\Pi_B(e_k)$=1 has a membership of 1 to $E^+(R)$, and an example, e_k, such as $\Pi_A(e_k)$=0.2 and $\Pi_B(e_k)$=0.001 has a membership of 0 to $E^+(R)$. It is easy to show that an example e_k such that $\Pi_A(e_k)$=1 and $\Pi_B(e_k)$=0 belongs to $E^-(R_i)$ with a membership degree of 1, that is, e_k ought to match with some other rule having the same antecedent but different consequent.

In case of a single variable consequents, the relation between $\Pi_B(e_k)$ and $\Pi_{\overline{B}}(e_k)$ is such that $max\{\Pi_B(e_k), \Pi_{\overline{B}}(e_k)\}$=1, and hence the following result

$$max\{\Pi_A(e_k) * \Pi_B(e_k), \Pi_A(e_k) * \Pi_{\overline{B}}(e_k)\} = \Pi_A(e_k)\,. \qquad (3.20)$$

Therefore, if the *max* t-conorm is used to define the union between two fuzzy sets, then the following equality holds:

$$E(R) = E^+(R) \cup E^-(R). \qquad (3.21)$$

The *uncertainty interval* $[\alpha,\beta]$, can be calculated for each rule from the fuzzy sets (3.17) to (3.19). Let us denote by $|A| = \sum_{u \in U} A(u)$, the *cardinality* of a fuzzy subset. The quotient

$$\left|E^+(R)\right| \big/ \left|E(R)\right| \qquad (3.22)$$

represents the proportion of positive examples (among those having some effect on the rule) and coincides with β. Therefore, the upper measure of the rule's uncertainty quantifies the proportion of positive examples.

Similarly, the quotient

$$\left|E^{-}(R)\right| \Big/ \left|E(R)\right| \tag{3.23}$$

represents the proportion of negative examples and coincides with (1 - α). It measures to what extent R is rejected (negatively supported) by the examples.

Therefore, the interval $[\alpha, \beta]$ has the following interpretation: each rule is affected by the proportions of the positive examples β (that is, examples matching the rule to a certain degree) and the fuzzy percentage of non-negative examples α (that is, one minus the percentage of examples matching the complement of the rule).

A rule R with an uncertainty interval [0,1] has 100% positive examples and 100% negative examples, that is, 100% examples that do not affect the rule (we have no information about the rule). Such an interval represents the "ignorance" about the sentence "rule R is a true rule" and can be used as the initial evidence of the rule. If the interval is [1, 1], the positive examples are 100% and the negative examples are 0%. For this case, the rule describes the data set accurately. The interval [0, 0] represents 0% positive examples and 100% negative examples; in this case all the information relative to the rule is concentrated in other rules with the same antecedent but different consequent.

3.2.3.1 Learning algorithm

After presenting the interpretation of the method of learning from examples and how to calculate the interval of uncertainty associated with each rule, the learning algorithm suggested by Delgado et al ([Delg, 1993]) is as follows:

Table 3.1 Learning-from-examples algorithm.

1. Collect a set of examples;
2. Compute the interval of uncertainty of each rule from the set of examples using formulas (3.22) and (3.23), that is, for each rule R, calculate the values:

   $a_0 = \left|E^{+}(R)\right|$, $\quad b_0 = \left|E^{-}(R)\right|$ $\quad$ and $c_0 = \left|E(R)\right|$

and initialise to uncertainty interval as

$$current\_interval\,(R) = \left[1 - \frac{b_0}{c_0}, \frac{a_0}{c_0}\right],$$

3. Update the interval of uncertainty when new examples are available, that is, for the new set of examples, calculate the values:

   $x = |E(R)|$, $y = |E^{+}(R)|$ and $z = |E^{-}(R)|$,

   and update the interval of uncertainty using

$$new\_interval\,(R) = \left[1 - \frac{b_0 + z}{c_0 + x}, \frac{a_0 + y}{c_0 + x}\right].$$

Now, each of the fuzzy rules in the rule base has an associated interval of uncertainty, $[\alpha, \beta]$, representing the evidence contained in the examples about the statement "rule R is a true rule", that is, for each fuzzy rule R we have:

$$R \equiv if\ \ X\ is\ A_i\ then\ Y\ is\ B_j \quad [\alpha, \beta] \tag{3.24}$$

After this and quoting Delgado et al in ([Delg, 1993]):

> "[…] any system identification is done with prediction […] purposes. [Thus], inference is to be made on the rules and the question about how to combine and propagate the interval [of uncertainty] we have introduced arises at once."

In the next section, the inference model proposed by Campos et al ([Camp, 1992]) and used in this studies is explained.

3.3 Fuzzy Inference Model

A fuzzy inference model allows to infer from vague or fuzzy premises, that is, given the inference problem stated in figure 3.5 below, the inference model allows to calculate an inferred output B^* from a fuzzy rule (3.25) and an input A^*. In contrast to the classical modus ponens, rule (3.25) allows to use fuzzy predicates *A, B* and A^*. Moreover, A^* is not required to be identical to *A*.

(fuzzy rule) If	X is A then	Y is B	(3.25)
(input)	X is A^*		
(inferred output)		Y is B^*	

where:
- X and Y are variables on domains D_1, D_2, respectively;
- *A, B,* are fuzzy sets of the domain of fuzzy variables X and Y, respectively;
- B^* is the fuzzy set obtained from fuzzy rule (3.25) and the restriction A^* on X.

Figure 3.5 Fuzzy inference problem

The method, introduced initially by Zadeh ([Zad, 1973]), proposes the following solution for the inference problem stated above

$$B^*(s) = \sup_r \{ A^*(r) \wedge \left(1 - A(r) + B(s)\right) \} \tag{3.26}$$

where:
- $\wedge$ stands for the *min* t-norm;
- r is a value of the antecedent universe U_1;
- s is a value of the consequent universe U_2.

Figure 3.6 Zadeh's compositional rule of inference.

Using expression (3.26), the membership function of B^* is given by the projection on U_2 of the intersection of the implication relation H, defined by

$H(r,s) = 1 \wedge (1 - A(r) + B(s))$, and the cylindrical extension on $U_1 \times U_2$ of the membership function A^*. When $A = A^*$, the predicates are crisp and $\wedge$ is the *Lukasiewicz* t-norm (defined by $a{*}b = \max\,(a + b - 1, 0)$), then (3.26) becomes the classical modus ponens, that is, $B = B^*$.

A more general version of the generalised modus ponens given by (3.26) can be obtained if we replace the *min* operator $\wedge$ by an alternative t-norm *, and the particular implication relation H (r, s) by another by implication $A \to B$ (see also [Dub, 1988], [Dub, 1991], [Magr, 1989]), thus obtaining

$$B^*(s) = \sup_r \{\, A^*(r) * (A \to B)(r,s) \,\} \tag{3.27}$$

This section aims to present the extension of the general inference model given by (3.27), proposed by Campos et al ([Camp, 1992] and [Camp, 1993]), in order to be able to reason with uncertain knowledge, that is, to allow to infer not only from fuzzy propositions such as

If X is *Low* then Y is *High*

but also from fuzzy rules expressing uncertain knowledge, as for example,

If X is *Low* then Y is *High*, with uncertainty interval [0.5, 0.8]

The rule "If X is *Low* then Y is *High*" contains no uncertainty apart from its fuzziness, that is, the propositions considered in the rule: "X is *Low*" and "Y is *High*" are fuzzy but there is no doubt about its certainty. The method of learning from examples explained in section 3.2 allows to associate a fuzzy rule with an uncertainty interval (say [0.5, 0.8] in the example above). The interval of uncertainty represents a pair of measures (lower and upper measures) of the uncertainty that lies in the rule about the relation between input and output described by a set of examples. In terms of the inference model, the lower and upper uncertainty measures provide a means of "weighting" the influence of each fuzzy rule on the inferred output value.

Given a rule

$$\text{If } X \text{ is } A \text{ then } Y \text{ is } B \tag{3.28}$$

the extended inference model is based on two main ideas ([Camp, 1993]):

1. The fuzzy rule (3.28) defines a relation among the elements of the sets $U_A=\{A, \neg A\}$ and $U_B=\{B, \neg B\}$.
2. The relation is interpreted as a conditioning represented by means of an uncertainty measure.

In other words, 1. states that the rule does not establish how each element from U_1 and each element from U_2 are related (i.e. it does not define a relation on the Cartesian product of the universe of variable X and the universe of Y: $U_1 \mathrm{x} U_2$), it only establishes the relation between the concepts represented by A, $\neg A$ and B, $\neg B$; 2. distinguishes the interpretation of a fuzzy rule given by this model, from other models that interpret the rule as a material implication ($\neg A \vee B$) ([Camp, 1993]).

An input A^* does not necessarily match any of the rules' antecedent. Bearing in mind 1. and 2., the inference model should translate the information contained in A^* to information about A and $\neg A$. This is achieved by calculating:

- the compatibility degree between input A^* and the antecedent of the rule, A;
- the compatibility degree between input A^* and the negation of the antecedent of the rule, $\neg A$.

These compatibility degrees will be interpreted as uncertainty measures generated by the current input A^*, on the set U_A. The measures on U_A will, then, be transferred, through the fuzzy rule, to an uncertainty measure on U_B using a propagation model.

Hitherto, the inference model enables to obtain an uncertainty measure on U_B. Finally, by combining the membership functions of B and $\neg B$, and the uncertainty values of B and $\neg B$ (given by the uncertainty measure on U_B), the output fuzzy set B^* will be obtained.

3.3.1 Propagation model

The formalism used by Campos et al in ([Camp, 1993]) to represent pieces of uncertain information is using the authors' own words:

> "by means of a class of fuzzy measures [mappings that satisfy Axioms 1 to 3, see footnote 1 in section 3.2.1], namely representable measures (also called lower and upper probabilities[3])"

Let us suppose that the fuzzy variables of inference problem are *X* and *Y*, taking values on $D_x = \{x_1, x_2, \ldots, x_n\}$ and $D_y = \{y_1, y_2, \ldots, y_n\}$, respectively. Let us associate with X a pair of representative measures, $\left((l_x(A), u_x(A)), \quad A \subseteq D_x \right)$, representing our knowledge about the values of *X*.

Through fuzzy rule (3.28), there is also conditional information about the values of *Y*, given that we know the true value of *X*. The conditional information on *Y* given some value $x_i \in D_x$ is modelled using also conditional representative measures $\left((l_x(B/x_i), u_x(B/x_i)), \quad B \subseteq D_y \right), \quad x_i \in D_x$.

Hence, a mechanism is necessary to propagate the information from *X* to *Y*, through a conditional relation between X and Y, in order to obtain on *Y* another pair of lower and upper probabilities representing the knowledge about the value of the variable *Y*.

---

[3] Let P be a family of probability measures on a referential D_X. We may associate a pair of lower and upper probabilities with P : (*l*, *u*), given by

$$\begin{aligned} l(A) &= \inf_{P \in \mathsf{P}} P(A), \qquad \forall A \subseteq D_X \\ u(A) &= \sup_{P \in \mathsf{P}} P(A), \qquad \forall A \subseteq D_X, \end{aligned} \tag{3.29}$$

where "inf" stands for the *infimum* operator and "sup" stands for the *supremum* operator. This defines a pair of ordered fuzzy measures in Sugeno´s sense (see [Lam, 1989] for details).)

In [Camp, 1992] the general solution for this problem was obtained by calculating the upper measure of any subset of D_y . This measure is obtained as the solution of the following linear program problem

$$u_Y(B) = \max \sum_{i=1}^{n} u(B/x_i)\ h_i$$
$$\text{subject to} \qquad (3.30)$$
$$\sum_{x_i \in A} h_i \le u_X(A), \quad \forall A \subseteq D_X$$

Particular cases of this problem can be directly solved without using optimisation techniques, namely the case in which the lower-upper probability measures (l_X, u_X) are Choquet capacities of order two. This is the case in these studies and, thus, the propagated measure $u_Y(B)$ on D_Y can be obtained by calculating the Choquet integral in finite domains[4] (see [Choq, 1953] and [Camp, 1992] for more details).

3.3.2 Description of the inference model

In order to simplify, for ease of understanding, the description of the inference model, firstly, let us consider rules with no associated uncertainty degree, that is, rules of format "If *X* is *A* then *Y* is *B",* where *X* and *Y* are fuzzy variables, and *A* and *B* are fuzzy sets of the domain of *X* and *Y*, defined in universe U_1 and U_2, respectively.

Let $U_A = \{A, \neg A\}$ and $U_B = \{B, \neg B\}$ be two crisp sets representing fuzzy partitions of U_1 and U_2. The basic idea, throughout this explanation, is to replace U_1 and U_2 by the

[4] Given a fuzzy measure *g* on a finite set $D_X = \{x_1, x_2, \ldots, x_n\}$ and a function $f : D_X \to R^+$, such that the *f* satisfies $f(x_1) \le f(x_2) \le \ldots \le f(x_n)$, then the Choquet integral of *f* with respect to the measure *g* is given by:

$$E_g(f) = \sum_{i=1}^{n} f(x_i)\ (\ g(A_i) - g(A_{i+1})\ )\ , \qquad (3.31)$$

where $A_i = \{x_i, x_{i+1}, \ldots, x_n\}$, $A_{n+1} = \varnothing$.)

fuzzy partitions U_A and U_B, respectively, and then to consider uncertainty measures on each fuzzy partition ([Camp, 1993]).

The conditional information comes from a semantic interpretation of the rule
"If *X* is *A* then *Y* is *B"* in the sense that the rule generates two conditional measures on U_B,

$$\left(l(./A), u(./A)\right), \quad \left(l(./\neg A), u(./\neg A)\right)$$

defined by ([Camp, 1993]):

$$\begin{aligned} l(B|A) &= 1 \quad u(B|A) = 1 \\ l(\neg B|A) &= 0 \quad u(\neg B|A) = 0 \end{aligned} \tag{3.32}$$

$$\begin{aligned} l(B|\neg A) &= 0 \quad u(B|\neg A) = 1 \\ l(\neg B|\neg A) &= 0 \quad u(\neg B|\neg A) = 1 \end{aligned} \tag{3.33}$$

Thus, the rule "If *X* is *A* then *Y* is *B*" is interpreted as a conditioning (instead of a material implication); in other words ([Camp, 1993]):

- if *A is true* is known, then we can assert that *B is true* (modelled as the total certainty (3.32));
- if *A is false* then nothing can be inferred about the truthfulness of *B* (modelled as the total ignorance (3.33)).

Moreover, an upper (lower) probability measure on U_A needs to be defined in order to propagate it to *Y* through the conditional information expressed in the rule.

The upper probability measure will be obtained by matching the input A^* with each of the values in U_A, that is, by matching A^* with A and A^* with $\neg A$. A matching mechanism, based on the compatibility degree between two fuzzy sets, *F* and *G*, given by:

$$c(F,G) = \sup_{r \in U} \left\{ F(r) * G(r) \right\} \tag{3.34}$$

is used to obtain the measure on U_A. Although any t-norm, *, could be used in (3.34), the Lukasiewicz t-norm ($a*b = \max(a+b-1,0)$) is chosen, because it satisfies the non-contraction law ($a*(1-a) = 0$, $\forall$a) and, thus, the compatibility degree between two complementary fuzzy sets is going to be zero :

$$c(F,\neg F) = \sup_{r \in U} \left\{ \max\left(F(r) + (\neg F)(r) - 1,0 \right) \right\} = 0 \qquad (3.35)$$

(with $(\neg F)(r) = 1 - F(r)$). This way, the elements in each partition U_A and U_B are going to be incompatible.

The upper measure on U_A induced by the input $A*$ is then, defined as:

$$\begin{aligned} u_X(A/A^*) &= c(A,A^*) = \sup_r \left\{ \max(A(r) + A^*(r) - 1,0 \right\} \\ u_X(\neg A/A^*) &= c(\neg A,A^*) = \sup_r \left\{ \max(A^*(r) - A(r),0 \right\} \end{aligned} \qquad (3.36)$$

(the lower probability measure is given by $l_X(H) = 1 - u_X(\neg H),\ H \in U_A$,). (3.37)

Because the input $A*$ is required to be a normalised fuzzy set, it can be easily proved that $$c(A,A^*) + c(\neg A,A^*) \geq 1 . \qquad (3.38)$$

Given (3.38), u_X and l_X can be interpreted as the upper and lower probability measure on U_A, respectively ([Camp, 1993]).

Therefore, the upper (lower) measure on U_B, u_Y, (l_Y), obtained from the conditional information (3.32) ((3.33)) and the upper (lower) measure on U_A, (3.36) ((3.38)) can be calculated using the Choquet integral with respect to the measure u_x (see footnote 4 for definitions and also [Camp, 1989] and [Lam, 1989] for proof that the pair of probability measures (l_X, u_X) defined in (3.36), (3.37), are Choquet capacities of order two)

The solution of the Choquet integral is ([Camp, 1993]):

$$u_Y(B) = 1, \quad u_Y(\neg B) = c(\neg A, A^*) \tag{3.39}$$

and, using the upper measure on U_B, defined as (3.39), the first solution of the fuzzy inference problem, is ([Camp, 1993]):

(rule)	If X is A then Y is B	
(input)	X is A^*	
(output)	Y is B is $[1-\lambda, 1]$	(3.40)
	Y is $\neg B$ is $[0, \lambda]$,	

where:
- $\lambda = c(\neg A, A^*)$;
- "Y is C is θ" means a proposition at *degree* θ[5]([Zad, 1983]).

Figure 3.7 First solution of the inference model.

If the fuzzy inference output is to be given in terms of a fuzzy set B^*, then the upper measure on U_B and the membership functions of B and $\neg B$ have to be combined, in order to obtain a result, B^*, as an expected value of B and $\neg B$, weighted by the values of the upper measures on U_B through a fuzzy integral (in [Camp, 1993] this is justified by a similar reasoning to the one used to calculate the measure u_Y , using the Choquet integral, as an average of the conditional measure $u(./.)$ weighted by the measure u_X).

In order to be coherent at the membership level with the previously stated, the total ignorance about the universe U_2 (i.e. the fuzzy variable Y has no restrictions) must be expressed as a fuzzy set B^* such that

$$B^*(r) = 1, \forall r .$$

[5] In this study, the uncertainty degree θ is represented by the uncertainty interval $[\alpha,\beta]$ where α and β are the lower and upper probability values, respectively.

This is achieved by defining a fuzzy integral based on a t-conorm operator which makes the fuzzy sets B and $\neg B$ exhaustive (i.e. $B * \neg B = U_2$); more precisely, the bounded sum t-conorm operator, defined as: $a \oplus b = a + b - ab$.

The fuzzy inference result obtained using the Choquet integral based on the sum t-conorm operator (see [Camp, 1993, pp.150 for details), yields:

$$B^*(r) = B(r) + \lambda - B(r) \cdot \lambda \tag{3.41}$$

or, equivalently:

(rule) If X is A then Y is B

(input) X is A^*

(output) Y is B^*

where:

- $B^*(r) = B(r) \oplus \lambda$ (3.42)
- $\lambda = c(\neg A, A^*)$;
- $\oplus$ is the bounded sum t-conorm.

Figure 3.8 Second solution of the inference model.

3.3.3 Inference model properties

The inference model verifies the following properties (see [Camp, 1993] for proofs):

- If $A^*{=}A$ then $B^*{=}B$, in other words, the inference model extends the classical *modus ponens*.

- If $A^* \subseteq A$ then $B^*{=}B$:

If the input A^* is included in A, the compatibility between A^* and $\neg A$ (given by (3.36)) is zero, and a similar result to the previous case is obtained. When the input is more precise than the antecedent, the rule can only infer the consequent, without adding information not contained in the rule (the rule only says "if X is A then Y is B", but it does not say, for instance, "the more X is A, then the more Y is B").

- If $A^* \subseteq \neg A$ then $B^*=U_2$:

  If $A^* \subseteq \neg A$, $\exists r \in U_1 : A^*(r) = 1$ and $A(r) = 0$. Hence, the compatibility between A^* and $\neg A$ is equal to one, and $B^*(.) = B(.) \oplus 1 = 1$.

  In other words, an uncertainty measure on U_B is obtained for this case, representing "total ignorance" on the truthfulness of B^*. Thus, for inputs completely different from A, the inference process gives no information about the universe U_2: and all the elements of the universe of the consequent variable are equally possible.

- If $A \subseteq A^*$ then $\exists a \in [0,1]$, $\forall s \in U_2$, $B^*(s) = B(s) \perp a$, where $\perp$ is a t-conorm.

  This property is suggested by Magrez and Smets ([Marg, 1989]) and it imposes two requirements in the inference model: first, whenever an imprecision appears in the output, each of the elements of the consequent domain should receive the same degree of imprecision, and second, the shapes of the output B^* and the input A^* should be independent. The proposed model verifies the before mentioned conditions considering as the t-conorm the bounded sum t-conorm and $a = c(\neg A, A^*)$.

The proposed inference model is easy to implement and computationally efficient; it only requires the calculation of the compatibility degree between the input A^* and $\neg A$ and to operate this value throughout a t-conorm (bounded sum) with the membership function of B.

3.3.4 Inference with uncertainty

In the following the inference model previously presented is going to be extended, in order to allow to infer from rules of the type:

$$\text{If } X \text{ is } A \text{ then } Y \text{ is } B \text{ is } [\alpha, \beta]. \qquad (3.43)$$

Rule (3.43) generates two conditional measures on U_B (pair of conditional lower and upper measures) given by ([Camp, 1993]):

$$\begin{array}{ll} l(B|A) = \alpha & u(B|A) = \beta \\ l(\neg B|A) = 1-\beta & u(\neg B|A) = 1-\alpha \end{array} \tag{3.44}$$

$$\begin{array}{ll} l(B|\neg A) = 0 & u(B|\neg A) = 1 \\ l(\neg B|\neg A) = 0 & u(\neg B|\neg A) = 1 \end{array} \tag{3.45}$$

Propagating the measure (3.36) through these conditional measures, a measure on U_B is obtained as follows (see [Camp, 1993] for details):

$$\begin{array}{l} u_Y(B) = \beta \oplus \lambda \\ u_Y(\neg B) = (1\text{-}\alpha) \oplus \lambda \end{array} \tag{3.46}$$

where $\lambda =$ c($\neg A$, A^*).

The output B^* produced by combining (3.46) and the membership function of B and $\neg B$ is:

$$B^*(s) = \left(B(s) \oplus \lambda \oplus (1-\alpha)\right) - \left(B(s)(1-\beta)(1-\lambda)\right) \tag{3.47}$$

The previous equation defines a non-normalised fuzzy set for $\beta \neq 1$. This imposes a restriction in the use of the inference model if the output (3.47) is going to be used as the input for another rule (forward chaining), because the propagation model requires the input fuzzy sets to be normalised. In this study, rules are going to be fired in parallel, and, therefore, the last comment does not apply. Nevertheless, and following Gonzalez suggestions, this research has decided to divide rule (3.43) in two rules in order to obtain a normalised fuzzy set as the output of the inference process for each rule. The division just aims for a better understanding of the influence of each of the bounds of uncertainty associated with rule (3.43) in the output fuzzy set, rather then by simply looking at expression (3.47).

Hence, rule (3.43) is divided in two rules in the following way:

- *Rule 1*: Imposing $\beta = 1$ in (3.43) gives

"If X is A then Y is B is α" (3.48)

and the conditional measure is now defined as:

$$\begin{aligned} l(B|A) &= \alpha \quad u(B|A) = 1 \\ l(\neg B|A) &= 0 \quad u(\neg B|A) = 1-\alpha \end{aligned} \tag{3.49}$$

with (3.39) unchanged. The inferred output fuzzy set is (see also figure 3.9, where a graphical interpretation of the inference model applied to rule 1 is shown in the top graphic, for trapezoidal fuzzy sets):

$$B_1^*(s) = B(s) \oplus \left(\lambda \oplus (1-\alpha)\right) \tag{3.50}$$

- *Rule 2*: Imposing $\alpha = 1 - \beta$, in the rule (3.48) gives

"If X is A then Y is B is $(1 - \beta)$" (3.51)

and the conditional measure is now defined by:

$$\begin{aligned} l(B|A) &= 1-\beta \quad u(B|A) = 1 \\ l(\neg B|A) &= 0 \qquad u(\neg B|A) = \beta \end{aligned} \tag{3.52}$$

with (3.39) unchanged. The output fuzzy set generated is (see also the middle graphic in figure 3.9):

$$B_2^*(s) = \left(1 - B(s)\right) \oplus \left(\lambda \oplus \beta\right) \tag{3.53}$$

Finally, the output B^* is obtained by combining outputs (3.50) and (3.53) with the Lukasiewicz t-norm (see also bottom of figure 3.9)

$$B^*(s) = B_1^* * B_2^* \tag{3.54}$$

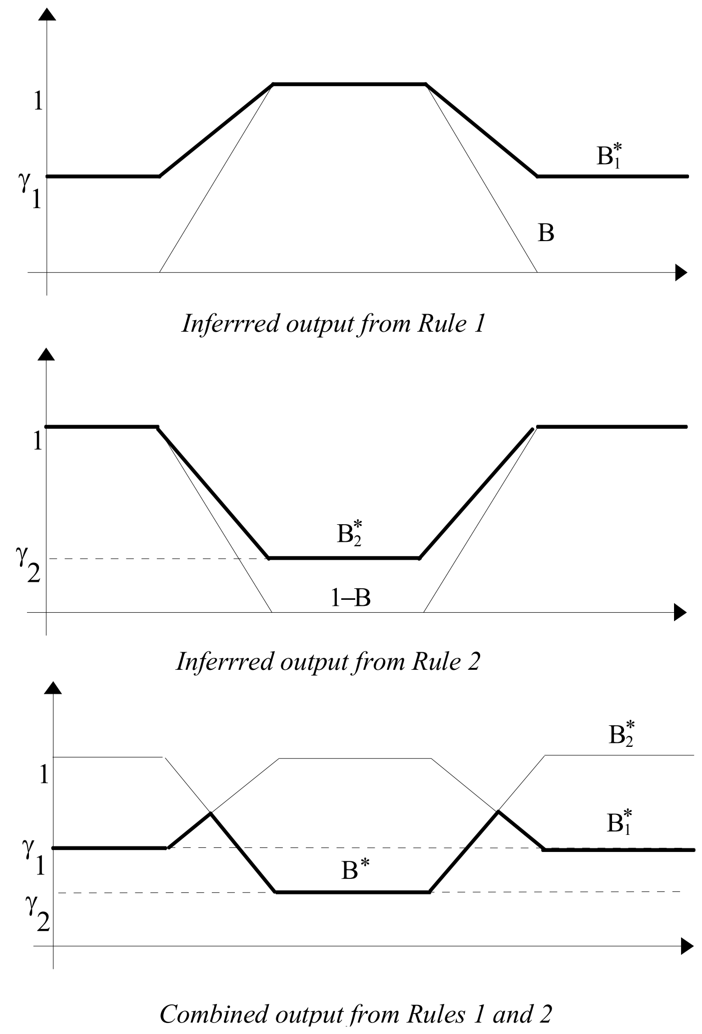

Figure 3.9 Inference model from rules 1 (3.48) and rule 2 (3.51); $\gamma_1 = \lambda \oplus (1-\alpha)$, $\gamma_2 = \lambda \oplus \beta$.

The output fuzzy set B* obtained in (3.54) combining the fuzzy sets B_1^* and B_2^* (resulting from fuzzy rules (3.48) and 3.51), respectively) with the Lukasiewicz t-norm, is exactly the same as (3.47).

$$B^*(s) = B_1^* * B_2^* = \max(B_1^* + B_2^* - 1{,}0)\,,$$

but because $B_1^*(s) \geq 0$ and $B_2^*(s) \geq 0$, $\forall s$ and α, β, $\lambda \geq 0$, comes:

$$B^*(s) = B_1^*(s) + B_2^*(s) - 1 = B_1^*(s) - (B(s)(1 - \lambda - \beta + \lambda\beta)$$

and, finally, (3.47)

$$B^*(s) = \left(B(s) \oplus \lambda \oplus (1-\alpha)\right) - \left(B(s)(1-\beta)(1-\lambda)\right)$$

3.3.5 Inference with connectives in premises

Let us consider the following fuzzy inference problem

(rule)	*If* X_1 *is* A_1 *and* X_2 *is* A_2 *and* ... *and* X_n *is* A_n	*then* *Y is B*	(3.53)
(input)	X_1 *is* A_1^* *and* X_2 *is* A_2^* *and* ... *and* X_n *is* A_n^*		(3.54)
(inferred output)	*Y is B**		

where:

- X_i are variables on universe U_i (i=1, …,n), Y is a variable on reference set S,
- A_i are fuzzy sets on fuzzy domains D_{X_i}, and B is a fuzzy set on D_Y;
- A_i^* are the inputs and B^* is the inferred fuzzy set.

Figure 3.10 Inference problem with conjunctions in premises

If the above fuzzy inference problem is rewritten with:

$$X=(X_1, X_2, ..., X_n),\ A=(A_1, A_2, ..., A_n) \text{ and } A^* = \left(A_1^*, A_2^*, \ldots, A_n^*\right)$$

then a similar inference problem to the one described in figure 3.5 is obtained. All that is left to be done, is to define a measure on U_A and a conditional measure representing rule (3.53), that is, to define the compatibility between the conjunction of A_i and the conjunction of A_i^*, $c(A, A^*)$.

Considering that:

1. the greater the compatibilities between each A_i and A_i^* (given by $c(A_i, A_i^*)$) are, the greater the compatibility between vector A and A^* is, because these factors denote a conjunction of facts;

2. if there is an i (i=1, …,n), such that the compatibility between A_i and A_i^* is low, the compatibility degree between A and A* should be low, too;

it seems appropriate to calculate $c(A, A^*)$ as a function of $c(A_i, A_i^*)$ through a conjunctive operator (e.g. a t-norm); the minimum t-norm was chosen because it satisfies requirements 1. and 2. above.

The compatibility between A and A* is, thus:

$$c(A, A^*) = \min_i \{c(A_i, A_i^*)\} \tag{3.57}$$

where $c(A_i, A_i^*)$ was defined in (3.34).

The negation of input vector A can be interpreted as the disjunction of the negations of each input A_i. Thus, the compatibility degree between $\neg A$ and A*, $c(\neg A, A^*)$, can be defined through a disjunctive operator (e.g. a t-conorm); the maximum t-conorm was the t-conorm chosen.

The compatibility between A and A*, $c(\neg A, A^*)$ is, thus:

$$c(\neg A, A^*) = \max_i \{c(\neg A_i, A_i^*)\} \tag{3.58}$$

Hence, from (3.57) and (3.58), a measure on the antecedent domain is defined as

$$\begin{aligned} u_x(A/A^*) &= c(A, A^*) \\ u_x(\neg A/A^*) &= c(\neg A, A^*) \end{aligned} \tag{3.59}$$

It can be easily proved that ([Camp, 1993])

$$u_x(A/A^*) + u_x(\neg A/A^*) \geq 1$$

because if each A_i is a normalised fuzzy set, then $c(A_i, A_i^*) + c(\neg A_i, A_i^*) \geq 1, \quad \forall i$. Thus, u_x can be interpreted as an upper probability measure.

Using the above measure, the conditional measure defined by (3.32) and (3.33) for rules without uncertainty and by (3.44) and (3.45) for rules with uncertainty, and the propagation model defined in section 3.2.2, the result of the inference model for conjunctions in premises is (3.42) for certain rules and (3.47) for uncertain rules, with

$$\lambda = \max_i \lambda_i, \quad \lambda_i = c(\neg A_i, A_i^*). \tag{3.60}$$

(rule)	*If X is A_1*	*or*			
	X is A_2	or …	or		
	X is A_m		*then*	*Y is B*	(3.61)
(input)	*X is A**				
(inferred output)			*Y is B**		

where X=(X_1, X_2, ..., X_n), A_s=(A_{s1}, A_{s2}, ..., A_{sn}), s = 1,…,m and $A^* = \left(A_1^*, A_2^*, \ldots, A_n^*\right)$.

Figure 3.11 Inference problem with disjunction in premises.

Similarly, for a fuzzy inference problem with disjunction in premises, such as the one shown in figure 3.11, a solution can be obtained as for the problem with conjunctions in premises shown in figure 3.10, by replacing the min t-norm operator in (3.57) with the max t-conorm operator and the max t-conorm operator in (3.58) with the min t-norm operator. Hence, a measure on the antecedent domain is obtained as:

$$\begin{aligned} u_X(A / A^*) &= c(A, A^*) = \max_{s=1..m} \{c(A_s, A^*)\} \\ u_X(\neg A / A^*) &= c(\neg A, A^*) = \min_{s=1..m} \{c(\neg A_s, A^*)\} \end{aligned} \tag{3.62}$$

Using measure u_{x}, the conditional measure defined by (3.32) and (3.33) for rules without uncertainty and by (3.44) and (3.45) for rules with uncertainty, and the propagation model defined in section 3.2.2, the result for inference problem (3.60) is (3.42) for certain rules and (3.47) for uncertain rules, with

$$\lambda = \min_i \lambda_i, \quad \lambda_i = c(\neg A_i, A_i^*). \tag{3.63}$$

3.4 Conclusions

The method of identification of fuzzy systems from examples, presented in section 3.2, enables to associate with each fuzzy rule an interval representing the lower and upper bounds of uncertainty in the truthfulness of the fuzzy rule, given the evidence supported by a set of examples (raw data consisting of on-line measurements on the fuzzy system input and output variables).

The inference model described in section 3.3 extends Zadeh's compositional rule of inference in order to infer from fuzzy rules with an associated uncertainty interval. Furthermore, the inference model extends the traditional "modus ponens" used in Boolean logic and satisfies the properties of a fuzzy inference model.

The learning algorithm described in section 3.2.3 and the inference algorithm for a single fuzzy rule described in sections 3.3.4, 3.3.5 are efficient in running time and storage requirements. The learning algorithm can be used in training mode, at an initial stage and in adaptation mode, after collecting new examples. The inference algorithm for a fuzzy system of *n* rules is going to be presented in the next chapter, adapted to the particular fuzzy rule format adopted in these studies.

The model of frequencies in fuzzy domains and the propagation model, described in sections 3.2.1 and 3.3.2, respectively, are flexible in the choice of the operators, that is, some other alternatives could be considered within the model, without affecting its basis. Let us briefly list some of them:

- the type of matching used in the model of frequencies in fuzzy domain (section 3.2.1) was based on the Lukasiewicz t-norm but other t-norms could have been used instead (for example, the minimum or the product t-norms);
- the propagation model presented in section 3.3.1 is based on the integration of conditional upper measures ($u(\cdot|A)$, $u(\cdot|\neg A)$) with respect to a marginal upper

measure ($u_Y(\cdot)$). Marginal lower measures could also have been chosen, but this would entail a change in the underlying concept of conditioning. Moreover, the Choquet integral was chosen for the integral operator, because the lower-upper measures ($l_X(.), u_X(.)$) are Choquet capacities of order two, but the choice could have fallen upon other integrals, such as the Sugeno integral ([Camp, 1993]).

The method of identifying fuzzy systems from examples will be applied in chapter 6 to an algorithm that automatically designs a fuzzy logic based system given a set of training examples and tunes the fuzzy rules obtained every time a set of testing examples is available (adaptation mode). Before that, chapter 5 presents the structure of the fuzzy logic based CAC (FCAC) proposed in this thesis and also a detailed description of the inference algorithm used in these studies, based on the propagation model presented in this chapter.

4 Fuzzy CAC structure

The selection of the fuzzy variables used in the fuzzy logic decision support system to CAC (in the following referred as FCAC) aimed at identifying the main traffic factors that influence the statistical multiplexing gain and the network performance in terms of cell losses. Ovural in [Onvur, 1994] suggests the following factors, for each of the multiplexed connections (see also figure 4.1):

1. average bit rate to peak bit rate ratio;
2. peak bit rate to link capacity;
3. burst length to buffer size ratio.

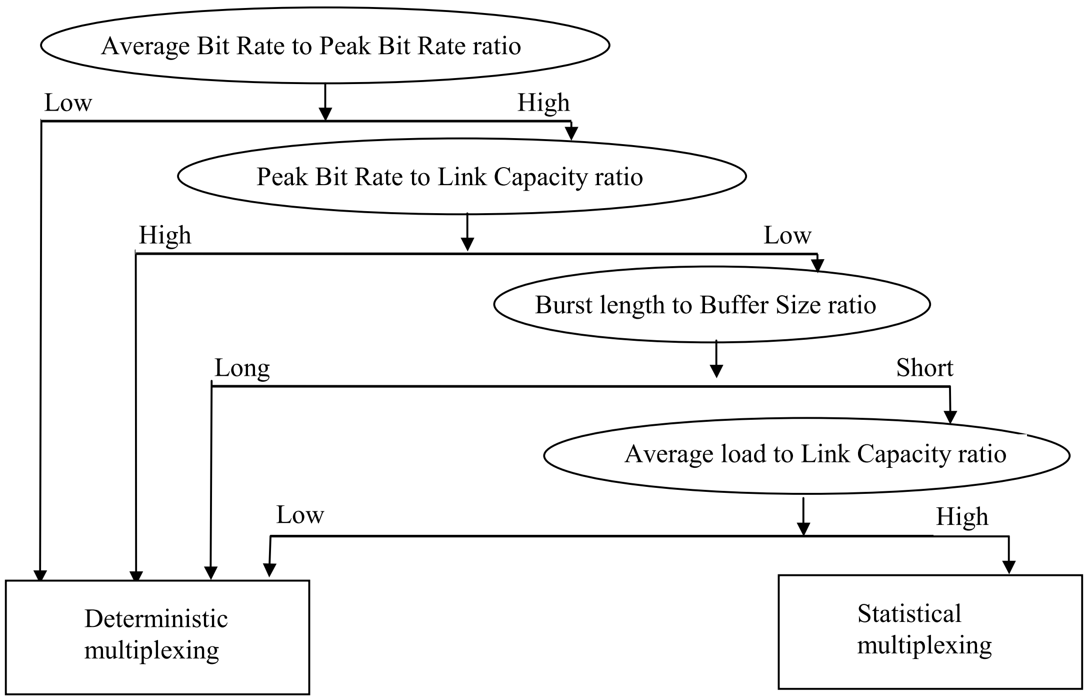

Figure 4.1 Multiplexing factors (modified from [Onvur, 1994])

Factors 1 and 2 quantify the resource consumption in terms of the bandwidth required by the connections (time resource) and 3 quantifies the resource consumption in terms of memory storage, i.e., the output buffer storage requirements (space resource). Factor

1 is a global factor and quantifies the network utilisation. This thesis proposes to add to this set of three factors another factor, the mean load (given by the sum of the mean bit rates of the multiplexed connections) to the link capacity ratio. This additional factor is a global factor and estimates how much bandwidth (time resource) is already allocated to the existing connections.

The identification of the traffic scenario through the before mentioned set of 4 traffic factors, is an attempt to capture the traffic variations in the long term (rate variations) and in the mean term (burst size variations) by relating the source traffic parameters, peak, mean rate and mean burst length to the available network resources expressed in terms of link capacity and output buffer size.

If in future, another traffic factor is identified whose addition to the current model of 4 factors will improve the characterisation of a specific traffic scenario, this factor can be easily added to the current model. But one should always bear in mind that it is not by increasing the number of parameters but by choosing the key parameters that contribute for the measured cell loss value, that a better traffic model can be achieved. Some of the CAC approaches proposed in the literature, such as the convolution approach ([Fabr, 1994]) and the neuro-fuzzy CAC ([Font, 1996]) are based on a bufferless model and, therefore, do not consider factor 3. This study considers that the increase in the complexity of the model, by considering factor 3, can be beneficial, if it contributes for a more accurate prediction, when the size of the output buffers is such that can absorb not only the cell level variations but can also queue partly or totally a burst of consecutive cells.

In the following, a more detailed description of the fuzzy linguistic variables selected for the antecedent and consequent of the rules is presented as well as the format of the fuzzy rules and the shapes of the associated fuzzy sets.

4.1 Rule structure

The rule structure chosen to represent the relation between input and output fuzzy variables is

$$R_i: \quad \text{If } X_1 \text{ is } A_{1i} \text{ and ... and } X_4 \text{ is } A_{4i} \quad \text{then} \quad Y \text{ is } B_i \tag{4.1}$$

where R_i denotes the i-th rule, X_j (i=1,...,4) are input fuzzy variables on universe [0, u_{max}] and Y is the output fuzzy variable defined on universe [0, v_{max}] and A_{ji} (j=1,...,4) and B_i are fuzzy sets of the fuzzy domains of variables X_j and Y, respectively.

4.2 Definition of the fuzzy linguistic variables

The definition of the input fuzzy (linguistic) variables is the following:

- X_1 is the aggregate mean bandwidth offered to the link (*Mean Offered Load*) and linguistically expresses the degree of utilisation of the system.

- X_2 is the aggregate ratio of the weighted mean to peak bandwidths (*Mean to Peak Ratio*) and linguistically expresses how close the behaviour of each of the connections is to that of a constant bit rate (CBR) connection, for which mean and peak bit rates are equal.

- X_3 is the *Relation between Peak rate and Link Capacity* and linguistically expresses how much bandwidth (time resource) the connections require in terms of peak bit rate.

- X_4 is the *Mean Buffer Allocation* and linguistically measures the requirements of the connections in terms of buffer space (space resource). This will have an influence on the amount of cells lost in case the sum of the bandwidths for each connection exceeds the link capacity and the excess cells have to be stored in the output buffer before being served.

The output linguistic variable, Y, represents the maximum predicted *cell loss ratio* (ratio of cells lost per cells sent) per connection to be expected for a multiplexed traffic scenario consisting of the new connection and the existing ones.

4.3 Definition of the fuzzy variables mappings

In order to match the crisp input values with the fuzzy variables in the antecedent of the rules, mappings need to be defined so that the input values are translated into values of the universe of the fuzzy terms (fuzzy sets) of the domain of each variable. This involves *normalising* each input to the maximum value that it can assume in reality, so obtaining a value in the interval [0, 1] and then *scaling* the normalised value to the maximum value of the universe of the correspondent variable.

The normalisation and scaling of the crisp inputs to match the fuzzy variables in the rule's antecedent (fuzzification) is given as follows:

- The crisp input, x_1, that matches variable X_1, the *Mean Offered Load*, is obtained by normalising the sum of the mean bit rates of all connections to the link capacity, C, and then, scaling the result to $u_{\max}$, where $u_{\max}$ is the upper bound of the universe of variable X_1. Thus, $x_1 = f_1(m_i)$, where m_i ($1 \leq i \leq N$) is the mean bit rate of connections of type i, N is the total number of connections and f_1 is defined as:

$$f_1 : [0,1]^N \rightarrow [0, u_{\max}]$$

$$f_1(m_1,\ldots,m_N) = u_{\max} \cdot \frac{\sum_{i=1}^{N} m_i}{C} \tag{4.2}$$

- The crisp input, x_2, that matches variable X_2, the *Mean to Peak Ratio*, is obtained by summing the "weighted" mean to peak ratios of all connections and normalising that value to the sum of the mean rates of all connections. Because the connections may have the same mean to peak ratio but different mean rates, the weight is given by the mean rate. Thus, $x_2 = f_2(m_i, p_i)$, where m_i is the mean bit rate and p_i the peak bit rate of connections of type i ($1 \leq i \leq N$), N is the total number of connections and f_2 is defined as:

$$f_2 : [0,1]^N \times [0,1]^N \to [0, u_{\max}]$$

$$f_2(m_1,\ldots,m_N, p_1,\ldots,p_N) = u_{\max} \frac{\sum_{i=1}^{N} (\frac{m_i}{p_i} \cdot m_i)}{\sum_{i=1}^{N} m_i}, \tag{4.3}$$

- The crisp input, x_3, that matches variable X_3, the *Peak rate to Link Capacity ratio* is obtained by calculating the "weighted" sum of the peak to link capacity ratio of each connection. The weights w_i are the square root of the peak to link capacity ratios. The reason for choosing this weight function is that it enables to exaggerate the influence of the actual peak to link ratio as soon as it starts to increase, and therefore, more easily can a high peak to link ratio be detected from the obtained value.

  Thus, $x_3 = f_3(p_i)$, where p_i the peak bit rate of connections of type i, ($1 \leq i \leq N$), N is the total number of connections and f_3 is defined as:

$$f_3 : [0,1]^N \to [0, u_{\max}]$$

$$f_3(p_1,\ldots,p_N) = u_{\max} \cdot \left(\frac{\sum_{i=1}^{N} \left(w_i \cdot \frac{p_i}{C} \right)}{\sum_{i=1}^{N} w_i} \right)^{-P}, \tag{4.3}$$

with $w_i = \sqrt{\frac{p_i}{C}}$, $P \in [0,1]$ is a scaling parameter representing the maximum allowed value for $\frac{p_i}{C}$, $1 \le i \le N$ (suggestion: $P = 0.5$) and C is the link capacity.

- The crisp input, x_4, that matches variable X_4, the *Mean Burst Length to Buffer Size ratio* linguistically measures the requirements of the connections in terms of buffer space (space resource). The crisp input to match this variable, x_4, is obtained by calculating the ratio between the mean burst length, b_i, and the buffer size, K, for each non-CBR connection i, and normalising the obtained value by a threshold value, a, $a \in N$. The parameter a is an integer multiple of the buffer size and enables to scale the mean burst length, b_i, when its value is over the size of the buffer (suggestion: $a = 100$).

Because the value of the input x_4 is required to be normalised (transformed in a value between 0 and 1) and considering that $b_i \in [0, aK]$ the following reasoning is applied: adding and dividing by K, comes $\frac{b_i + K}{K} \in [1, (a+1)]$, then applying logarithms, a normalised value is obtained as $\frac{\log_{10}\left(\frac{b_i + K}{K}\right)}{\log_{10}(a+1)} \in [0,1]$. Summing the previously obtained ratio for N non-CBR connections, yields the expression for f_4:

$$f_4 : [0,1]^N \rightarrow [0, u_{\max}]$$

$$f_4(b_1, \ldots, b_N) = u_{\max} \cdot \frac{\sum_{i=1}^{N} \log_{10}\left[\frac{b_i + k}{k}\right]}{N \cdot \log_{10}(a+1)} \qquad (4.5)$$

If there are CBR connections among the multiplexed connections, the mean burst length of CBR traffic can be considered to assume its maximum value, that is, $b_{CBR} = ak$ and in this case N will refer to the total number of connections, including also non-CBR.

- The crisp output value, y, obtained after defuzzification of the value of output variable Y, the *Maximum Cell Loss Ratio*, represents the negative power of ten for the maximum cell loss ratio per connection obtained for a traffic scenario consisting of the existing connections and the new connection. The universe of Y is the set of integers between 1 and $v_{\max}$.

Previously, when we refer to N, the total number of connections, we refer to the connections already accepted in the ATM link and also the connection upon which the CAC decision is going to be made.

4.4 Definition of the membership functions

The values of fuzzy variables X_1, to X_4 are linguistic terms (fuzzy sets) which are defined by membership functions. In order to achieve flexibility in the definition of the membership function shape, *symmetrical exponential* membership functions (see figure 4.2) have been chosen for the linguistic terms of the antecedent of the rules. These functions enable to obtain membership functions' shapes that can be close to triangular or trapezoidal by varying the values of definition parameter β in expression (4.6) below. Moreover, the amount of overlapping between adjacent fuzzy sets can be adjusted by varying parameter σ in expression (4.6). Symmetrical exponential functions are defined by ([Ng, 1994]):

$$f(x) = \exp\left[-\frac{|x-\alpha|^{\beta}}{\sigma}\right] \tag{4.6}$$

where:

- x is a value of the universe of fuzzy set f, U_f;
- α ($\in U_f$) is the *position* parameter which describes the centre of symmetry of the fuzzy set f;

- β ($\in$ [1.5, 5.0]), is the *shape* parameter which enables to obtain fuzzy sets' shapes such as triangles and trapeziums;
- σ ($\in$ [0.1, 3.0]), is the *scale* parameter which modifies the base-length of the membership function and determines the amount of overlapping.

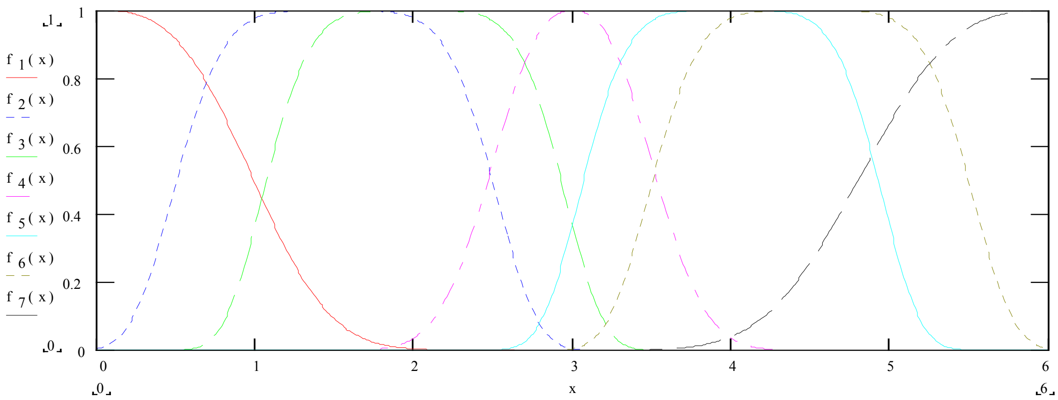

Figure 4.2 Bell-shaped membership functions.

$$f_i(x) = \exp\left[-\frac{|x-\alpha_i|^{\beta_i}}{\sigma_i}\right], \text{ i = 1, ..., 7}$$

$\alpha_1=0$, $\alpha_2=1.5$, $\alpha_3=2$, $\alpha_4=3$, $\alpha_5=4$, $\alpha_6=4.5$, $\alpha_7=6$;
$\beta_1=3$, $\beta_2=5$, $\beta_3=5$, $\beta_4=2.5$, $\beta_5=5$, $\beta_6=5$, $\beta_7=3$;
$\sigma_1=\sqrt{2}$, $\sigma_2=\sqrt{2}$, $\sigma_3=1$, $\sigma_4=0.3$, $\sigma_5=1$, $\sigma_6=\sqrt{2}$, $\sigma_7=2.5$;

The output linguistic variable, Y, *cell loss ratio* is described by linguistic terms represented by singletons, $\{s\}$, where s is the negative power of ten of the cell loss ratio. The membership function of $\{s\}$ is the usual for a conventional set:

$$f_{\{s\}}(x) = \begin{cases} 1, & x = s \\ 0, & x \neq s \end{cases}. \qquad (4.7)$$

Singletons were chosen for the consequent fuzzy variables, aiming to simplify the learning process. This way the number of examples required for training and testing is much less than the required to obtain a floating point value for the cell loss ratio. Furthermore, for CAC purposes, the predicted cell loss ratio does not need to have a floating point precision, because the predicted value helps in the CAC decision making and is not to be used for any further calculations.

4.5 FCAC Inference algorithm

In the following, FCAC's inference algorithm is presented for a rule-base with *n* fuzzy rules and for fuzzy rules with fuzzy sets in the antecedent of the rule and singletons in the consequent. The rule base has the form:

$$R_1 \equiv if\ \ X\ is\ A_1\ then\ Y\ is\ B_1 \quad [\alpha_1, \beta_1]$$
$$R_2 \equiv if\ \ X\ is\ A_2\ then\ Y\ is\ B_2 \quad [\alpha_2, \beta_2]$$
$$\ldots$$
$$R_n \equiv if\ \ X\ is\ A_n\ then\ Y\ is\ B_n \quad [\alpha_n, \beta_n]$$

where:

- $X = (X_1, X_2, \ldots, X_4)$, where X_i is defined in the continuous universe [0, u_{max}], $1 \le i \le n$,
- $A_i = (A_{i1}, A_{i2}, \ldots, A_{i4})$ are fuzzy sets of the domain of X_i,
- Y is defined in the discrete universe $\{1, \ldots, v_{max}\}$, where v_{max} is an integer,
- B_i is a fuzzy set of the domain of *Y*.

The inference algorithm shown in table 4.1 is inspired on the inference model presented in chapter 3, section 3.3, and based on the research by Campos and Gonzalez ([Camp, 1993]). The inference method described in chapter 3 has been applied to a single rule, the inference algorithm further presented ([Ram, 1996]) enables to infer from a rule base with *n* rules and has been adapted to the fuzzy rule format adopted in this study .

Table 4.1 FCAC Inference algorithm ([Ram, 1996])

1. *Initialisation*

$\lambda_{\min}[r] = 1, \quad \lambda_{\max}[i,r] = 0, \quad B^*[r] = 1,$ (4.8)

where i is the index for the rule, $1 \le i \le n$, and r is the integer negative power of ten representing the value that for the consequent variable, Y, can assume, $r \in \{x \in \mathrm{N}: 1 \le x \le m\}$.

2. *Normalisation and scaling*

Calculate the crisp values x_i for the crisp input vector $A^* = (x_1, x_2, x_3, x_4)$ (for more details on how to calculate the crisp values see section 4.3).

3. *Propagation through the fuzzy rule base* (*parallel inference*)

For each of the n rules do (i is the index of the rule):

3.1. Calculate $\lambda_{\max}[i, r] = \max\limits_{j=1..4}\left(1 - A_{ij}(x_j)\right)$ (see footnote [1]) (4.9)

where A_{ij} are the fuzzy sets that variables X_j (j=1…4) in the antecedent of rule i assume as its value and r is the integer negative power of ten that variable Y in the consequent of rule i assumes as its value.

3.2. If ($\lambda_{\max}[i, r] < \lambda_{\min}[r]$) then calculate the output fuzzy sets $B^1[r]$ and $B^2[r]$ given by (3.50) and (3.53), respectively (see chapter 3, section 3.3.4)

3.2.1. $\lambda_{\min}[r] = \lambda_{\max}[i,r]$; (4.10)

3.2.2. $B^1[r] = \lambda_{\max}[i,r] \oplus (1 - \alpha[i])$; (4.11)

3.2.3. $B^2[r] = \lambda_{\max}[i,r] \oplus \beta[i]$, (4.12)

where $\alpha[i]$ and $\beta[i]$ are respectively, the lower and upper bounds of the uncertainty interval associated with rule i.

3.3. For each of the possible r values, $r \in \{x \in \mathrm{N}: 1 \le x \le m\}$, that variable Y can assume do:

3.3.1. If ($\lambda_{\min}[r] > 0.5$) then $B^*[r] = 0$;

3.3.2. Else, for each of the possible s values, $s \in \{x \in \mathrm{N}: 1 \le x \le m\}$, that variable Y can assume do (see footnote [2]):

$$B^*[r] = \begin{cases} \min\left(B^*[r],\, B^2[s]\right) & s = r \\ \min\left(B^*[r],\, B^1[s]\right) & s \ne r \end{cases} \quad (4.13)$$

4. *Defuzzification*

---

[1] Conjunction in the antecedents: $\lambda_{max}[i, r]$ is given by (see chapter 3, section 3.3.2, expressions (3.34) and (3.58))

$$\lambda_{max}[i, r] = c(\neg A, A^*) = \max_{j=1..4}\left(c(\neg A_{ij}, x_j)\right) = \max_{j=1..4}\left(\sup_{x \in U_{X_j}}\left\{\max\left(x_j(x) + \neg A_{ij}(x) - 1, 0\right)\right\}\right)$$

$$= \max_{j=1..4}\left(\sup_{x \in U_{X_j}}\left\{\max\left(x_j(x) + 1 - A_{ij}(x) - 1, 0\right)\right\}\right) = \max_{j=1..4}\left(\sup_{x \in U_{X_j}}\left\{\max\left(x_j(x) - A_{ij}(x), 0\right)\right\}\right)$$

$$= \max_{j=1..4}\left(1 - A_{ij}(x_j)\right), \text{ because } x_j(x) = \begin{cases} 1, & x = x_j \\ 0, & x \ne x_j \end{cases}.$$

[2] Disjunction in the antecedents: a set of n rules of the form "If X is $A(i)$ then Y is B", i=1, ..., n, can be seen as a rule "If X is $A(1)$ and ... and X is $A(i)$ ... and X is $A(n)$ then Y is B", where $A(i)$ is the fuzzy vector in the antecedent of rule i. Applying expression (3.62) comes (4.13).

4.1. Calculate $M = \max_{1 \le r \le m} \left(B^*[r] \right)$ (4.14)

4.2. Calculate the output crisp value, y^*, using the *Mean of Maximum* defuzzification method, that is, by averaging the support values, r, of the inferred fuzzy set B^*, whose membership function value, $B^*[r]$, reaches the maximum value, M:

$$y^* = \frac{\sum_{1 \le r \le m : B^*[r] = M} r}{n_r} \tag{4.15}$$

where n_r is the number of quantified levels, r, of the support of B^* which reach the maximum membership value, M.

4.6 Conclusions

The set of traffic factors chosen to characterise the traffic behaviour of the connections multiplexed on an ATM link enabled to identify the fuzzy variables in the antecedent of the rules.

The format of the rules chosen, more precisely the use singletons in the consequent of the fuzzy rules, enables to have a smaller precision for the set of possible values for the cell loss ratio (integer negative power of ten) which is compensated by the addition of one more traffic factor, the mean burst length to buffer size ratio, in the identification of the traffic scenario under study. Thus, FCAC considers not only the bandwidth requirements (time resource) of the multiplexed connections but also the queueing requirements in terms of the output buffer (space resource).

Bell shaped fuzzy sets were chosen for the fuzzy domain of the input variables in order to add flexibility to the definition of the correspondent membership function. Bell shaped fuzzy sets can look like triangular or trapezoidal fuzzy sets by an adequate choice of its definition parameters. Moreover, the amount of overlapping between adjacent fuzzy sets can also be controlled.

The inference algorithm is efficient in terms of storage and processor requirements. The algorithm uses four vectors with *m* elements (m is set, for example, to 12 if the smallest cell loss ratio to be predicted is 10^{-12}) and one matrix with $n \times m$ entries, where *n* is the number of rules.

The inference calculations require, for *n* rules:

- to calculate the maximum of 4 possible values (expression (4.9)),
- to calculate expressions (4.10 to (4.12),
- the calculate expression (4, 13), B*, requires a maximum of m^2 iterations, but in practice only 3 *m* or 4 *m* iterations are required, if the inferred cell loss ratio is one of three or 4 possible values, respectively.
- to calculate the maximum of *m* possible values in expression (4.14). This is straightforward because in practice just one or two values are different from zero.

The maximum number of rules permitted is not fixed for FCAC's rule base and the actual number of rules is given as a side result of the automatic design method explained in the following chapter.

5 Using genetic algorithms to design automatically the fuzzy knowledge base

A learning technique can be viewed as a search technique in which the states of the search space consist of all the possible hypotheses (rules) for a particular concept and the aim is to find one or more states satisfying a pre-established acceptance criterion. With this in mind, the learning algorithm would be an algorithm which after generating a rule, tests to see if it satisfies a pre-established acceptance criterion; if not, it continues generating and testing rules and in doing so, searches the space of solutions until the acceptance criteria are fully satisfied.

Recently, the need for fast problem solving algorithms for optimisation problems has brought to light probabilistic search algorithms, including probabilistic hill climbing, simulated annealing (see [Ackl, 1987] for more details), genetic algorithms (GA) (see [Holl, 1975]), evolutionary programming (EP) (see [Fogel, 1966]) and evolution strategies (ES) (proposed by I. Rechenberg, see [Rech, 1973]), and Hans-Paul Schewfel, see [Schw, 1981]). Michalewicz in ([Mich, 1992], page 13) presents the term "probabilistic algorithms", referring that:

> "probabilistic algorithms […] do not guarantee the optimum value, but by randomly choosing sufficiently many 'witnesses' the probability of error may be made as small we like. […]
> In general, any abstract task to be accomplished can be thought of as solving a problem, which, in turn, can be perceived as a search through a space of potential solutions. […] For small spaces, classical exhaustive methods usually suffice; for larger spaces special artificial intelligence techniques must be employed."

A search algorithm requires that two main objectives are balanced: *exploitation* of the best solutions and *exploration* of the search space. Hill climbing is a good example of a strategy that mainly exploits the promising regions of a search space, whereas random search strategies do not select promising regions of a search space before commencing their exploration of it. GA are general purpose (domain independent) search methods which provide a good balance between exploration and exploitation of the search space ([Mich, 1992]).

This chapter describes the application of GA to non-symbolic systems, that is, which are systems that do not represent knowledge explicitly (e.g. neural networks, fuzzy logic based systems). The learning algorithm is based on the algorithm by Herrera et al. in ([Herr, 1995]) and performs the following functions:

- automatic generation of the set of fuzzy rules that describe the relation between the traffic behaviour of a set of multiplexed connections and the maximum expected cell loss ratio per connection. The generation process uses pairs of input-output data, also referred to as training examples, that consist of on-line measurements (using ATM traffic analysers or ATM simulators) on the fuzzy system input and output variables,
- tuning of the fuzzy sets for each of the rules every time a new set of test examples is available in order to update the knowledge of the traffic to take into account possible changes in the observed traffic patterns.

A brief introduction to GA and their associated terminology is presented in section 4.1.

Section 4.2 describes the characteristics and implementation aspects of the GA used in each of the following three learning phases:

1. generation of the fuzzy rules;
2. simplification of the rule set previously obtained;
3. tuning of the fuzzy sets for each rule of the simplified rule set.

The characteristics of the GA used in each phase of the algorithm are described and implementation aspects discussed.

5.1 Introduction to Genetic Algorithms

Genetic Algorithms (GA) are categorised as “weak” Artificial Intelligence (AI) problem solving strategies, because, as opposed to “strong” problem solving strategies (e.g. Evolution Strategies), GA make few assumptions about the problem domain. Hence, they enjoy wide applicability. Weak AI strategies can cause combinatorial explosive solution costs when scaling up to large problems. On the other hand, making strong assumptions about the problem domain in the design of an algorithm, very often requires significant redesign when the same algorithm is applied even to related problems ([Mich, 1992]).

GA are best described as search algorithms rather than optimisation algorithms because the GA search paradigm is not meant to fit a particular problem domain but a large variety of problem domains. GA can find one or more optimal solutions to a problem; these solutions may also be optimum, although this is not guaranteed.

GA are different from normal optimisation and search procedures in several ways.

- GA work with a coding of the parameter set, the *chromosome*, not the parameters themselves;
- GA search from a set of possible solutions, the *population*, not a single solution;
- GA use payoff information, the objective function, not derivatives or other auxiliary knowledge;
- GA use probabilistic transition rules, not deterministic rules.

GA are probabilistic algorithms which start with an initial set of likely problem solutions or a randomly initialised set, if no solutions are known, and then evolve towards optimal solutions (see also [Gold, 1989] for more details). New solutions are generated with the use of genetic operators inspired by the reproductive processes observed in nature.

Table 5.1 Comparison of natural and artificial genetics terminology ([Gold, 1989, pp.22)

Natural Genetics	Artificial Genetics
chromosome	string
genotype	structure
phenotype	alternative solution
gene	feature
allele	feature value
locus	string position

GA terminology used in the literature is rooted in both natural genetics and computer science and the corresponding terms are used interchangeably. For instance, following Goldberg's "lingo" ([Gold, 1989], pp. 21):

> "Roughly speaking, the *strings* of artificial genetic systems are analogous to *chromosomes* in biological systems. In natural systems, one or more chromosomes combine to form the total genetic prescription for the construction and operation of some organism. In natural systems, the total genetic package is called the *genotype*. In artificial genetic systems the total package of strings is called a *structure* […]. In natural systems, the organism formed by the interaction of the total genetic package with its environment is called the *phenotype*. In artificial genetic systems, the structures decode to form a particular *parameter set*, *solution alternative*, or *point* (in the solution space).
> […] In natural terminology, we say that chromosomes are composed of *genes*, which may take on some number of values called *alleles* [and] the position of a gene [is] its *locus*. In artificial genetic search strings are composed of *features* or *detectors*, which take on different *values*. Features may be located at different *positions* on the string."

Goldberg in ([Gold, 1989]) describes GA as:

> "search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomised information exchange […]. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the fittest of the old ones [cross-over]; an occasional new part is tried for good measure [mutation]."

Let us assume that the space of solutions is the set of integers between 1 and 63 and that we want to search, using a GA, the maximum solution of the function $f(x) = x^2$.

First of all, five components need to be defined for the above mentioned problem:

1. a genetic *representation* of potential problem solutions or *coding* of the solutions (e.g. for the problem above mentioned we could use a binary string of six digits, say 100100, to represent the integer solution 36);

2. a method for creating an *initial population* either using likely problem solutions or initialising the population randomly;

3. a function verifying the *fitness* of a chromosome (for the problem under study $f(x) = x^2$ could be the function used);

4. genetic *operators* changing a gene's contents in a chromosome:
   reproduction: selection of a pair of genes (parents) that will be crossed-over;
   cross-over: joining and mixing characteristics of the two “parent”-chromosomes in two “child”-chromosomes;
   mutation: arbitrary change of one or more randomly chosen genes of a selected chromosome;

   (for example, let us suppose that from a population of n individuals, the individuals with chromosomes 110010 and 000110 have been selected for reproduction using a “roulette wheel” or proportional selection (see [Gold, 1989], page 11 for details on selection algorithms). If the chromosomes are selected for cross-over, with probability p_{cross}, at a random string position, say 3, the two child chromosomes 110110 and 000010 are obtained. If mutation is applied to the chromosome 110110, with probability p_{mutat}, at randomly selected positions 3 and 5, the chromosome obtained is 111111, and the maximum solution to the above problem, 63, has been found);

5. *values* for the parameters used by the algorithm:
   popsize - maximum number of chromosomes in the population;
   maxgen - maximum number of generations;
   p_{reprod} - probability of a specific chromosome to be selected for reproduction;

p_{cross} - probability of a specific chromosome to be selected for cross-over;

p_{mutat} - probability of a specific gene to be mutated;

threshold - maximum (minimum) value assumed by the fitness function (if known).

Table 5.2 describes the sequence of steps of the simple Genetic Algorithm proposed by Goldberg (see also [Gold, 1989], chapter 1 and see chapter 3 for a computer implementation of the algorithm).

Table 5.2. Simple Genetic Algorithm ([Gold, 1989]).

Step 1: Initialisation of parameters: *popsize, maxgen,* p_{reprod}, p_{cross}, p_{mutat};

Step 2: Initialise population.

Step 3: Repeat while number of generations is less than *maxgen* or *threshold* has not been reached:

- 3.1. Repeat while index(population) is less than *popsize*:
  - 3.1.1. Increment index(population) by two.
  - 3.1.2. Select two parent chromosomes for reproduction with probability p_{reprod};
  - 3.1.3. Cross-over (with probability p_{cross}) and mutate (with probability p_{mutat}) the parent chromosomes to obtain the two child chromosomes;
  - 3.1.4. Calculate the "fitness" value associated with each of the child chromosomes;
- 3.2. Replace population of parents with population of children.

The success behind the search paradigm of the GA is that it implicitly seeks for similarities among strings of the population by relating them with the fitness value of each of the strings. To explain this, Holland ([Holl, 1968], [Holl, 1975]) suggested the term "schema" (plural, "schemata") to refer to a similarity template. A schema describes a subset of strings with similarities at certain string positions Let us consider the binary alphabet for ease of explanation and let us add to it the symbol * or "don't care" symbol. With this ternary alphabet {0, 1, *}, schemata can now be created and the meaning of the schema can be thought of as a pattern matching mechanism: a schema matches a particular string if at every location in the schema a 1 (0) matches a 1 (0) or a * matches either 0 or 1. For example, for strings of size 3, the string *10 matches two strings {110, 010}.

Goldberg relates GA performance to the performance of the schemata by introducing the notions of "building blocks" and "implicit parallelism" ([Gold, 1993]):

> "Highly fit, short-defining-length schemata (we call them *building blocks*) are propagated generation to generation by giving exponentially increasing samples to the observed best; all this goes in parallel with no special bookkeeping or special memory other than our population of *n* strings […]. It turns out that the number [of schemata processed successfully in each generation] is something like n^3. This compares favourably with the number of function evaluations (*n*). [To this processing], we give [the name of] *implicit parallelism*."

Goldberg ([Gold, 1993], pp.41) adds that the reason why short-length schemata are preferred instead of large-length schemata is that they are less susceptible of being destroyed after crossover and mutation of the sampled strings. Another reason is, as stated before, related to the concept of a *building block* ([Gold, 1993], pp.41):

> "In a way by working with these particular schemata (the building blocks), we have reduced the complexity of our problem; instead of building high-performance strings by trying every conceivable combination, we construct better and better strings from these partial solutions of past samplings […] just as a child creates magnificent fortresses through the arrangement of simple blocks of wood, so does a genetic algorithm seek near optimal performance through the juxtaposition of short, low-order, high-performance schemata, or building blocks".

GA have proved to work well as function optimisers for a number of smooth, unimodal, noisy multimodal problems and combinatorial optimisation problems ([Gref, 1985], [Gref, 1987]). Nevertheless, and because limited empirical evidence is not enough to prove a theoretical statement about the convergence of GA for all types of optimisation functions, Bethke ([Beth, 1981]), has generated a number of optimisation test cases that are misleading for the simple GA with operators: reproduction, crossover and mutation. These are, for that reason, called *GA-deceptive* functions. GA-deceptive functions are functions that contain isolated optima, that is, the solutions with a higher fitness value tend to be surrounded by solutions with lower fitness value. Goldberg states ([Gold, 1989], pp. 45) that the application of genetic algorithms as function optimisers for GA-deceptive functions requires long waiting times to arrive at near optimal solutions, and this has to do with the fact that for these functions the building blocks are misleading due to the coding or the fitness function being used.

Moreover, the binary coding used in a simple GA, is not adequate for numerical problems that require a high floating point precision in the values of their solutions. Regarding this, Michalewicz states that ([Mich, 1996]):

> "The binary representation traditionally used in genetic algorithms has some drawbacks when applied to multidimensional, high precision numerical problems. For example, for 100 variables with domains in the range [-500, 500] where a precision of six digits after the decimal point is required, the length of the binary solution vector is 3000. This, in turn, generates a search space of about
> 10 1000 [possible solutions]. For such problems genetic algorithms perform poorly."

The usual justification for using a binary encoding has been that this representation gives the maximum number of schemata per bit of information of any coding ([Gold, 1989]). However, several authors have tried to extend schemata theory to be able to use not only binary but also real coded chromosomes: Antonisse in [Anto, 1989] argued that the "implicit parallelism" characteristic of binary GA does not depend on the use of binary coding. Eshelman and Schaffer in [Esh, 1993] introduced the concept of *interval-schemata* as a tool for real-coded GA analysis. Furthermore, Goldberg in [Gold, 1991] presents a convergence theory for GA with parameters encoded as real numbers.

Given this, De Jong's queries in ([DeJo, 1985]) :

> "What should one do when elements in the space to be searched are most naturally represented by more complex data structures such as arrays, trees, digraphs, etc? Should one attempt to 'linearize' them into a string representation or are there ways to creatively redefine crossover and mutation to work directly on such structures?"

find an answer by Michalewicz in [Mich, 1992], when he states:

> "In particular, for parameter optimisation problems with variables over continuous domains, we may experiment with real-coded genes together with special "genetic" operators developed for them [in order to] move the genetic algorithm closer to the problem space."

Real encoded GA maintain the principles of evolution and inheritance, but use (1) a set of data structures which are richer than the simple binary encoding for chromosome representation and (2) a set of genetic operators which are related to the data representation used (for more details on real encoded GA see also [Mich, 1992], [Mich, 1992a], [Jani, 1990], [Mich, 1991], [Davis, 1989], among others).

In the following, the genetic operators proposed in the literature for real-coded GA (see also [Mich, 1992], [Herr, 1995]) are presented and the specific characteristics of each operator explained.

5.1.1 Reproduction Mechanisms

The reproduction mechanism plays an important role in driving the genetic search towards better solutions than the solutions considered in past generations and also in maintaining a high diversity in terms of the genotype of the individuals in the population ([Bäck, 1991]. The reproduction algorithm defines the selection (reproduction) probabilities for each individual within a population, that is, it defines the potential that each individual has for producing offspring in the new population.

The simple GA proposed by Goldberg (see [Gold, 1989]) and described in table 5.2, uses proportional selection (see definition 5.1 below) to select the individuals for reproduction. This algorithm suffers from premature convergence, that is, the entire population tends to converge to a particular genotype, representing a local maximum (minimum) and the diversity of the search towards a global maximum (minimum) is, therefore, annulled. Baker in [Bak, 1985] gives reasons that help to explain these phenomena:

> "A common cause of rapid convergence is the existence of super individuals which will be rewarded with a large number of offspring in the next generation. Since the population size is typically kept constant, the number of offspring allocated to a super individual will prevent some other individuals from contributing any offspring to the new generation."

and also ([Bak, 1985]):

> "Rapid convergence is not caused solely by an individual receiving too many offspring, but also by the often related situation of many individuals being denied any offspring."

To control or prevent rapid convergence, one either develops a fixed selection algorithm which avoids rapid convergence; or a hybrid system, which adapts the selection algorithm to handle rapid convergence when it occurs ([Bak, 1985]). The first approach is chosen in this thesis because using Baker's *ranking system* as the selection mechanism (see also [Bäck, 1991] for other selection mechanisms) together with the *stochastic universal sampling* algorithm ([Bak, 1987]) premature convergence can be controlled, by fixing the expected number of offspring (the expected value) for each individual in the population. This way, individuals with a high fitness value are not allowed to receive an unlimited number of offspring, blocking other less fit individuals to produce any offspring (premature convergence). The expected value is based on the "rank" of the individual's performance and not in its magnitude. The algorithm used to convert floating point expected values to an integer number of offspring for each individual is called the *sampling* algorithm.

Consider the GA population at generation $t \in \mathbb{N}$, $P^t = (a_1^t,\ldots,a_\lambda^t) \in I$, where λ>1 is the population size and I is the space of individuals a_i^t, i=1, …, λ.The fitness function $f: I \to \mathbb{R}$ provides the environmental feedback to calculate the selection probabilities, $p_s(a_i^t)$, that each individual a_i^t has to be selected for reproduction (the expected value) and $\sum_{i=1}^{\lambda} p_s(a_i^t) = 1$.

Definition 5.1 (*Proportional selection*, [Hol, 1975]):

$$p_s(a_i^t) = \frac{f(a_i^t)}{\sum_{j=1}^{\lambda} f(a_j^t)} \tag{4.1}$$

For reproduction mechanisms based on "ranking", a mapping $rank: I \to \{1,\ldots,\lambda\}$ is defined satisfying:

$$\forall i \in \{1,\ldots,\lambda\}: rank(a_i^t) = i \quad \Leftrightarrow \quad \forall j \in \{1,\ldots,\lambda-1\}: f(a_j^t) \lozenge f(a_{j+1}^t)$$

where ◊ denotes the ≤ relation in case of a minimisation problem and ≥ in case of maximisation. In the following, it is assumed that individuals are always sorted according to their fitness values , with a_1^t representing the best individual of P^t and i denoting the *rank* of individual a_i^t.

Definition 5.2 (Linear ranking selection [Bak, 1985])

$$p_s(a_i^t) = \frac{1}{\lambda}\left(\eta_{\max} - (\eta_{\max} - \eta_{\min})\frac{i-1}{\lambda - 1}\right), \quad (4.2)$$

where $\eta_{\min} = 2 - \eta_{\max}$ and $1 \le \eta_{\max} \le 2$.

The quantities $\eta_{\max}$ and $\eta_{\min}$ are the upper and lower bounds for the expected values of each individual, respectively. Usually, the value chosen for η_{max}, is 1.1, because according to studies done by Baker (see also [Bak, 1985]), the value 1.1 forces the percentage of the current population which contributes offspring to the next generation to a desirable value between 94% and 100%.

Definition 5.3 (Stochastic Universal Sampling [Bak, 1987]). The following is a "C" code fragment to select individual i, given the expected value, ExpVal[i]:

```
ptr = Rand();
for (sum= i= 0; i < popsize; i ++)
      for (sum += ExpVal [i]; sum > ptr;  ptr ++)
      Select_individual (i);
```

Thus, an individual i is guaranteed $\lfloor \text{ExpVal}\ [i] \rfloor$ offspring and no more than $\lceil \text{ExpVal}\ [i] \rceil$, where $\lfloor x \rfloor$ and $\lceil x \rceil$ are, respectively, the biggest integer that do not exceeds x and the smallest integer that exceeds x. Baker ([Bak, 1987]) warns that the population should be "shuffled" before crossover in order to eliminate the positional bias of an individual in the population.

A selection scheme for reproduction purposes can be classified as "elitist" or pure selection, according to the number of times parents are allowed to be selected to reproduce, throughout generations. Quoting Bäck et al in [Bäck, 1991]:

> "Normally, parents are allowed to reproduce in one generation only. Then, they die out and are replaced by some offspring. A selection scheme which enforces a life time of just one generation for each individual regardless of its fitness is referred as pure selection. In an elitist selection scheme some or all of the parents are allowed to undergo selection with their offspring [DeJo, 1975]. This might result in 'unlimited' life times of super-fit individuals."

Definition 5.4 (*Elitist Selection*, [Bäck, 1991])

A selection scheme is called *elitist* or *k-elitist* if and only if

$$\exists k \in \{1,\dots,popsize\} \quad \forall t > 0 \quad \forall i \in \{1,\dots,k\}: \quad f(a_i^t) \lozenge f(a_i^{t-1})$$

Definition 5.5 (*Pure Selection* [Bäck, 1991])

A selection scheme is called *pure* if and only if there is no $k \in \{1,\dots,\lambda\}$ which satisfies the *k*-elitist property.

5.1.2. Genetic operators

The operators used in real-coded GA are different from the ones used in binary coded GA, more precisely ([Mich, 1992], pp.125):

- the space of solutions is a *real valued space* R^q, where a solution is coded as a vector with q floating point components;
- genetic operators can be *non-uniform*, if their action depends on the age of the population.

The size of the chromosome is the same as the length of the vector which is the solution to the problem; in this way, each gene represents a variable of the problem. The values of each gene are forced to remain within the boundaries of the interval domain of each variable by requiring that the space of solutions, generated by the genetic operators crossover and mutation, be convex, that is, satisfies the following requirements ([Mich, 1992]):

R1: for any two points s and t in the solution space S, the linear combination $a \cdot s + (1-a) \cdot t$, where $a \in [0,1]$, is a point in S;

R2: for every point $s \in S$ defined in interval $[l, u]$ and any line p such that $s \in p$, p intersects the boundaries of S at precisely two points, say l_p^s and u_p^s.

5.1.2.1 Mutation operator

The mutation operator in real-coded domains is quite different from the mutation operator in binary coded ones, regarding the actual gene mutation: the gene, being a floating point number is mutated in a dynamic range.

Let us assume that $C = (c_1, \ldots, c_q)$ is the chromosome selected to be mutated, with probability p_{mutat}, and c_i ($i \in \{1, \ldots, q\}$) is a gene of chromosome C selected at random among the set of genes of chromosome C. The result of applying the mutation operator to C is a chromosome $C' = (c_1, \ldots, c_i', \ldots, c_q)$ where c_i', defined in domain $[l_i, u_i]$, is given by:

> **Definition 5.6** (*uniform mutation*, [Mich, 1992]): c_i' is a random value from the range $[l_i, u_i]$, calculated using the uniform probability distribution.

> **Definition 5.7** (*boundary mutation*, [Mich, 1992]): c_i' is either l_i or u_i with equal probability.

Definition 5.8 (*non-uniform mutation*, [Mich, 1992]): $c_i^{'}$ is given by expression

$$c_i^{'} = \begin{cases} c_i + \Delta(t, u_i - c_i) & if \;\; a = 0 \\ c_i - \Delta(t, c_i - l_i) & if \;\; a = 1 \end{cases}$$

where a is a random number which value is either 0 or 1, and $\Delta(t, y)$ is given by:

$$\Delta(t, y) = y \cdot r . \left(1 - \frac{t}{T}\right)^b$$

where r denotes a random number in the interval [0, 1], T is the maximum number of generations and b is a parameter, defined by the user, which determines the degree of dependency with the number of iterations. The function $\Delta(t, y)$ returns a value in the range $[0, y]$ such that the probability of $\Delta(t, y)$ being close to 0 increases as the number of generations increases. This enables non-uniform mutation to search the space of solutions uniformly when the number of generations is small and very locally at later stages ([Mich, 1992]).

5.1.2.2 Cross-over operator

Let us assume that chromosomes $C^1 = (c_1^1, \ldots, c_i^1, \ldots, c_q^1)$ and $C^2 = (c_1^2, \ldots, c_i^2, \ldots, c_q^2)$ are selected, with probability p_{cross}, to be crossed-over. Then the result of the application of the cross-over operator is (in the following only cross-over operators that satisfy the convexity requirements R1 and R2, in section 5.12, are presented):

Definition 5.9 (*Arithmetical cross-over,* [Mich, 1992], [Herr, 1994]): the two chromosomes:

$$D^1 = (d_1^1, \ldots, d_i^1, \ldots, d_q^1) \text{ and } D^2 = (d_1^2, \ldots, d_i^2, \ldots, d_q^2)$$

where $d_i^1 = a \cdot c_i^1 + (1-a) \cdot c_i^2$ and $d_i^2 = a \cdot c_i^2 + (1-a) \cdot c_i^1$ and $a \in [0, 1]$ is a constant (uniform arithmetical cross-over) or varies with the maximum number of generations (non-uniform arithmetical cross-over).

Definition 5.10 (*Max-Min arithmetical cross-over,* [Herr, 1995]): the two best chromosomes among the following four chromosomes:

$$D^1 = (d_1^1, \ldots, d_i^1, \ldots, d_q^1),$$

$$D^2 = (d_1^2, \ldots, d_i^2, \ldots, d_q^2),$$

$$D^3 = (d_1^3, \ldots, d_i^3, \ldots, d_q^3),$$

$$D^4 = (d_1^4, \ldots, d_i^4, \ldots, d_q^4),$$

where $d_i^1 = a \cdot c_i^1 + (1-a) \cdot c_i^2$, $d_i^2 = a \cdot c_i^2 + (1-a) \cdot c_i^1$, $d_i^3 = \max(c_i^1, c_i^2)$, $d_i^4 = \min(c_i^1, c_i^2)$ and $a \in [0, 1]$ is a constant (uniform max-min arithmetical cross-over) or varies with the maximum number of generations considered (non-uniform max-min arithmetical cross-over).

5.2 Learning algorithm applied to the design process

The application of GA to the automatic design of fuzzy logic systems has been purposed often by researchers trying to introduce self-adaptation in fuzzy based tools applied to control and decision making ([Surm, 1993]). Furthermore, the search capabilities of GA have been used in automatic fuzzy systems design, more precisely, to tune the fuzzy sets membership functions ([Surm, 1993], [Ng, 1994], [Karr, 1992]), to determine the number of fuzzy rules and the fuzzy set for the consequent of the rules ([LeeM, 1993], [Thri, 1991]).

This section will present an application of GA to the automatic design of FCAC. The algorithm enables to create the fuzzy rule base and to define the membership functions of the fuzzy sets in the antecedent and consequent of the rules. The number of fuzzy variables in the antecedent and consequent of the rules is fixed and so is the format of the fuzzy rules.

The learning algorithm described in the following, is based on the algorithm proposed by Herrera et al. ([Herr, 1995]) to automatically design a fuzzy logic system using GA. The algorithm uses the:

1. method of learning from examples presented in chapter 3, section 3.2.3,
2. the inference algorithm proposed in chapter 4, section 4.5.

The learning algorithm used to automatically design FCAC is composed of three main phases (see also figure 5.1):

1. *Generation* phase. In this phase the initial set of fuzzy rules, R_0, is generated, more precisely, the fuzzy sets for each of the fuzzy variables in the antecedent and consequent of the rule) are defined (fuzzy domain F_0), using the knowledge implicitly expressed in a set of training examples;

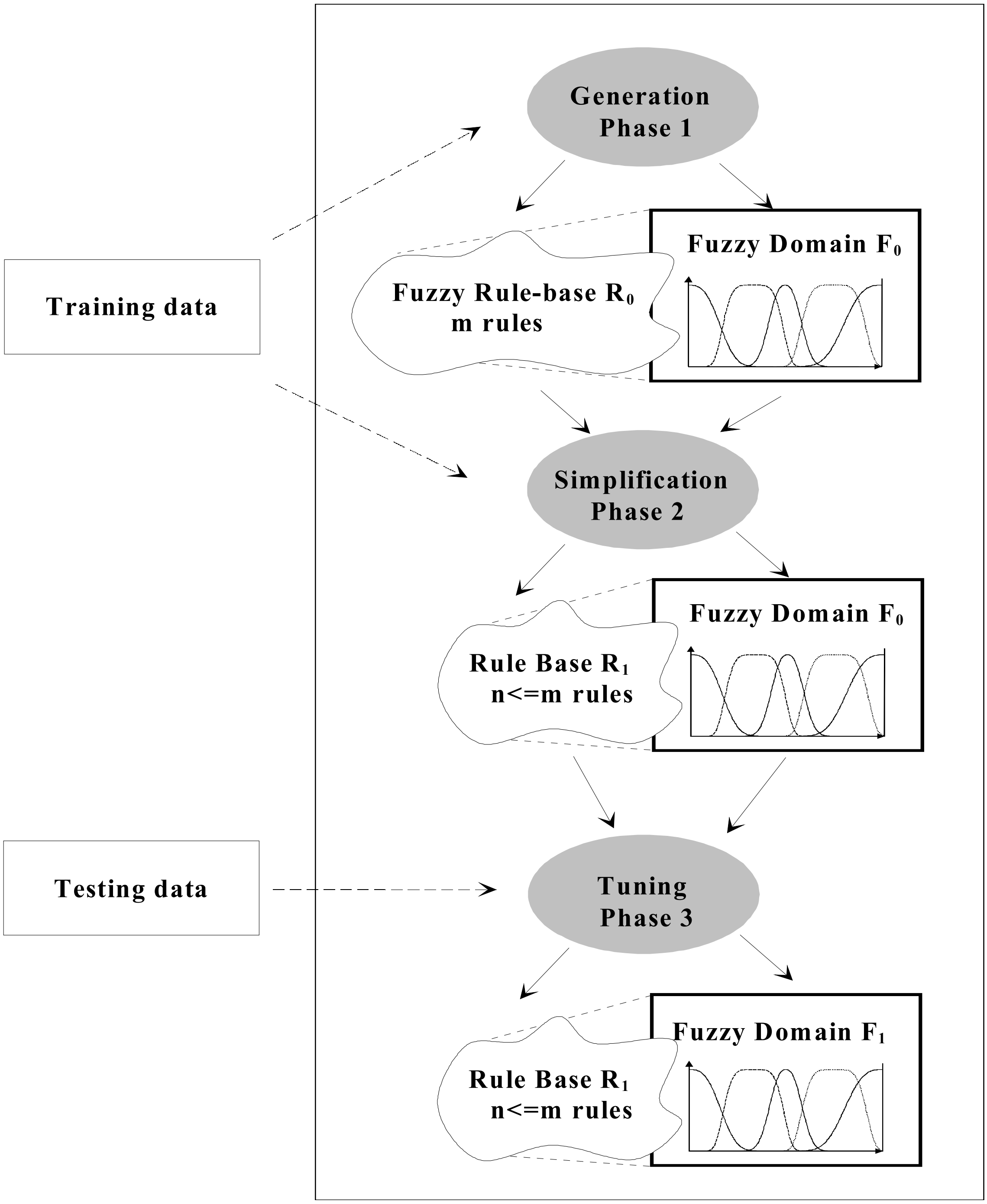

Figure 5.1 The algorithm for automatically design FCAC using training and test data sets.

2. *Simplification* phase. In this phase, the rule set R_0, obtained formerly, is simplified, that is, a rule set R_1, with a number of rules less than or equal than the number of rules in R_0 is obtained. The fuzzy domain F_0 remains unchanged. The simplification

algorithm aims to eliminate redundant rules (e.g. duplicated rules), that is, rules which elimination does not alter the performance of the system. The performance of FCAC is evaluated for a set of training examples (see section 5.2.3 for more details);

3. *Tuning* phase. In this phase, the fuzzy sets for each of the rules in rule set R_1 are tuned using a set of test examples, thus obtaining a new fuzzy domain F_1. The tuning algorithm aims to increase the performance of the FCAC by comparing its inference response for each of the test examples (see section 5.2.4 for more details).

In order to enable FCAC to adapt to changes in the traffic patterns an *adaptation mode* algorithm, similar to the one outlined before, is used every time a new set of test examples is available.

Table 5.3 Adaptation algorithm

1. The *performance* of FCAC is tested by measuring the mean square error between the cell loss prediction stated in the example and the cell loss prediction given by FCAC, for each of the test examples. If for any of the test examples, FCAC fails to give a prediction then the square of the cell loss value stated in the example is added to the current error sum. The expression to calculate FCAC performance is stated in section 5.2.4.1.3, expression (T2);

2. The performance of FCAC is compared with a *threshold* given by the sum of the squared cell loss values for all the test examples;

2.1 If the performance of FCAC is less than the threshold then the tuning phase is invoked to tune the fuzzy sets of each of the fuzzy rules.

2.2 If the performance of FCAC is equal to the threshold, then the three-phase algorithm described in figure 5.1 is invoked using as training data the new set of test examples. A new set of rules, R_{TEST}, and the respective fuzzy domain, F_{TEST}, are generated for the test data. The rule set R_{TEST} is added to the current set of rules, R_{CUR}, and the simplification phase is invoked to eliminate redundant rules. A new rule set, R_{NEW}, is obtained from the simplification process and the correspondent fuzzy domain, F, is then tuned to obtain the final fuzzy domain F_{NEW}.

Case 2.1 occurs when the characteristics of the traffic represented in the examples is not very different from the characteristics of the traffic studied up to then. To tune the fuzzy sets is enough to provide for the adaptation of FCAC to small variations in the traffic characteristics that do not require changes in the rules associations and can be captured by adjusting the parameters that describe the fuzzy sets for each of the fuzzy rules. Case 2.2 occurs when the characteristics of the multiplexed traffic have changed significantly from the traffic previously studied and FCAC lacks the knowledge to predict a cell loss value.

The sampling mechanism used in the selection of the examples to be included in the set of test examples plays a very important role in the success of the adaptation algorithm presented above. This is because if the set of test examples contains examples representing traffic mixes very distinct from the ones studied up to then, it is very likely that the simplification algorithm will eliminate from the set of rules R_{CUR}, the rules that describe the traffic scenarios observed in the past. Considering that the elimination of rules expressing past knowledge which is not yet outdated is not a desirable feature of the algorithm, some care must be put in the selection of the examples included in the set of test examples. The sampling mechanism should monitor the traffic patterns in the link for a wide period of time and statistically treat the traffic data in order to include in the set of test examples a higher percentage of test examples for those traffic patterns which are more popular (and, thus more likely to appear in the future) and a lower percentage of test examples corresponding to traffic patterns that occur sporadically. The use of a carefully selected set of test examples also envisages to avoid the case 2.2 of the adaptation algorithm above which is time consuming and, thus, to increase the speed of the adaptation process in a real implementation.

Before explaining in detail each of the phases of the learning method, the definitions and general assumptions made are reviewed briefly.

5.2.1 Definitions and general assumptions

5.2.1.1 Training and test examples

The examples used for training and testing FCAC are obtained using the same procedure.

Assuming that for a specific trafic pattern consisting of N multiplexed traffic connections defined by the peak bit rate, p_i, mean bit rate, m_i and mean burst length, b_i, (i=1,...,N) and a traffic scenario consisting of a link with capacity C Mbit/s and an output buffer size of K cells, the maximum cell loss ratio obtained per connection is y. The example that describes this traffic pattern is an example $e = (ex, ey)$, where $ex=\{ex_1,ex_2,ex_3,ex_4\}$ is the vector of example values that corresponds to the 4 antecedent variables, X_1, X_2, X_3 and X_4 and ey is the example for the consequent variable, Y (see chapter 4, section 4.3 for the definition of these variables) is constructed as follows:

- the values of each of the elements of vector ex are obtained using formulas (4.2) to (4.5) given in chapter 4, section 4.3. Thus, $ex_1 = f_1(m_i)$, where $f_1(m_i)$ is given by expression (4.2), $ex_2 = f_2(m_i, p_i)$, where $f_2(m_i, p_i)$ is given by expression (4.3), $ex_3 = f_3(p_i)$, where $f_3(p_i)$is given by expression (4.4) and finally $ex_4 = f_4(b_i)$, where $f_4(b_i)$is given by expression (4.5);
- the value for ey is obtained by rounding the floating point number y (the measured cell loss ratio) to the nearest negative integer power of ten and assigning ey to this integer value.

5.2.1.2 Soft completeness and consistency conditions

The acceptance criterion used by search algorithms applied to concept learning should ensure that the concept description is simple, complete and consistent. The concept description considered when studying the application of fuzzy logic to CAC corresponds to the set of fuzzy rules describing the relation between traffic characteristics and the measured cell loss ratio.

A concept description is said to be *simple* in the sense that the acceptance criteria prefers descriptions which are shorter, where "shorter" could mean that the rules have the smallest number of conjunctions in the rule's antecedents or that the rule base should contain the smallest number of rules. A concept description is *complete* over a set of positive examples if and only if it covers all positive examples in the set. A concept description is *consistent* over a set of negative examples if and only if it does not cover any negative example in the set.

Often a complete and consistent concept description can not be achieved. The reason for this is that sometimes the language of concepts is too restrictive or the set of positive and negative examples overlap. The latter be related to the discrepancies between observations and examples or between concepts and concept descriptions ([Raedt, 1992]). This is particularly important when considering concepts in fuzzy domains where the concept "descriptions" are expressed in terms of fuzzy variables and fuzzy rules. Because the rules are fuzzy, the examples have a partial matching with the rules. Hence, the consistency and completeness conditions become a matter of degree. Taking this into account, "soft" definitions of consistency and completeness have been proposed by Gonzalez and Perez ([Gon, 1995]) will be presented in the following paragraphs.

Let us suppose we know a set of *h* examples $E_h = \{e_1, e_2, ..., e_h\}$ of the system under study, consisting of the values that the variables take during an experiment of the type: "in time $t = k$, the value of the input variables X_r (r=1, ..., M) and output variable Y (antecedent and consequent of the rules, respectively) is ex_{rk} and ey_k, respectively".
The example $e_k = (ex_k, ey_k)$, k=1, …, h with vector $ex_k = \{ex_{1k}, ex_{2k}, ..., ex_{Mk}\}$, and each ex_{rk} belongs to the domains of X_r (r =1,…M) and ey_k to the domain of Y.

Definition 5.11 (*Soft*-consistency): Let $R \in \Delta$ be a rule and $k \in [0,1]$ be a fixed parameter. Rule R, such that $n_E^+(R)>0$, satisfies the ***k**-consistency* condition if and only if

- $k = 0$, then $n_E^-(R)=0$;
- $k \in (0,1]$, then $n_E^-(R) < \boldsymbol{k}\ n_E^+(R)$, (C1)

where $n_E^-(R) = \left|E^-(R)\right|$ and $n_E^+(R) = \left|E^+(R)\right|$, with $|A| = \sum_{u \in U} A(u)$ (see chapter 3, section 3.2.3 for the definition of fuzzy sets $E^-(R)$ and $E^+(R)$).

The *hard*-consistency definition is included in this definition as a 0-consistency condition. Parameter k may be interpreted as a noise threshold, since a rule satisfies the k-consistency condition when a noise level less than 100k % of the positive examples, appears in the example set. Parameter k will be referred to as the *consistency parameter*.

Definition 5.12 (*Soft*-completeness): A rule set Δ *satisfies* the *λ-completeness* condition if and only if it covers all possible situation-action pairs, $e_k \in E_h$. This may be formulated, for some constant $\lambda \in [0, 1]$ as:

$$C_\Delta\left(e_k\right) = \bigcup_{i=1..T} R_i(e_k) \geq \lambda, \quad k = 1,\ldots,h \qquad \text{(C2)}$$

where:

- T is the number of rules and h the number of examples;
- $R_i(e_k)$, $i = 1, ..., T$, is the compatibility degree between the rule R_i and the example e_k, given by $R_i\left(e_k\right) = \Pi_{A_i}(ex_k) * \Pi_{B_i}(ey_k)$, with * the Lukasiewicz t-norm, see also (3.13), chapter 3;

- $A_i = (A_{i1},\ldots,A_{iM})$ and B_i are fuzzy sets in the antecedent and consequent of the fuzzy rules, respectively;
- $\Pi_{A_i}(ex_k)$ is given by expression (3.9), chapter 3;
- $\Pi_{B_i}(ey_k)$ is given by expression (3.6), chapter 3.

Parameter λ will be referred to as the *completeness parameter*.

Definition 5.13 (*Covering value*): Given a set of rules Δ, the *covering value* of example e_k, k=1,...,h, by the set of rules Δ, $CV_\Delta(e_k)$, is a value (between 0 and 1) given by

$$CV_\Delta(e_k) = \sum_{i=1}^{T} R_i(e_k)$$

and the soft-coverage condition (**ε**-*covering*, with $\varepsilon \in [0, 2]$) of a set of examples E_h requires

$$CV_\Delta(e_k) \geq \varepsilon \tag{C3}$$

for all examples e_k,, k=1, ...,h in the set of examples E_h, that is,

$$\bigcap_{e_k \in E_h} CV_\Delta(e_k) \geq \varepsilon \tag{C3'}$$

where the left side of the inequality above represents the coverage degree of the set of examples E_h, by the set of rules Δ, $CV(E_h, \Delta)$. ε is referred in the following as the *covering* parameter.

5.2.2 The generation phase

The *generation* phase consists in iteratively:

1. applying a Genetic Algorithm (GA_{gen}) over the set of training examples to generate a fuzzy rule,
2. assigning to every example the relative covering value,
3. eliminating the examples with a covering value greater than the covering parameter.

The algorithm referred in the previous paragraph has been proposed by Herrera et al in [Herr, 1995]) and is referred as the *covering* algorithm, because it aims to cover all examples of a set of training examples by a set of rules, using a GA to generate each of the rules. Thus, in each iteration of the algorithm, the GA (GA_{gen}) is executed and a new fuzzy rule, R, is generated, more precisely, values are obtained for the parameters that define the fuzzy set membership functions for each of the fuzzy variables of the antecedent and consequent of the rule. The new rule R is added to the rule base RB_{gen} (initially empty) if it satisfies the consistency and completeness conditions (C1) and (C2) previously stated (see definitions 5.11 and 5.12). The example(s) that satisfy coverage condition (C3) (see definition 5.13) for rule base RB_{gen} are eliminated from the set of examples and the iterative process starts again. The algorithm finishes when the example set is empty, that is, when all examples from the initial example set have been "covered". This is the case when all examples have a coverage degree greater or equal than the covering parameter.

Table 5.4 Generation algorithm (based on covering algorithm of Herrera et al. ([Herr, 1995]).

1. Initialisation of completeness parameter λ, consistency parameter k and covering parameter ε; assign coverage value, CV[i], to zero for every example $e_i \in E_h$, $i = 1, ..., h$ and $RB_{gen} = \varnothing$.
2. While E_h is not empty do:
    2.1 Application of the GA_{gen} using a set of examples E_h.
    2.2 Selection of the best chromosome C_r with R_r the associated fuzzy rule.
    2.3 If k-consistency (C1) and λ-completeness (C2) conditions are verified for rule R_r then:
        2.3.1 Calculate the interval of uncertainty $[\alpha_r, \beta_r]$ associated with R_r;

2.3.2 Introduce R_r in the set of rules RB_{gen};

2.4 For every $e \in E^+(R_r)$ do (see (3.18), chapter 3 for the definition of $E^+(R_r)$):

2.4.1 Update CV[i] = CV[i] + $R_r(e_i)$ (see (3.13), chapter 3 on how to calculate $R_r(e_i)$),

2.4.2 If CV[i] $\geq \varepsilon$ then remove example e_i from E_h.

The values suggested for parameters λ, k and ε used in the covering algorithm are λ=0.7, k = 1.5 and ε=0.5.

5.2.2.1 *GA*

In the following, a detailed description of the characteristics of the GA used to generate each of the fuzzy rules, GA_{gen}, is presented. GA_{gen} generates a fuzzy rule by defining the fuzzy sets for each of the fuzzy variables in the antecedent and consequent of the rules.

Each of the chromosomes of GA_{gen} represents a fuzzy rule, and the fitness of each chromosome is evaluated by means of the concepts of frequency, positive and negative examples defined in chapter 3, section 3.2.2. GA_{gen} searches for rules with a high frequency value, a high number of positive examples and a small number of negative examples.

5.2.2.1.1 GA_{gen} genetic representation

In GA_{gen}, a candidate solution (chromosome) C_r, represents a fuzzy rule:

"If X_1 is A_{r1} and ... and X_4 is A_{r4} Then Y is B_r"

where the universe of every input fuzzy set A_{rj} is a closed real interval U= [0, 10]. Similarly, the universe of the output fuzzy set B_r is the discrete set $\{i \in \mathrm{N}: 1 \leq i \leq 10\}$ (see chapter 4, section 4.2 for the definition of X_1, X_2, X_3, X_4, and Y).

The chromosome represents the parameters that define the linguistic terms (fuzzy sets) of the antecedent and consequent of the rules. The antecedent fuzzy sets are defined by the symmetrical exponential membership functions given by (3.59) and therefore completely defined by parameters $\alpha\in[0, 10]$ (position parameter), $\beta\in[1.5, 5.0]$ (shape parameter) and $\sigma\in[0.1, 3.0]$ (scale parameter). The linguistic terms for the consequent of the rules are singletons (crisp sets) given by (3.60) and therefore completely defined by parameter $b\in\{i\in\mathrm{N}: 1\leq i\leq 10\}$.

Thus, C_r codes the real values:

- α_{rj}, β_{rj} and σ_{rj} which are the parameters defining the membership functions of fuzzy sets A_{rj}, j=1, ...,4
- b_r, the parameter defining singleton B_r, r is the index associated with the rule.

$$C_r = (\alpha_{r1}, \beta_{r1}, \sigma_{r1}, \ldots, \alpha_{rn}, \beta_{rn}, \sigma_{rn}, b_r)\ . \qquad \text{(G1)}$$

5.2.2.1.2 Initialisation of GA_{gen} population

The initial gene pool is created partially from a subset with t examples, E_t, of the set of examples E_h ($t \leq h$) and the rest randomly. Setting t = min $\{h, \frac{popsize}{2}\}$, t examples randomly chosen from E_h are used to create t chromosomes in the following way:

$$C_k = (\ ex_1^k, 2, 0.2, \ldots, ex_4^k, 2, 0.2, ey), \qquad \text{(G2)}$$

for an example $e_k = (ex, ey) \in E_t$ where component $ex = (ex_1^k, \ldots, ex_4^k)$, k=1,…, t.
The genes for the remaining ($popsize$ - t) chromosomes of the initial population are chosen randomly within the domain of the respective gene.

5.2.2.1.3 Evaluation of the fitness of each chromosome in GA_{gen}

In order to assess which are the "best" rules for a set of training examples, we need to be able to decide whether an example is positive or negative for a rule and to count the number of positive and negative examples. The fuzzy sets $E(R_i)$, $E^+(R_i)$ and $E^-(R_i)$

defined in chapter 3, section 3.2.3, by expressions (3.17) to (3.19), respectively, will be used for this purpose.

The objective is to *maximise* the fitness function value, $F(R)$, given by

$$F(R) = \Psi_E(R) \cdot \Gamma_{E^+}(R) \cdot g(n_E^-(R)), \qquad \text{(G3)}$$

and defined as the product of:

* $\Psi_E(R)$, the frequency of a rule R given an example set E_h (given by expression (3.12) in chapter 3, section 3.2.3);

* $\Gamma_{E^+}(R)$ is the frequency of rule R in the set of positive examples for the rule, given by expression

$$\Gamma_{E^+}(R) = \frac{\sum_{e_k \in E^+(R)_\lambda} R(e_k)}{n_E^+(R)} \qquad \text{(G4)}$$

  with $E^+(R)_\lambda$ the set of the positive examples of R, for which the compatibility degree between the example and the rule is greater than covering parameter $\lambda \in [0, 1]$, that is, $E^+(R)_\lambda = \{e_k \in E^+(R): \ R(e_k) > \lambda\}$ and $n_E^+(R)$ is the number of positive examples in $E^+(R)_\lambda$.

* $g(n_E^-(R))$ represents a weight measure on the number of negative examples given by

$$g(n_E^-(R)) = \begin{cases} 1 & , n_E^-(R) \le z \\ \left(n_E^-(R) - z + \exp(1)\right)^{-1} & , otherwise \end{cases} \qquad \text{(G5)}$$

  with $E^-(R)$ the set of the negative examples of R defined as $E^-(R) = \{e_k \in E: \ R(e_k) = 0 \ and \ A(ex_k) > 0\}$ and $n_E^-(R)$ is the number of negative examples in $E^-(R)$ and z is the admissible number of negative examples.

5.2.2.1.4 GA_{gen} genetic operators

The genetic operators are the non-uniform mutation operator proposed by Michalewicz (see definition 5.8) that enables to fine tune the capabilities of the GA as the number of generations increases and the max-min-arithmetical crossover proposed by Herrera et al (see definition 5.10) because, as shown in [Herr, 1993]) it enables a real encoded GA to achieve a greater accuracy in the best result for a varied number of fitness functions, when compared with real encoded GA that use other crossover operators.

In addition, elitist selection (see definition 5.4) is used to select the individuals for the next generation. Linear ranking (see definition 5.2) with stochastic universal sampling (see [Bak, 1987]) are used to select the individuals for reproduction. Linear ranking enables to control premature convergence by controlling the range of trials for reproduction allocated to any single individual during a generation.

5.2.2.1.5 GA_{gen} configuration parameters

The values selected for GA_{gen} parameters were obtained according to suggestions by Herrera et al in [Herr, 1995] and also by trial and error running GA_{gen} and observing the variations of performance.

Table 5.5 *GA*$_{gen}$ configuration parameters.

Parameter description	***Parameter value***
maximum number of generations	*maxgen* = 200
population size	*popsize* = 100
probability of crossover	$p_{cross} = 0.6$
probability of mutation	$p_{mutat} = 0.6$
probability of gene mutation	$p_{gene} = \frac{p_{mutat}}{13}$ (13 is the length of the chromosome)
max-min arithmetic crossover parameter	$a = 0.35$
non-uniform mutation parameter	$b = 5$
maximum expected value for ranking	η_{max}, = 1.1
covering parameter	$\lambda = 0.7$
consistency parameter	$k = 0.2$
covering method parameter	$\varepsilon = 1.5$

5.2.2.1.6 *GA*$_{gen}$ package

GA$_{gen}$ uses a GA package designed by Michalewicz, the GEnetic algorithm for Numerical Optimisation for Constrained Problems or GENOCOP, version 2.0[1] (see also [([Mich, 1992b] or [Mich, 1992], chapter 7). GENOCOP addresses a way of handling constraints in numerical optimisation problems by eliminating the equalities present in the set of constraints, and to use genetic operators that satisfy the convexity requirements of the space of solutions.

The software is written in C and runs under either DOS or UNIX operating systems.

---

[1] GENOCOP can be obtained from Department of Computer Science, University of North Carolina, Charlotte, NC 28223, USA, or mailing zbyszek@mosaic.uncc.edu

5.2.3 The simplification phase

The *simplification* algorithm takes as input the generated rule set RB_{gen} and searches for a subset of candidate rules $RB_{\text{simp}} \subseteq RB_{\text{gen}}$ that are able to describe the traffic knowledge implicitly expressed in the set of training examples. The reason for this simplification is that the generation process might produce two (or more) similar rules and, therefore, some of the generated rules can be eliminated.

5.2.3.1 GA_{simp} implementation aspects

In the following, a detailed description of the characteristics of the GA used to simplify the rule set RB_{gen} is presented.

5.2.3.1.1 GA_{simp} genetic representation

A binary GA, GA_{simp}, is used in this phase where the chromosomes are *m*-bit binary strings, and *m* is the number of rules of rule base RB_{gen} obtained in the generation phase. The *m*-bit chromosome $C = (c_1,...,c_m)$ represents either a subset of rules RB_{simp} of the rule set RB_{gen} of candidate rules or the whole rule set RB_{gen}, in the following way:

$$If\ c_i = 1\ then\ R_i \in R_{simp}\ else\ R_i \notin R_{simp},\ i = 1, ..., m \qquad \text{(S1)}$$

In other words, chromosome *C* represents a solution vector $(i_1, i_2, ..., i_m)$, $i_r \in \{0,1\}$, r=1, ...,m, where i_r=1 means that rule *r* belongs to the set of simplified rules, RB_{simp}, and i_r= 0 means that rule *r* does not belong to the set of simplified rules.

In order to codify the *m* rules in chromosome C using (S1), the rule set obtained in the generation phase, RB_{gen}, is scrambled (ordered in an arbitrary way) because statistically the algorithm will proceed more quickly. Each rule is associated with a gene of the *m*-bit chromosome. Hence, the chromosome $C = (c_1,...,c_m)$ represents a subset of n ($n \le m$) of the m candidate rules as described in (S1) if $\sum_{i=1}^{m} c_i = n$.

5.2.3.1.2 Initialisation of GA_{simp} population

In the initial GA_{simp} population, a chromosome with c_i=1, i=1,...,m is introduced representing the complete rule set, RB_{gen}, obtained by the generation algorithm (table 5.3). All the other chromosomes are obtained by performing mutations on the genes of this chromosome with probability p_{gene} (see table 5.5 for the values of GA_{simp} parameters).

5.2.3.1.3 Evaluation of the fitness of each solution in GA_{simp}

The obtained rule set RB_{simp} is the rule set for which the mean square error function over a set of examples E_h is minimum (E_h is the same set of training examples that is used in the generation phase).

Given the set of h examples E_h, where example $e_k = (ex_k, ey_k) \in E_h$, k=1,…, h, is composed of the vector of values for the antecedent variables, ex_k, and consequent fuzzy variable, ey_k, the fitness function of GA_{simp} for a chromosome C is defined as:

$$F(C) = \begin{cases} \frac{1}{2h}\left[\sum_{i=1}^{h}\left(ey_i - FI(ex_i)\right)^2\right] \times \ln\left(\sum_{i=1}^{m} c_i\right), & \textit{if } CD(E_h, \Delta_C) \ge \varepsilon \textit{ and } FI(ex_i) \ne 0 \\ \frac{1}{2h}\sum_{i=1}^{h}(ey_i)^2, & \textit{otherwise} \end{cases} \quad \text{(S2)}$$

where $\sum_{i=1}^{m} c_i$ represents the number of rules represented in C and FI (ex_i) is the output value obtained after inferring from a fuzzy rule base Δ_C for input vector ex_i (see chapter

4, section 4.5 for details on the fuzzy inference algorithm). Δ_C is obtained from chromosome *C*, using (S1).

If nothing can be inferred from rule base Δ_C for input ex_i, that is, $FI(ex_i)=0$, or the coverage degree of an example set E_h by a rule set Δ_C, $CD(E_h,\Delta_C)$, given by $CD(E_h,\Delta_C) = \bigcap_{i=1}^{h} CV_{\Lambda_c}(e_i)$, does not satisfy condition (C3'), then the value $(ey_i)^2$ is added to the current sum instead of the squared difference between ey_i and $FI(ex_k)$.

Summarising, GA_{simp} aims to minimise the mean square error between the output given by the examples and the output obtained from fuzzily infering based on the set of rules represented in the chromosome. Rule bases with a big number of rules are penalised using the logarithmic function on the number of rules, *n*, given by $f(n) = ln(n)$.

5.2.3.1.4 GA_{simp} genetic operators

The classical mutation and crossover operators for binary strings are used (see [Gold, 1989]). Elitist selection (see definition 5.4) is used to select the individuals for the next generation. Linear ranking (see definition 5.2) and stochastic universal sampling (see [Bak, 1987]) are used to select the individuals for reproduction in order to control premature convergence by controlling the range of trials for reproduction allocated to any single individual during a generation.

5.2.3.1.5 GA_{simp} configuration parameters

The values selected for GA_{simp} parameters were obtained according to suggestions published in the literature (see [Gold, 1989])) and also by trial and error running GA_{simp} and observing the variations of performance.

Table 5.6 GA_{simp} configuration parameters.

Parameter description	***Parameter value***
maximum number of function evaluations per GA_{simp} run	$t = 1000$
population size	$popsize = 100$
probability of crossover	$p_{\text{cross}} = 0.6$
probability of mutation	$p_{\text{mutat}} = 0.01$
probability of gene mutation	$p_{\text{gene}} = \frac{p_{mutat}}{m}$ (m is the length of the chromosome for rule base RB_{gen})
maximum expected value for ranking	$\eta_{max}, = 1.1$
coverage parameter	$\varepsilon = 1$

5.2.3.1.6 GA_{simp} package

GA_{simp} uses a GA implementation written by Bäck ([Bäck, 1992]), GENEsYs-1.0 [2], which is an extension of the original GA software package, GENESIS-4.5, written by Grefenstette ([Gref, 1987a]). The extensions, implemented in GENEsYs-1.0, to the GA implementation in GENESIS-4.5, refer to:

- an enlargement of the set of crossover and mutation operators to the ones implemented in GENESIS-4.5; proportional ([Hol, 1975]) or linear ranking selection ([Bak, 1985]) can also be used among other selection mechanisms ([Bäck, 1992]),
- enhancements in the actual software in order to make it more user-friendly, such as, either the command line or a "setup" program can be used to configure the GA and also the provision of a table of objective functions, allowing the user to choose from one of the objective functions when invoking the GA.

The software is written in C and runs under the UNIX operating system.

[2] GENEsYs-1.0 can be obtained from ftp from lumpi.informatik.uni-dortmund.de: /pub/GA/src/GENEsYs-1.0.tar.Z

5.2.4 The tuning phase

The *tuning* algorithm tunes the set of rules, RB_{simp}, obtained by the simplification algorithm by tuning the shapes of the fuzzy sets for the antecedents of the fuzzy rules. The number of the rules and the consequent of the fuzzy rules in RB_{simp} remains the same.

5.2.4.1 GA_{tun} implementation aspects

In the following, a description of the genetic algorithm, GA_{tun}, used in this phase is presented.

5.2.4.1.1 GA_{tun} genetic representation

A chromosome $C = \left(C_1 C_2 ... C_t\right)$, represents a rule base with t rules ($t<m$ is the number of fuzzy rules obtained by the simplification phase) and each C_i, i=1,…,t is given by

$$C_i = \left(\alpha_{i1},\beta_{i1},\sigma_{i1},\alpha_{i2},\beta_{i2},\sigma_{i2},...,\alpha_{i4},\beta_{i4},\sigma_{i4}\right)$$

where α_{ij}, β_{ij}, σ_{ij}, i=1,…,t, j=1,…,4 are the actual values of the parameters that define the fuzzy sets of the antecedent of rule i.

5.2.4.1.2 Initialisation of GA_{tun} population

In the initial GA_{tun} population, one of the chromosomes represents the rule base obtained by the simplification process. This is done by assigning the triplets (α_{ij}, β_{ij}, σ_{ij}), j=1,…,4, in chromosome C with the real encoded values of each of the 4 fuzzy sets in the antecedent of the rules, (αr_{ij}, βr_{ij}, σr_{ij}), j=1,…,4, for each of the t rules (i=1,…,t).

All the other chromosomes are obtained by performing non-uniform mutation on this chromosome, with probability p_{mutat} (see table 5.6 for the values of GA_{tun} parameters). The range of values of the elements of the triplet (α_{ij}, β_{ij}, σ_{ij}), j=1,…,4, i=1,…,t, is:

$$\alpha_{\text{ij}} \in [\alpha r_{\text{ij}} - 1., \alpha r_{\text{ij}} + 1.], \qquad \beta_{\text{ij}} \in [1.5, 5.0], \qquad \sigma_{\text{ij}} \in [0.1, 3.0].$$

5.2.4.1.3 Evaluation of the fitness of each chromosome in GA_{tun}

The parameters defining the membership functions of the fuzzy sets in the antecedent of the rules are tuned in order to minimise the mean square error between the output inferred from the fuzzy rule base specified in chromosome C and the output given in the example for a specific input value, for all examples in a test example set.

The fitness function of GA_{tun} for a chromosome C and a test set of n examples, TD, is defined as:

$$F(C) = \begin{cases} \dfrac{1}{2n}\left[\sum_{i=1}^{n}\left(ey_i - FI(ex_i)\right)^2\right], & \text{if } CD(TD,\Delta_C) \geq \varepsilon \\ \dfrac{1}{2n}\sum_{i=1}^{n}(ey_i)^2, & \text{otherwise} \end{cases} \qquad \text{(T2)}$$

where FI (ex_i) is the output value obtained after inferring from a fuzzy rule base Δ_{C} for input vector ex_i (see section 3.4.5 for details on the fuzzy inference algorithm). Δ_{C} is obtained from chromosome C, by attaching the values of the consequent variable, obtained for each of the rules in RB_{simp}, to the correspondent set of 4 triplets (α, β, σ) in C that define the fuzzy sets of the variables in the antecedent of the rules.

If the coverage degree of an example set E_{h} by a rule set Δ_{C}, $CD(TD,\Delta_C)$, given by $CD(TD,\Delta_C) = \bigcap_{i=1}^{n} CV_{\Delta_C}(e_i)$ does not satisfy condition (C3'), then the value $(ey_{\text{i}})^2$ is added to the current sum instead of the squared difference between ey_{i} and FI (ex_k).

5.2.4.1.4 GA_{tun} genetic operators

The genetic operators chosen for GA_{tun} are the same used as the ones used in GA_{gen}, given that GA_{tun} is also a real coded GA. These are the non-uniform mutation operator proposed by Michalewicz (see definition 5.8) and the max-min-arithmetical crossover proposed by Herrera et al (see definition 5.10). Elitist selection (see definition 5.4) is used to select the individuals for the next generation. Linear ranking (see definition 5.2) and stochastic universal sampling (see [Bak, 1987]) are used to select the individuals for reproduction.

5.2.4.1.5 GA_{tun} configuration parameters

The values selected for GA_{tun} parameters are the same as the ones selected for GA_{gen} (see table 5.4), except the value of the coverage parameter which takes the value 1.

Table 5.7 *GA_{tun}* configuration parameters.

Parameter description	***Parameter value***
maximum number of generations	$maxgen = 200$
population size	$popsize = 100$
probability of crossover	$p_{\text{cross}} = 0.6$
probability of mutation	$p_{\text{mutat}} = 0.6$
probability of gene mutation	$p_{\text{gene}} = \frac{p_{mutat}}{l}$ (l is the length of the chromosome)
max-min arithmetic crossover parameter	$a = 0.35$
non-uniform mutation parameter	$b = 5$
maximum expected value for ranking	$\eta_{max}, = 1.1$
coverage parameter	$\varepsilon = 1$

5.2.4.1.6 GA_{tun} package

GA_{tun} uses the same package as GA_{gen}: the GEnetic algorithm for Numerical Optimisation for Constrained Problems or GENOCOP, version 2.0, proposed by Michalewicz (see also [([Mich, 1992b] or [Mich, 1992], chapter 7).

5.3. Conclusions

The algorithm presented in this chapter enables to automatically design a fuzzy logic based system for CAC (FCAC) using a set of examples that express the relation between the traffic characteristics of a particular traffic scenario and the maximum cell loss obtained for each of the connections.

The algorithm uses GA to generate both the fuzzy rules and the fuzzy sets associated with each rule. GA are used also to simplify the rule base previously obtained and to tune the fuzzy sets of each of the rules. Moreover, the tuning process can be invoked every time a new set of test examples is available, so allowing FCAC to capture changes in the traffic patterns that have an effect on the maximum cell loss ratio measured per connection.

The cell loss ratio mentioned in the previous paragraphs refers to the cell loss ratio measured in a steady state, that is when the cell loss values obtained are reduced to the influence of burst and rate-variation scales and the cell level influence has been smoothed by the output buffer.

In the literature, other learning methods for the automatic design a fuzzy logic system (definition of fuzzy rules and fuzzy sets) from examples have been presented, more precisely learning methods based on neural networks (NN):

- Fontaine and Smith ([Font, 1996]) proposed a neuro-fuzzy approach to CAC that does not take into account the size of the bursts and predicts the average cell loss probability for a specific traffic scenario. This is an hybrid approach that, using the authors own words, takes “advantage of both the learning capabilities of NN and the interpretability properties of fuzzy logic, preventing the scheme from being only a black box from which it would be impossible to interpret and understand the knowledge learnt by a NN” ([Font, 1996]);

- Hiramatsu [Hiram, 1995] uses NN to predict the average of the cell loss probability for a network state described by a vector containing the number of connections per traffic classes of audio, video and data connections. The use of user-declared traffic parameters to describe traffic scenarios is not recommended by Hiramatsu given that "users declare their traffic parameters before they start their connection, usually they declare the traffic parameters for the expected maximum traffic" and, also "when the number of connections is large, it is not easy to estimate QoS by using all those parameters" [Hiram, 1995];
- Khalil et al ([Khal, 1994] also propose a NN tool in which the network state is modelled using a vector consisting of the number of connections per traffic class.

The main differences between the NN approach proposed by Khalil et al and the one proposed by Hiramatsu's are:

- the method used to train the NN. Khalil et al use off line training and back propagation and Hiramatsu proposes on-line training using the relative target method;
- the method used to obtain the training data. Khalil et al use training data obtained from a method for calculating individual cell loss probabilities advocated by [Miyao, 1993] and Hiramatsu proposed a new method of monitoring, every 100 ms, the cell loss ratio obtained for virtual output buffers, the "virtual output buffer method" (see [Hiram, 1995] for details);
- the cell loss value targeted. The NN proposed by Khalil et al predicts the cell loss ratio per traffic type whereas the NN proposed by Hiramatsu predicts the average of the cell loss distribution for a particular traffic scenario.

Approaches that categorise the traffic connections in classes face the same challenge as approaches that use the user-declared traffic parameters to model the network state: an accurate source characterisation method is necessary either to identify the traffic class that a source belongs to or to determine the traffic parameters that define the traffic behaviour of the source. These studies are beyond the scope of this research but can be found in [ITU I371, 1995], [ATM-F, 1993], [Andr, 1994], [Lel, 1993], [Eck, 1991], [Magl, 1988] and [Heff, 1986] among others.

As far as the learning method is concerned, both GA based methods and NN are valid methods to automatically design a fuzzy logic based system for CAC purposes, as long as the training data is sampled carefully via off-line or on-line methods.

The learning method based on GA presented in this chapter fixes the number of fuzzy variables in the antecedent and consequent of the rules. Apart from that, the NN based approaches previously referred also assume as fixed the number of fuzzy sets of the domain of each fuzzy variable. The learning method defines the membership functions for each of the fuzzy sets and the fuzzy rules using the knowledge (implicitly) expressed in the training data.

The sampled data used by a learning mechanism applied to CAC (either based on NN or on GA) in order to adapt to changes in traffic patterns, can be collected via off-line or real-time (on-line) sampling. In off-line sampling, the learning system is trained with a set of data already sampled, or data obtained from computer simulations. Real-time sampling, on the other hand, uses data monitored in real time from an ATM node. If a representative data set or an accurate model of the buffer behaviour exists, the learning system can be installed on the ATM node after tuning with off-line data, without tuning with real time data. But usually there are differences between the simulations and the actual situation or the user-declared traffic parameters do not describe accurately the generated traffic characteristics. This is the reason why tuning a system with real time data is advisable in order to adjust for these errors (see also [Hiram, 1995]). In the experiments reported in the next chapter, data monitored on a real ATM link has been used as much as it was possible. In the absence of this type of data, computer simulations were used to collect data for off-line training using a tool developed by Pitts ([Pitts, 1993]).

In this thesis, no mechanism for monitoring the cell loss training data is proposed. This is because the only monitoring facility available in the ATM control plane is the UPC function but other monitoring mechanisms can become available on a real ATM network (e.g. a source characterisation tool). FCAC requires that the cell loss ratio for all active connections be monitored for a period of time in which there are no changes in the network state (no connections acceptances or connections releases) and the cell loss refers to the cell loss obtained in a steady state, that is, when the cell level influences on the obtained cell loss value are no longer predominant.

The cell loss ratio predicted by FCAC is expressed in terms of an integer value representing the negative power of ten associated with the maximum cell loss ratio per connection. The reason for not predicting a cell loss ratio with floating point precision, has to do with the reduction envisaged on the number of examples used in the adaptation mode algorithm and also on the time spent in on-line training. This might not be an advantage on a real system because there will not be a scarcity of training data but it is worth to be considered if the system has to adapt within a short period.

The actual performance of the FCAC learning mechanism in terms of learning times and accuracy of the cell loss prediction can only be evaluated using real traffic in an ATM link. Nevertheless, in the next chapter some experiments are reported on the performance of FCAC, using artificiallly generated traffic in an ATM test bed. A comparison of the cell loss prediction given by FCAC is also compared with cell loss predictions given by theoretical CAC approaches and a learning CAC approach based on a combination of fuzzy logic and NN.

6 Experiments

The experiments presented in the following sections were designed to evaluate:

- the accuracy of the cell loss ratio (CLR) predicted by FCAC, compared with predictions using other CAC approaches,
- the ability of FCAC to generalise its prediction capabilities when applied to traffic scenarios for which there is no previous knowledge,

in the presence of:

- homogeneous (same type of sources) and heterogeneous (different types of sources) traffic mixes composed of On-Off traffic sources; the type of sources is chosen according to published suggestions in [Yang, 1996] and [R2061 D.28, 1994];
- ATM network configurations with short and large output buffer sizes and for various link transmission speeds.

The comparison criteria for different CAC approaches will focus on the simplicity of development, speed of response and accuracy of the CLR prediction envisaging to maximise the use of network resources. The comparison should make it possible to find the strong and weak points of each method in the presence of different traffic and link configurations.

The experiments on comparative CLR prediction consider the following methods:

- the fuzzy-based decision support system for CAC (FCAC) proposed in this thesis;
- the convolution-based CAC developed by Marzo et al ([Marzo, 1993]), and referred to using the authors terminology, as "enhanced" convolution approach (ECA);
- the (M+1)-state MMDP approximation proposed by Yang and Tsang ([Yang, 1995]);
- the neuro-fuzzy CAC developed by Fontaine et al ([Font, 1996]), in the following referred as NFCAC.

The convolution and the (M+1)-state MMDP approximation are examples of CAC approaches based on a theoretical model of the traffic behaviour. The first has been chosen because it provides an upper bound estimation of the average CLR to be obtained for an ATM link with a small output buffer (capable of queueing the cell level fluctuations). The latter provides a tighter bound for the average CLR prediction particularly when the buffers are large (the convolution approach does not take in consideration the actual size of the buffer) and the traffic exhibits long bursts (e.g. data transfer, image retrieval).

The neuro-fuzzy CAC approach and the fuzzy-based CAC (FCAC) share the same modelling philosophy in the sense that are both based on fuzzy logic to represent the traffic knowledge and use ATM traffic data, collected from measurements on an ATM node, to acquire knowledge on possible traffic patterns and its relation with the expected CLR. The difference between NFCAC and FCAC is:

- NFCAC uses neural networks instead of genetic algorithms as the learning methodology;
- NFCAC assumes that the maximum number of fuzzy sets per input and output fuzzy variable is fixed and learns the fuzzy relations between variables (rules) and the associated fuzzy sets. FCAC assumes neither of the latter as fixed;
- NFCAC models the traffic using the same input variables as FCAC, except that NFCAC does not consider the fourth input variable, representing the mean buffer allocation. Therefore, NFCAC does not take into account the burst length of each of the connections (space resource).

The CLR predicted by the (M+1)-state MMDP approximation, ECA and FCAC are compared with CLR results obtained using a cell rate simulator, LINKSIM, developed by Pitts ([Pitts, 1993]). LINKSIM was chosen because

> "it accurately models the burst scale component of queueing in ATM buffers which is the key behaviour determining cell loss, [and also because] in comparison with cell by cell simulation, cell rate simulation enables the low cell loss probabilities required of ATM networks to be measured within reasonable computing times" ([Pitts, 1993], pp. 165).

The CLR predicted by NFCAC and FCAC will be compared with CLR results monitored on an ATM test-bed by RACE Consortium EXPLOIT ([R2061 D.28, 1994]). See also table 6.1 for the layout of the test experiments.

The reason why not all the CLRs were obtained from on-line measurements is that there was only a limit period of time to perform the measurements and also because the equipment available at the ATM test bed only allows to monitor a small number of sources at a time. Hence, LINKSIM was used to calculate the CLR for the other traffic experiments reported in this chapter.

Table 6.1 Experiments layout.

Traffic mix ***(Experiments)***	***Link Capacity*** ***(Mbit/s)***	***Buffer size*** ***(cells)***	***FCAC*** ***(max. CLR per connection)***	***NFCAC*** ***(max. CLR per connection[1])***	***ECA*** ***(average CLR)***	***(M+1)-MMDP*** ***(average CLR)***	***Reference*** ***(max. CLR per connection)***
homogeneous[2] (A.1 to A.6)	variable	50	yes		yes	yes	simulations
homogeneous (A.7)	7	2000	yes		yes	yes	simulations
homogeneous (B1. to B.6)	155.52	27	yes	yes			on-line measurements
2 traffic types (C.1 to C.8)	155.52	27	yes	yes			on-line measurements
6 traffic types (D.1 to D.9)	155.52	27	yes				on-line measurements and simulations

[1] In [Font, 1996], the average cell loss probability is used. For the comparative experiments, both FCAC and NFCAC, have been trained to predict the maximum CLR for the connections.

[2] The average CLRs predicted by ECA and the (M+1)-MMDP approximation are compared with the maximum CLR per connection given by FCAC, because for homogeneous traffic scenarios these ratios are approximately the same.

6.1 Comparative cell loss ratio experiments (theoretical CAC approaches)

The approaches for CLR prediction considered in the experiments described in this section are:

- the *enhanced convolution* approach (ECA) developed by Marzo et al. ([Marzo, 1993]) is a CAC approach inspired by the convolution algorithm (see also [Ivers, 1990]). The convolution algorithm determines the exact distribution of the aggregated bit rate on the link. It assumes that the behaviour of the traffic sources is independent from each other and does not take into account the duration of the bursts. This algorithm has a high cost in terms of calculation and storage requirements and the calculation time increases with the number of multiplexed connections. ECA aims to overcome these drawbacks and still to obtain an accurate CLR prediction: the connections are classified in traffic classes whose rate distribution is modelled using the multinomial distribution function. Real-time calculations are possible because a set of mechanisms have been specifically designed for the algorithm to reduce the computational times;
- the approach to estimate the cell loss probability in an ATM multiplexer for traffic scenarios with the same type of traffic sources (homogeneous traffic) described by Yang and Tsang in ([Yang, 1995]), that uses the Markov Modulated Deterministic Process (MMDP) to approximate the actual arrival process and models the ATM multiplexer as an MMDP/D/1/k queueing system. Cells arrive according to a deterministic renewal process whose rate is controlled by a Markov process $\{X(t), t\geq 0\}$. Using queueing analysis, Yang at al derived a formula for the cell loss probability expressed in terms of the limiting probabilities of a Markov chain. Two approximations methods were proposed based on the analysis results. The first method approximates the actual arrival process by an $(M+1)$-state MMDP, where M is the number of identical sources. The modulating Markov process $X(t)$ is the

number of sources that are active at time t (hence, the MMDP has M+1 states). This approximation provides accurate results "for all cases in which burst-level congestion is the main contributing factor of loss" ([Yang, 1995]). The second method models the arrival process using a two-state MMDP and the two states of the modulating Markov process $X(t)$ are matched in such a way that state 0 characterises the underload region of the number of sources that are active at time t (the total cell arrival rate in any state is less than or equal to the link capacity) and state 1 represents the overload region (the total cell arrival rate in any state is greater than the link capacity). This approximation is "sufficiently accurate for applications where the average burst length is large (such as large file transfers, image retrievals, etc.)" ([Yang, 1995]). In this study only the first method, the (M+1)-MMDP approximation, is considered because it is sufficiently accurate to estimate the cell loss probability for the scenarios studied.

The traffic sources used in the experiments are On-Off sources defined by parameters: peak and mean bit rate and mean burst length (see also table 6.2) and are in conformance with the traffic characteristics for the On-Off sources presented by Yang et al in [Yang, 1995], table II, pp. 122. The On-Off sources considered in each of the traffic mixes are assumed to be independent of each other.

Table 6.2 Traffic characteristics of On-Off sources.

Traffic Class	*Peak Rate (Mbit/s)*	*Mean Rate (Mbit/s)*	*Burst Length (cells)*
Voice	0.064	0.022	58
Data	10	1	339
Image	2	0.087	2604

In order to obtain the cell loss ratio for each of the traffic scenarios considered in the experiments, simulations were executed in 8 SOLARIS2 (SUN) workstations using the ATM cell rate simulator, LINKSIM ([Pitts, 1993]), for two months. For each traffic scenario, the criterion for stopping the simulation is that the width of the 95%

confidence interval should be less than 10% of the estimated cell loss probability. For traffic scenarios with a very high number of low rate sources, LINKSIM failed to produce a result satisfying the stopping criterion after running during two months. Hence, the simulations were forced to stop before they met the criterion. For these cases, the average and maximum CLR were considered to be the same and its value was read from the simulation result obtained by Yang et al for the specific traffic scenario (see [Yang, 1995], pp.123-4). These results are signalled with an asterisk in the tables presented in the Appendix that show the CLR obtained for experiments A.1 to A.6.

The CLR obtained from simulations for experiments A.1 to A.6 (table 6.3) was used to construct a set of training examples (see in chapter 5, section 5.2.1.1 how to construct this example set). The identification of the set of fuzzy rules and the definition of the fuzzy sets for each of the rules, that is, the design of FCAC, was obtained automatically, applying the method described in chapter 5, and using the knowledge implicitly stated in the set of training examples.

Table 6.3 Specification of the network scenario and traffic types for the experiments with homogeneous traffic mixes (theoretical CAC approaches)

Experiment	*Buffer size (cells)*	*Link Capacity (Mbit/s)*	*Traffic type*	*Figure*
A.1	50	7	voice	6.1
A.2	50	0.7	voice	6.2
A.3	50	350	data	6.3
A.4	50	52	data	6.4
A.5	50	30	image	5.5
A.6	50	7	image	6.6

The experiments consist of multiplexing homogeneous traffic on the same ATM link for various traffic mixes and evaluating the CLR predicted by each of the CAC approaches:

(M+1)-MMDP, ECA and FCAC (see figures 6.1 to 6.6). The traffic mixes are obtained varying the number of traffic sources in the mix. The number of sources chosen for the experiments and the link capacity values were chosen in order to get a wide range of CLRs (between 10^{-2} and 10^{-8}).

All calculations were performed on Sun SPARC stations running SOLARIS2 and both the fuzzy-based CAC (FCAC) and the convolution-based approach (ECA) were developed in ANSI C.

Experiments A.1 to A.6 refer to a single ATM link. The results are plotted in figures 6.1 to 6.6 and show:

- the average CLR predicted by the (M+1)-MMDP approximation and ECA approach (against the average CLR obtained via simulations),
- the maximum CLR per connection predicted by FCAC (against the maximum CLR obtained also via simulations).

Given that the traffic scenarios studied in experiments A.1 to A.6 are composed of homogeneous traffic mixes, the maximum CLR per connection and the average CLR for all connections obtained by simulation are not significantly different (their difference is less than 10^{-1}). Therefore, comparative results can be shown for the CLR predicted by FCAC and the CLR predicted by ECA and the (M+1)-MMDP approximation.

Note that the FCAC prediction is given in terms of an integer representing the negative power of ten for the maximum CLR per connection, whereas the predictions given by the (M+1)-MMDP approximation and ECA are given in terms of a floating point value for the average CLR. Hence, differences in the predicted CLR which are less than 10^{-1} are not considered for comparison purposes.

The value of the CLR considered in the examples used to train FCAC, is obtained by rounding the floating point CLR obtained by simulations to the nearest power of ten, that is, a CLR with a value of 4.13×10^{-3} is actually "seen" by FCAC as 10^{-3} and a CLR value of 5×10^{-3} as 10^{-2}. This can be easily changed if, in a real implementation, the CLR predicted becomes too optimistic and, therefore, a more conservative prediction is required. For example, a CLR measured as 4.13×10^{-3} might be truncated to 10^{-2}, instead of 10^{-3}.

6.1.1 Voice traffic

The traffic studied in figures 6.1 and 6.2 is composed of homogeneous traffic mixes of lightly load traffic: the mean bit rate of each traffic source is 34% of the peak bit rate and the ratio of the peak bit rate of each source to the total link capacity is 9.14×10^{-3}, increasing to 9.14×10^{-2} in figure 6.2.

The CLR predicted by ECA is quite accurate for the traffic scenarios plotted in figures 6.1 and 6.2, although the curve of the average CLR obtained by simulation is above the corresponding value predicted by ECA in figure 6.1 and below in figure 6.2 (ECA ignores the decrease in the CLR obtained by queueing part or the whole burst).

The CLR predicted by the (M+1)-MMDP approximation is slightly more accurate than the corresponding ECA prediction in the case of figure 6.1, but can be very optimistic in the case of figure 6.2 for low average loads; for a traffic mix with 15 sources the CLR predicted by the (M+1)-MMDP approximation is 3×10^{-6} and the value obtained by simulation is 5×10^{-5}. This is because the influence of the burst scale on the obtained CLR, for low average loads (less than 66% of the link capacity), is not significant when the mean burst length of the traffic sources is small.

The CLR predicted by FCAC conforms with the value for the CLR given by simulation, except in figure 6.1 for a traffic mix with 260 sources. This is because the value read from the simulation results for the maximum CLR increases dramatically from 1.09×

10^{-5} (250 voice sources) to 2.59×10^{-4} (260 voice sources). FCAC cannot make any sense of this change given the reduced number of training examples (the total number of training examples is 37 and only 6, among these, refer to the traffic mixes studied for experiment A.1).

Experiments A.1 and A.2 allow the observation of the influence of the time resource (link capacity) for very low bit rate sources with short mean burst length (58 cells). For a CLR requirement of 10^{-4}, a link capacity of 7Mbit/s (experiment A.1) enables to achieve a higher mean load than a link of 0.7Mbit/s (experiment A.2): 81% and 47% of the link capacity, respectively.

6.1.2 Data traffic

The traffic studied in figures 6.3 and 6.4 is composed of data traffic, for which the mean bit rate of the traffic sources is 10% of the peak bit rate and the mean burst length is 339 cells.

The CLR predicted by ECA provides an upper bound for the average cell loss probability for the traffic scenarios plotted in figures 6.3. and 6.4. The prediction given by the (M+1)-MMDP approach is also accurate and the accuracy increases with the average load as observed previously for a slow link (see figure 6.2). FCAC predicts a CLR in the range of the value obtained by simulation, that is with a maximum error of 10^{-1}. In figure 6.4, the values obtained via simulations for the maximum and average CLR differ significantly for a traffic mix of 12 sources (more than 0.05). This explains why FCAC predicts a more conservative value than ECA and (M+1)-MMDP for this traffic pattern.

The traffic sources that form the traffic mix for the experiments depicted in figure 6.4, have a high peak to link ratio (19%) compared to the same ratio for the traffic sources of figure 6.3 (0.03%) and this explains why the allowed average load, using FCAC for a CLR requirement of 10^{-4}, is much less in figure 6.4 (19% for a traffic scenario of 10 sources) then the allowed average load in figure 6.3 (62.8% for a traffic scenario of 220 data sources).

Yang et al ([Yang, 1995]) describe experiment A.3 (figure 6.3) as a traffic scenario where “burst-level congestion” is predominant and that accounts for the accuracy of the (M+1)-MMDP prediction for this traffic scenario. The mean burst length of the traffic sources is quite long, 339 cells, and therefore, the influence on the CLR obtained caused by queueing part of the burst is more significant than for voice traffic. Experiment A.4 (figure 6.4) is described in [Yang, 1995] as the traffic scenario obtained when transferring large files across the link; the output buffer only copes with cell level fluctuations and all the loss is caused by lack of bandwidth resources. This constitutes an ideal scenario for the application of the ECA approach.

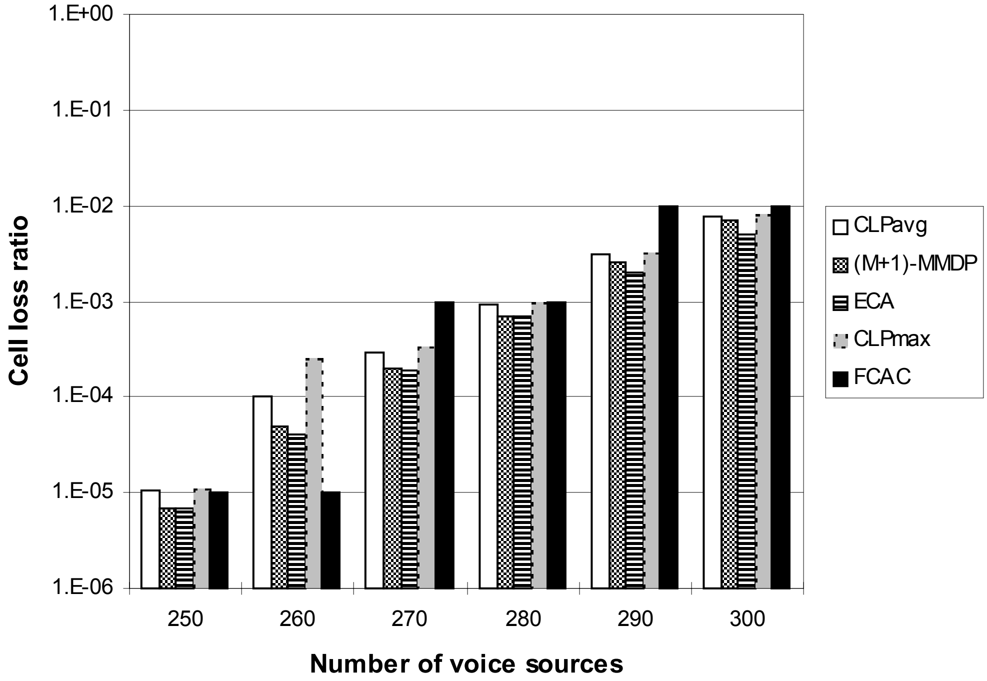

Figure 6.1 Traffic scenarios of *voice* sources (C = 7 Mbit/s and K=50) (fig 2 of [Yang, 1995], pp123).

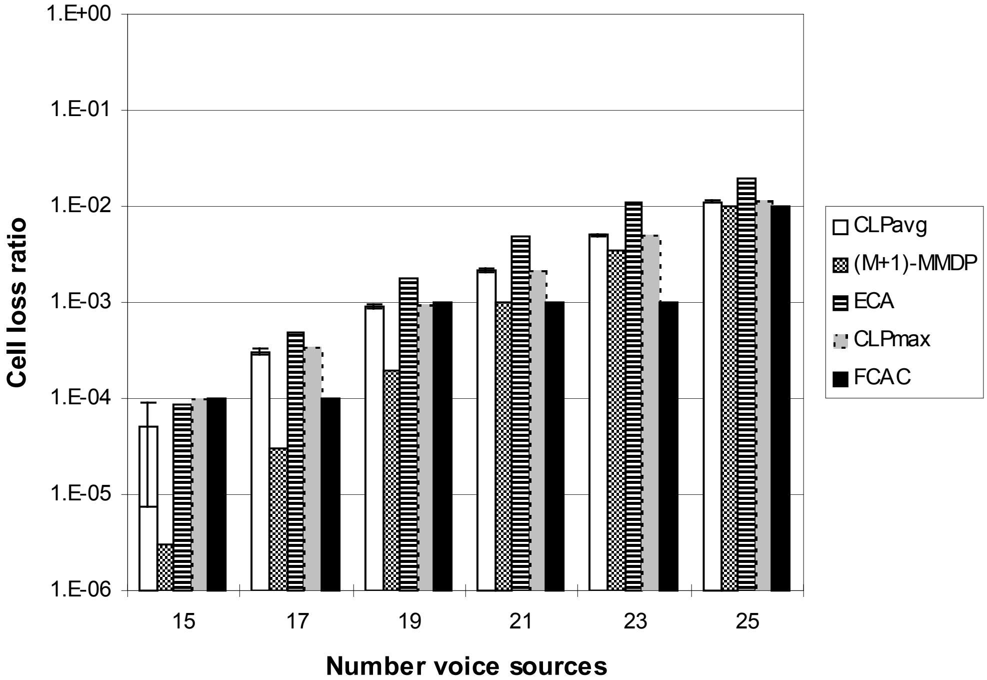

Figure 6.2 Traffic scenarios of *voice* sources (C = 0.7Mbit/s and K=50) (fig 8 of [Yang, 1995], pp123).

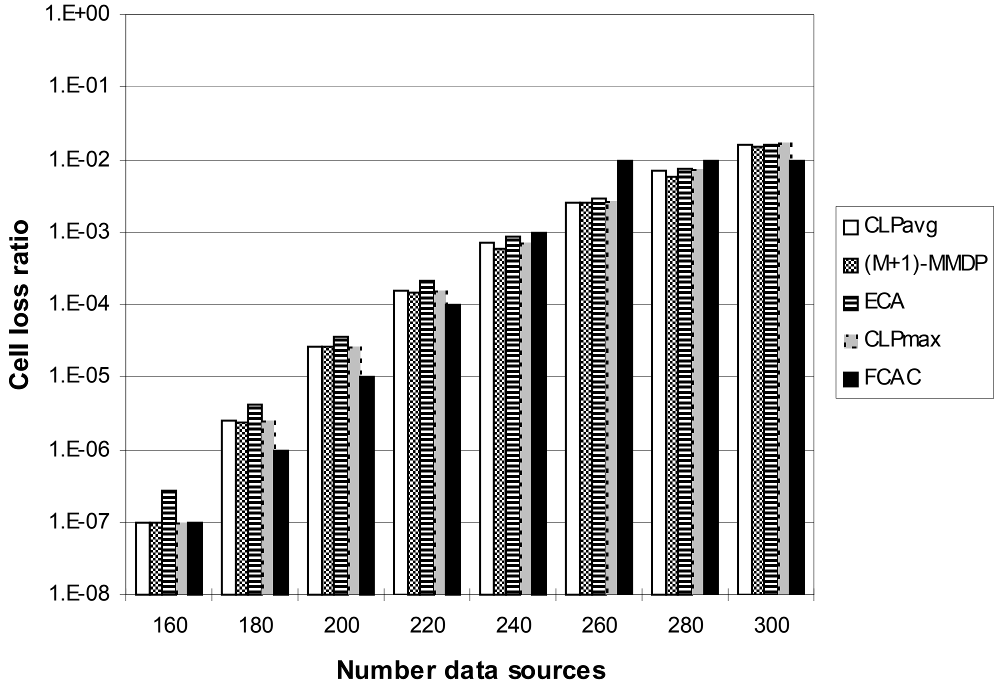

Figure 6.3 Traffic scenarios of *data* sources (C = 350 Mbit/s and K=50) (fig 3 of [Yang, 1995], pp123).

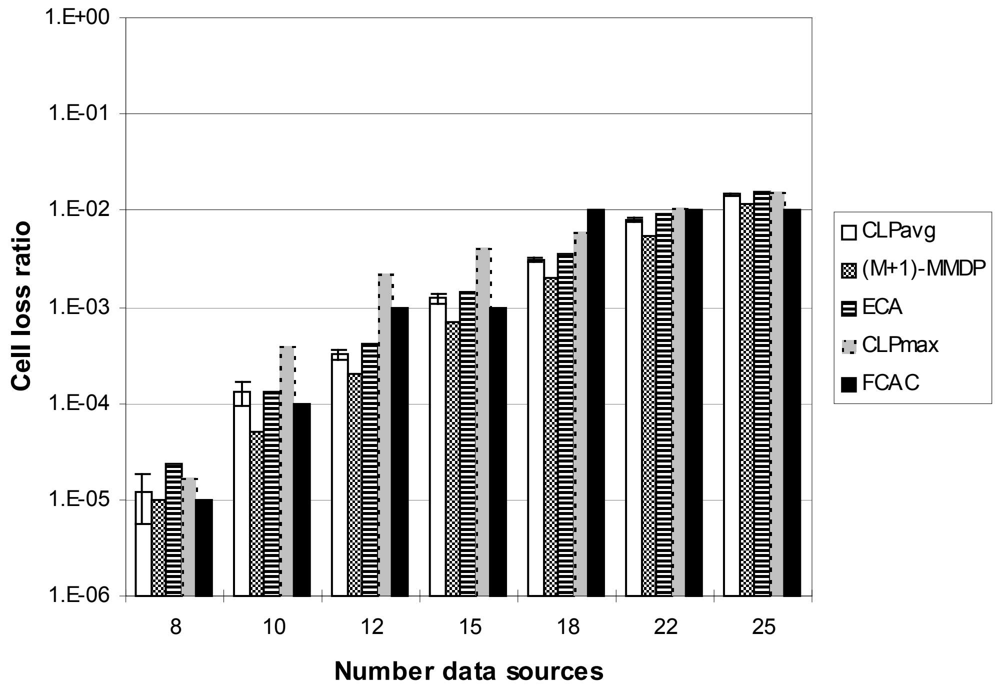

Figure 6.4 Traffic scenarios of *data* sources (C = 52 Mbit/s and K=50) (fig 9 of [Yang, 1995], pp123).

6.1.3 Image traffic

The traffic studied in figures 6.5 and 6.6 is composed of traffic for which the mean bit rate is 4% of the peak bit rate and the mean burst length is very high: 2604 cells. The peak to link ratio of the sources is small in the case of figure 6.5 (0.066), but quite high in figure 6.6 (0.28).

The CLR predicted by ECA is below the curve corresponding to the average CLR obtained via simulations, for both figures 6.5 and 6.6a, although the difference between the two values is always less than 10^{-1}. A similar comment is given for the CLR predicted by the (M+1)-MMDP. The CLR predicted by FCAC is as expected when reading the value obtained via simulations for the maximum CLR per connection.

Yang et al ([Yang, 1995]) describe experiment A.5 (figure 6.5) as a traffic scenario where “burst-level congestion” is predominant and this accounts for the accuracy of the CLR predicted by (M+1)-MMDP. Nevertheless, the long term influence on cell loss caused by the successive arrival of bursts is not taken into account by the traffic model used in the (M+1)-MMDP approximation and, thus, the predicted CLR is slightly optimistic. In both experiments A.5 and A.6, the mean burst length of the sources is very long (2604 cells) and the output buffer size chosen (50 cells) cannot cope with fluctuations at cell level. This explains why the ECA approach does not provide an upper bound for the average CLR both in figures 6.5 and 6.6.

Figure 6.6b shows an example of a traffic scenario for which ECA gives a very pessimistic prediction and (M+1)-MMDP cannot provide a prediction at all, given the complexity of the calculations involved (see on chapter 7 a discussion on the complexity of the CAC approaches presented in this chapter). For this scenario, the characteristics of the traffic sources and the value of the link capacity are exactly the same as in figure 6.6a. The output buffer size is increased to 2000 cells. ECA is based on a bufferless model and thus, the reduction on the number of cells lost obtained by queueing part of the burst is not taken into account when predicting the cell loss ratio value. This explains why the CLR prediction given by ECA is so pessimist. FCAC takes into account the actual size of the output buffer of the switch, through the mean burst length to buffer size ratio and predicts a CLR which conforms with the results obtained by simulation.

Overall, it can be concluded that, for the traffic scenarios studied in figures 6.1 to 6.6a, the CLR predictions given by the (M+1)-MMDP approximation and the ECA approach are generally accurate when compared with the average CLR obtained via simulation. The FCAC predictions are also in the range of the expected maximum CLR values obtained via simulation (within a maximum error of 10^{-1}) for the totality of the experiments.

The network scenario shown in figure 6.6b is an example of a scenario where the application of FCAC can provide an advantage over ECA and certainly over (M+1)-MMDP. Such long output buffers might prove to be useful in switches specifically designed to take in consideration the long mean burst sizes that characterise data and image traffic.

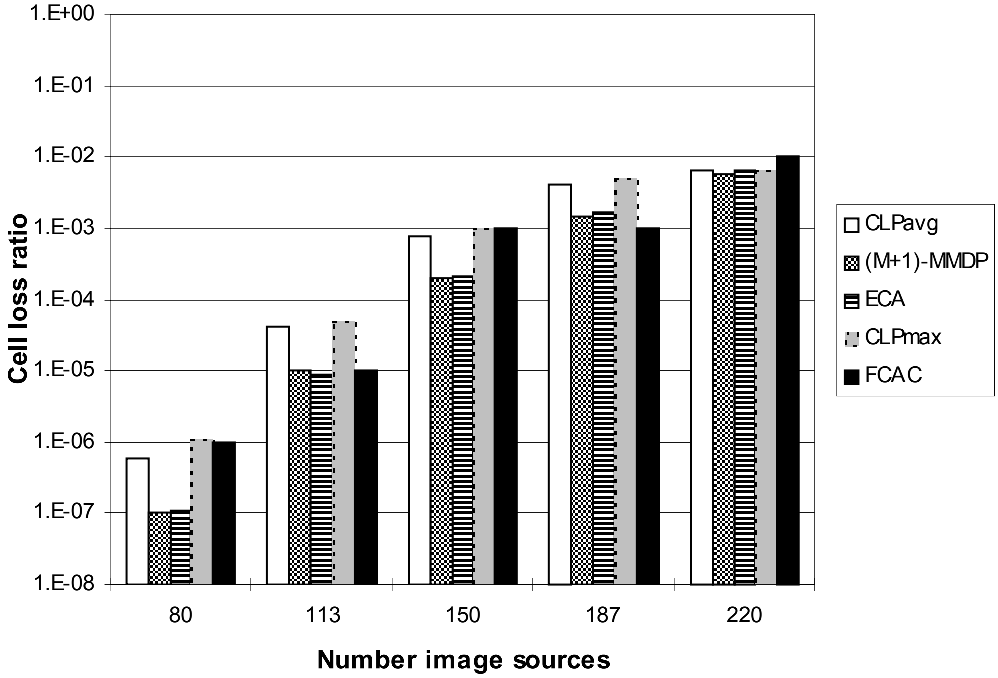

Figure 6.5 Traffic scenarios of *image* sources (C = 30 Mbit/s and K=50) (fig. 4 of [Yang, 1995], pp. 123).

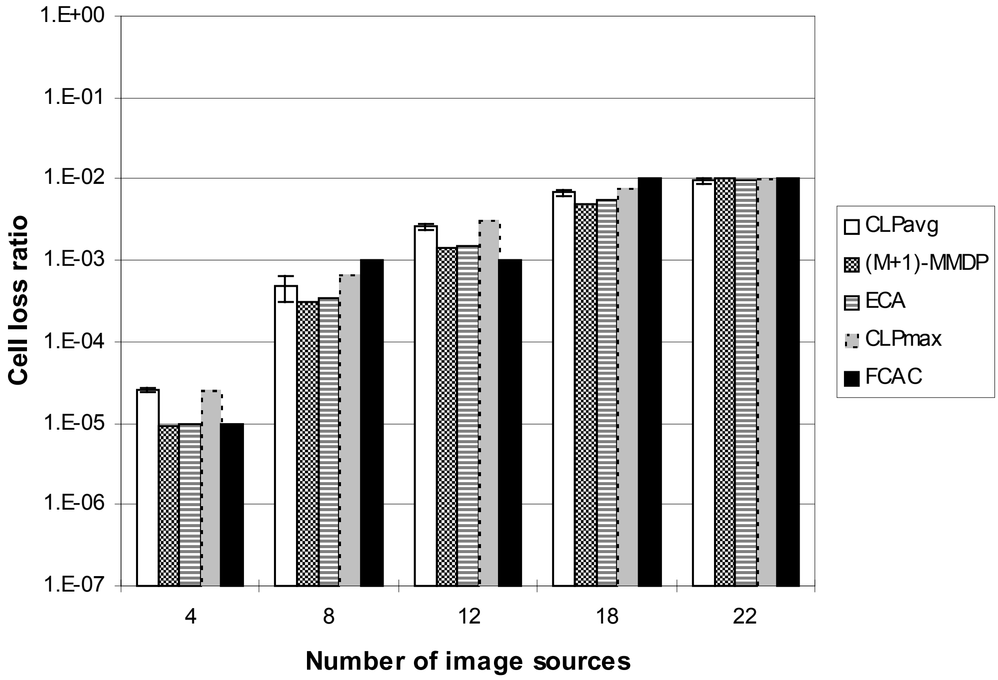

Figure 6.6a Traffic scenarios of *image* sources (C = 7 Mbit/s and K=50) (fig 10 of [Yang, 1995], pp. 124).

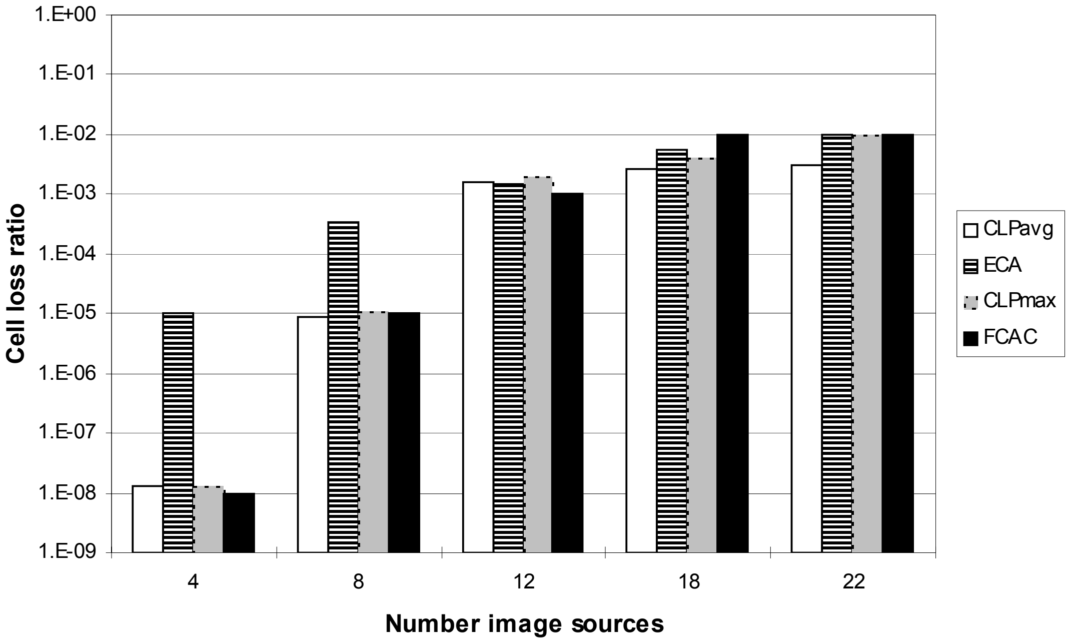

Figure 6.6b Traffic scenarios of *image* sources (C = 7 Mbit/s and K=2000)

6.1.4 FCAC generalisation results

The results shown in figures 6.7 to 6.12 refer to the CLR predictions obtained using FCAC for the same configuration of experiments as before (see table 6.3) but for an extended set of values for the possible number of sources in the traffic mix. The value shown in the first Y-axis (left side of the graphs) is the predicted maximum CLR per connection for the number of sources in the traffic mix shown in the X-axis. The value shown in the second Y-axis (right side of the graphs) is the degree of certainty (value from 0 to 1) on the predicted CLR for that specific number of multiplexed traffic sources. The degree of certainty corresponds, in terms of the implementation of FCAC, to the membership degree of the inferred CLR value in the fuzzy set resulting from the fuzzy inference process.

The minimum and maximum number of sources shown in the X-axis refer to the minimum and maximum number of sources for which FCAC could provide a CLR prediction, respectively. The results shown in figures 6.7 to 6.12 do not intend to evaluate the accuracy of the CLR predicted by FCAC by comparing it with the CLR obtained by simulation. The aim is, rather, to visualise the ability of FCAC to predict a CLR within a maximum error of 10^{-1}, considering traffic mixes for which the tool has no previous knowledge. The values obtained from simulations, $CLP_{\max}$, and used to construct the set of training examples, are shown in the figures only as a reference.

Figure 6.7 shows that FCAC behaves poorly (but still within the maximum error value) for traffic mixes of voice sources with a number of sources between 255 and 265, approximately. A CLR of 10^{-4} is expected for such traffic mixes but FCAC predicts a CLR of 10^{-5} for traffic mixes with a maximum of 261 voice sources and then "jumps" to 10^{-3}. A careful observation of the values obtained via simulations for $CLP_{\max}$ (shown as a triangle in the figures) for a traffic mix with 260 voice sources and the $CLP_{\max}$ obtained for a traffic mix with 250 voice sources reveals that the difference between the two $CLP_{\max}$ values is greater than 10^{-1}. This explains why FCAC cannot make much sense of the actual CLR for traffic mixes with a number of voice sources between 255 and 260 with such scarceness of training examples.

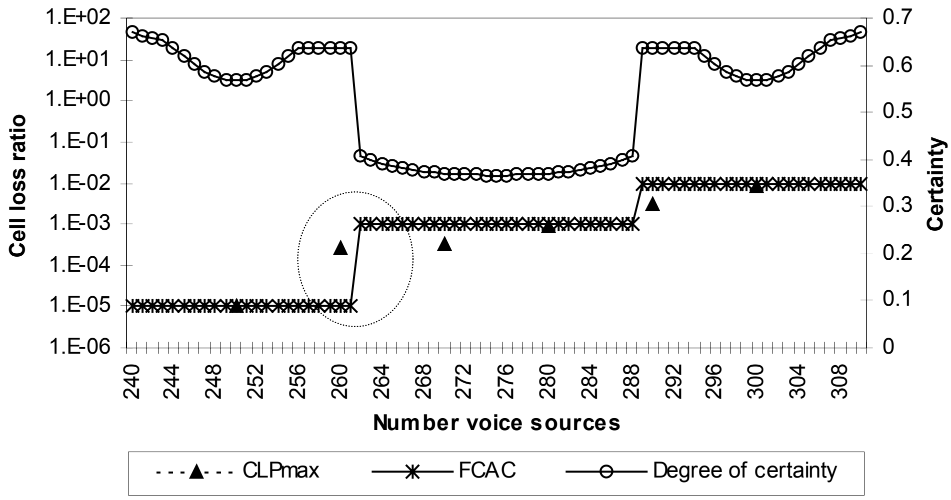

Figure 6.7 FCAC prediction for voice sources (experiment A.1, C = 7Mbit/s, K = 50 cells).

Figure 6.7 also confirms the intuitive rule that: "if the number of sources in the traffic mix is less than 250 sources, then the degree of certainty in predicting a CLR of 10^{-5}

will increase", and, by a similar reasoning, "if the number of sources in the traffic mix is more than 300, then the degree of certainty in predicting a CLR of 10^{-2} will increase".

For the traffic scenario shown in figure 6.8, the prediction given by FCAC is consistent with the results obtained from simulations. For this scenario, the peak to link ratio of the sources is much higher than for the traffic scenario of figure 6.7 (9.1×10^{-2} and 9.1×10^{-3}, respectively) and thus, the influence on the CLR caused by adding one more source to the existing traffic is more easily captured from the examples. A correct CLR prediction can, thus, be obtained even using a reduced number of examples.

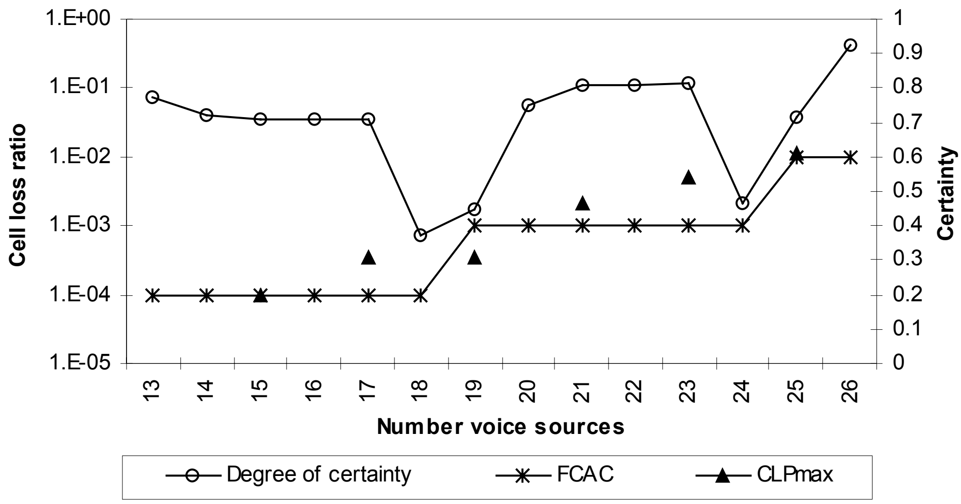

Figure 6.8 FCAC prediction for voice sources (experiment A.2, C = 0.7Mbit/s, K = 50 cells).

Figures 6.9 and 6.10 show the CLR predicted by FCAC for traffic mixes of data sources (experiments A.4. and A.5, respectively). Similar comments to those for figure 6.8 can be given for figure 6.10. The peak to link ratio of the sources in figure 6.10 is higher than the same ratio for the sources in figure 6.9 and thus, FCAC can easily extrapolate from the knowledge expressed in the examples (corresponding to traffic mixes with 8, 10, 12 15, 18, 22 and 25 data sources, see also figure 6.10). Hence, in the traffic scenario of figure 6.10:

- the maximum CLR obtained from simulations for a traffic mix with 10 data sources is 3.9×10^{-4} and the maximum CLR for 12 sources is 2.16×10^{-3}. FCAC reads these values as 1.e-4 and 1.e-3, respectively, due to the rounding process to the nearest negative integer power of ten (see chapter 5, section 5.2.1.1 for more details on the rounding algorithm). Hence, for traffic mixes with 11 sources, the CLR is likely to be approximately 5×10^{-4}. Because 5×10^{-4} is understood by FCAC as 10^{-4} (rounding process), a CLR of 10^{-4} has a higher degree of certainty than a CLR of 10^{-3} and, thus, FCAC predicts 10^{-4}. If, in future, it turns out that the rounding process causes the prediction given by FCAC to be optimistic, a more conservative rounding value can be used, that is, a CLR of 5×10^{-4} would be interpreted as 10^{-3}.

- for traffic mixes with 16 data sources a more conservative reasoning is applied because there is more traffic knowledge for traffic scenarios characterised by high loads than for traffic scenarios with low loads. The CLR value obtained via simulation for a traffic mix with 15 data sources is 4.13×10^{-3} and for a traffic mix with 18 data, 6.07×10^{-3}; FCAC reads these values as 10^{-3} and 10^{-2}, respectively. Hence, for 16 sources the predicted CLR value changes immediately to 10^{-2}.

In figure 6.9, the fluctuation of the value for the degree of certainty associated with the CLR predicted by FCAC can be observed. The fluctuation occurs when the CLR predicted by FCAC is about to change to the next CLR “in step”, that is, to a lower CLR if the load decreases or to a higher CLR if the load increases. For this traffic scenario, FCAC has enough knowledge to be able to make a generalisation to scenarios for which it has no previous knowledge. A CLR prediction that has an associated degree of certainty less than 0.5 implies a small degree of confidence in the actual value of the predicted CLR. For example, in figure 6.10, the CLR predicted for traffic mixes with a number of data sources between 12 and 15 is 10^{-3} but, given that the certainty value associated is less than 0.5, then there is a possibility that the actual observed CLR would be 10^{-2}.

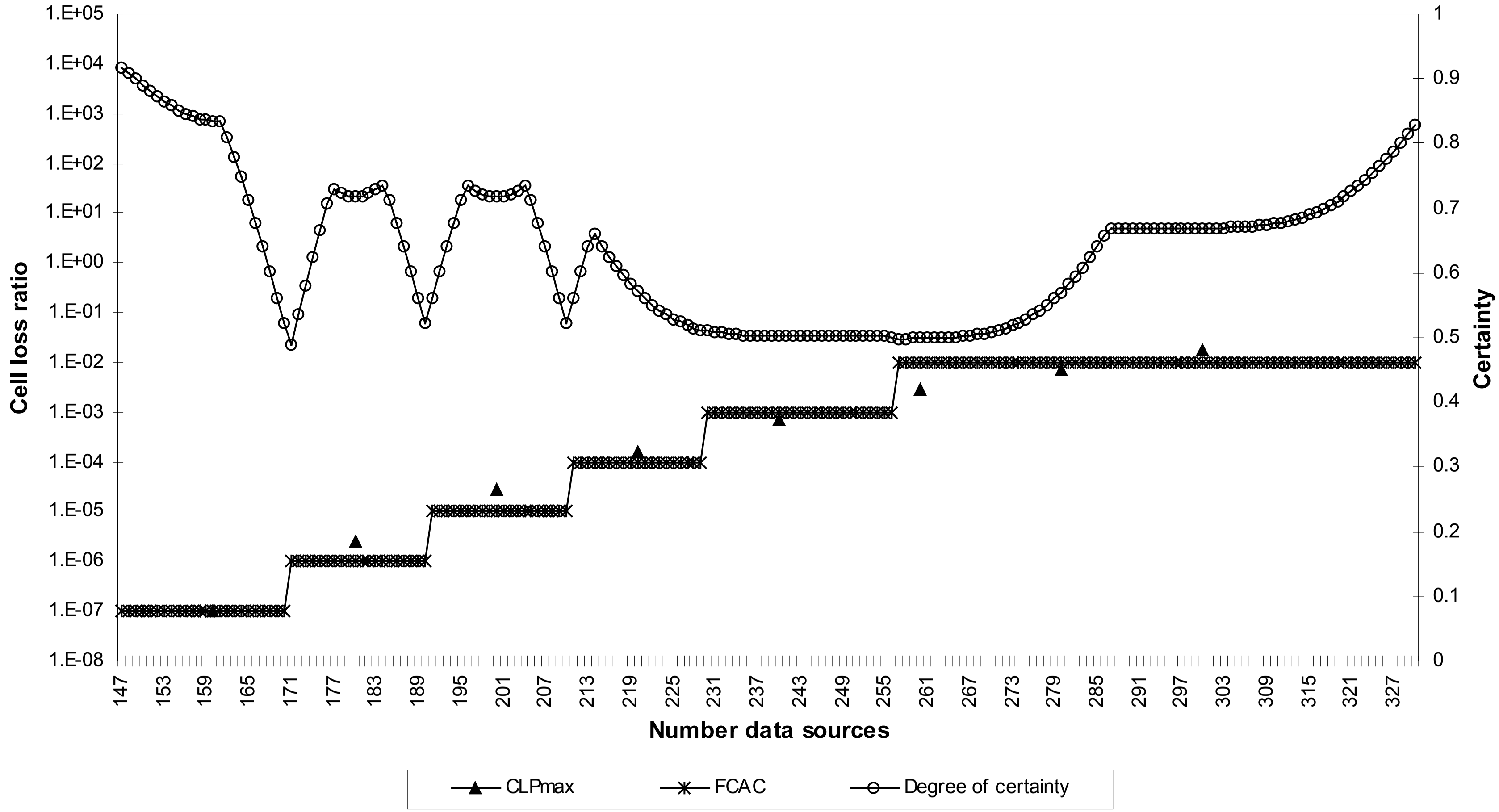

Figure 6.9 FCAC prediction for data sources (C=350 Mbit/s, K=50 cells).

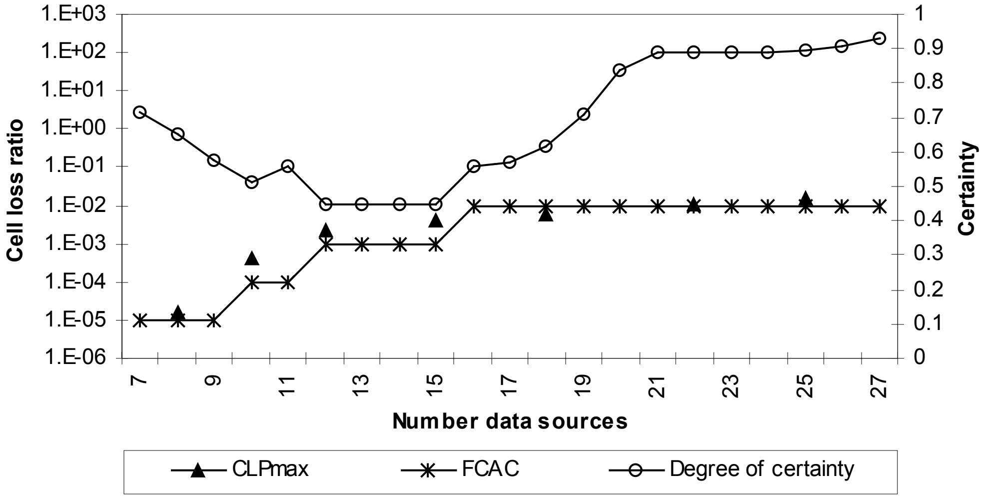

Figure 6.10 FCAC prediction for data sources (experiment A.4, C = 52 Mbit/s, K = 50 cells).

Figures 6.11 and 6.12 show the predicted CLR values for traffic mixes of image sources. As before, FCAC can generalise the CLR prediction to "new" traffic mixes for the traffic scenario for which the degree of granularity in the sampled data (training examples) is the highest. In figure 6.12, the sampled data refers to traffic mixes with 4, 8, 12, 18 and 22 image sources; in figure 6.11, the sampled data refers to traffic mixes with 80, 113, 150, 187 and 220 image sources. Thus, the degree of granularity in the sampled data is higher for the scenario shown in figure 6.12. This implies that FCAC can easily generalise on the relation between the characteristics of the multiplexed traffic and the expected CLR for the scenarios shown in figure 6.12, but the same is not necessarily valid for figure 6.11. This explains why, in figure 6.11, FCAC does not give any prediction for traffic mixes with 93 to 100 sources and 126 to 128 sources. In a real implementation, if a CLR prediction is not available, a conservative decision making based on peak rate allocation can be used until FCAC has knowledge to predict a CLR for new traffic patterns (see more on this in the conclusions, chapter 8).

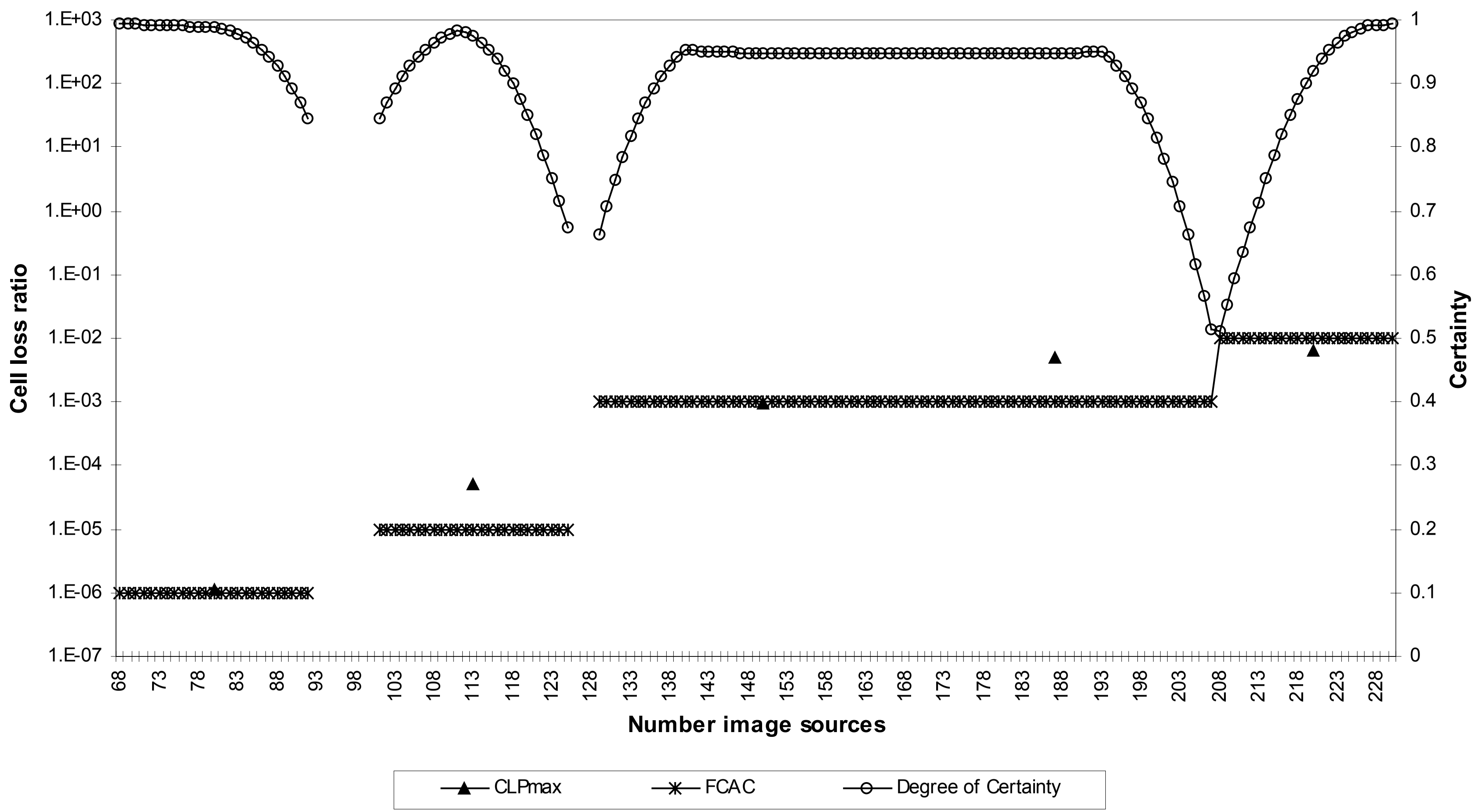

Figure 6.11 FCAC prediction for image sources (experiment A.5, C = 30 Mbit/s, K = 50 cells).

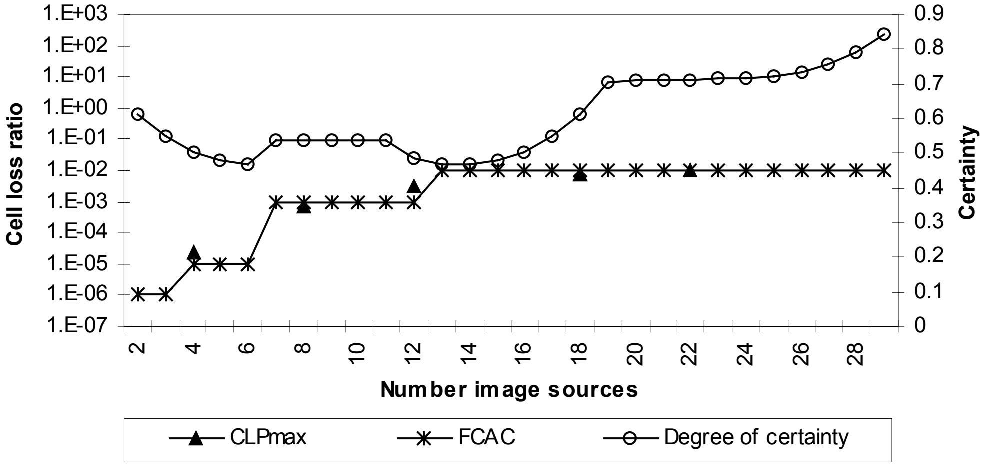

Figure 6.12 FCAC prediction for image sources (experiment A.6, C = 7 Mbit/s, K = 50 cells).

6.2 Comparative cell loss ratio experiments (learning CAC approaches)

This section reports the results obtained when using FCAC and a neuro-fuzzy based tool (NFCAC) developed by Fontaine et al (see [Font, 1996]) to predict the maximum CLR per connection in homogeneous and heterogeneous traffic scenarios. The CLR predictions obtained by both approaches were compared with on-line measured CLRs. These measurements were performed by RACE consortium EXPLOIT in the ATM test bed at Basel (see also [R2061 D.28, 1994]) and refer to the maximum CLR measured at the output buffer to the link for each of the connections. A single ATM link is considered, with link capacity 155.52 Mbit/s, and the output buffer of the switch has a storage capacity of 27 cells.

The traffic sources used in the experiments are On-Off sources defined by the peak bit rate and the mean on and off periods. The characteristics of the traffic sources are shown in table 6.4 (values obtained from [R2061 D.28, 1994]). The traffic mixes for homogeneous and heterogeneous scenarios described in table 6.5 are also the same as the ones considered in [R2061 D.28, 1994].

Table 6.4 Traffic characteristics of On-Off sources (from [R2061 D.28, 1994]).

Traffic Class	*Peak Rate (Mbit/s)*	*Mean On period (s)*	*Mean Off Period (s)*
A.3.1	31.1	20	80
A.3.2	31.1	10	190
B.3.1	7.78	50	50
B.3.2	7.78	190	10
C.3.1	1.94	50	50
C.3.2	1.94	20	80

The CLR obtained via measurements for each of the traffic mixes considered in the experiment scenarios of table 6.5 were used to create a set of training examples (see chapter 5, section 5.2.1.1 how to construct the example set). FCAC and NFCAC were trained using this example set and no previous knowledge existed either for FCAC or NFCAC.

Table 6.5 Specification of the experiments for homogeneous and heterogeneous traffic mixes (learning approaches)

Experiment	*Buffer size (cells)*	*Link Capacity (Mbit/s)*	*Traffic type*	*Figure*
B.1	27	155.52	A.3.1	6.13
B.2	27	155.52	A.3.2	6.14
B.3	27	155.52	B.3.1	6.15
B.4	27	155.52	B.3.2	6.16
B.5	27	155.52	C.3.1	6.17
B.6	27	155.52	C.3.2	6.18
C.1	27	155.52	B.3.1 + 2A.3.1	6.19
C.2	27	155.52	B.3.1 + 4A.3.1	6.20
C.3	27	155.52	B.3.2 + 2A.3.1	6.21
C.4	27	155.52	B.32 + 4A.3.1	5.22
C.5	27	155.52	C.3.1 + 2A.3.1	5.23
C.6	27	155.52	C.3.1 + 4A.3.1	5.24
C.7	27	155.52	C.3.1 + 4B.3.1	5.25
C.8	27	155.52	C.3.1 + 8B.3.1	5.26

The rule base obtained for FCAC, using the automatic design method described in chapter 5, contains 101 rules. NFCAC uses a rule base with a maximum of 27 rules since 3 input fuzzy variables are considered and the fuzzy domain of each variable contains 3 fuzzy sets. The training example set contains 80 examples.

The experiments reported in this section aim to evaluate the generalisation capability of FCAC and NFCAC, in terms of the CLR prediction, to traffic mixes for which there is no previous knowledge other than the knowledge obtained from the set of training examples used. The traffic mixes used to test the predictions given by FCAC and NFCAC, are composed of the same type of sources as the sources used in the training examples, but considering a wider range of traffic patterns for the multiplexed traffic, obtained by varying the number of the sources for one of the traffic types in the traffic mix. The evaluation of the accuracy of the CLR prediction is not the main aim of the experiments reported in this section, but rather to evaluate, for the totality of the traffic mixes studied, how well (number of test cases for which an unexpected CLR is obtained, number of test cases for which no CLR prediction is given) do FCAC and NFCAC behave for the traffic mixes considered. Given that NFCAC predicts a CLR expressed as a floating point number whereas FCAC predicts a CLR expressed as an integer (representing the negative power of ten of the previously referred ratio) a strict comparison is not possible. Instead, an assumption is made that a maximum difference of 10^{-1} between the CLR predicted by FCAC and the CLR predicted by NFCAC is allowed. If the difference between the two CLR predictions is bigger than 10^{-1}, then the test case is analysed in more detail in order to determine the cause of divergence.

6.2.1 Homogeneous scenarios

The CLR predictions obtained using FCAC and NFCAC for the homogeneous traffic scenarios corresponding to experiments B.1 to B.6 are plotted in figures 6.13 to 6.18. The degree of certainty associated with the prediction given by FCAC is plotted in each of the graphics in the second *Y*-axis (right hand side).

The results of experiments B.1 to B.3 (plotted in the graphics of figures 6.13 to 6.15) show that the CLR predictions given by FCAC and NFCAC are, for the majority of the cases, within the expected value obtained when extrapolating from the result obtained via on-line measurements. Only two cases are irregular:

1. in figure 6.13, FCAC failed to provide a CLR prediction for traffic mixes with 11 A.3.1 sources;

2. in figure 6.15, for low loads (in particular the case of a traffic mix of 23 B.3.1 sources), the maximum allowed difference (10^{-1}) between the CLR predictions of FCAC and NFCAC was reached.

In the first case, the traffic sources have a very high peak to link ratio and, thus, adding one more source to the traffic mix will create a completely different traffic pattern. FCAC cannot generalise its CLR prediction for the new traffic pattern based on the existing knowledge. For a traffic mix with 14 A.3.1 sources, a certain oscillation of the CLR curve predicted by NFCAC can also be observed. For the second case, the CLR prediction given by FCAC is optimistic for low loads and this is because of lack of knowledge of such traffic patterns.

In figure 6.16, the CLR predicted by FCAC and NFCAC differ significantly for traffic mixes with a number of B.3.2 sources in the range of 211 to 214 sources. Furthermore, the CLR predicted by FCAC is optimistic for traffic mixes with a number of B.3.2 sources in the range of 201 to 214, when compared with the results obtained by measurements. A similar comment is given for figure 6.17, for traffic mixes with a number of C.3.1 sources in the range of 115 to 120, 123 to 125, and 133 to 134. Both of these figures show that the degree of certainty associated with the CLR predicted by FCAC is close to 0.5 or decreases to 0.5 (or even less) for the above mentioned traffic mixes. Thus, FCAC is making a CLR prediction with lack of knowledge on those traffic patterns and this explains the optimism in the predicted CLRs.

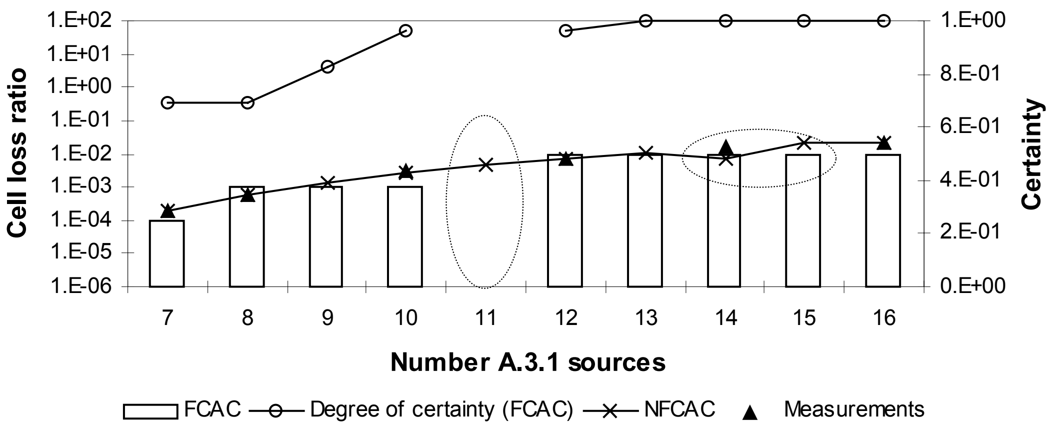

Figure 6.13 Comparison of CLR predictions for traffic mixes with A.3.1 traffic sources.

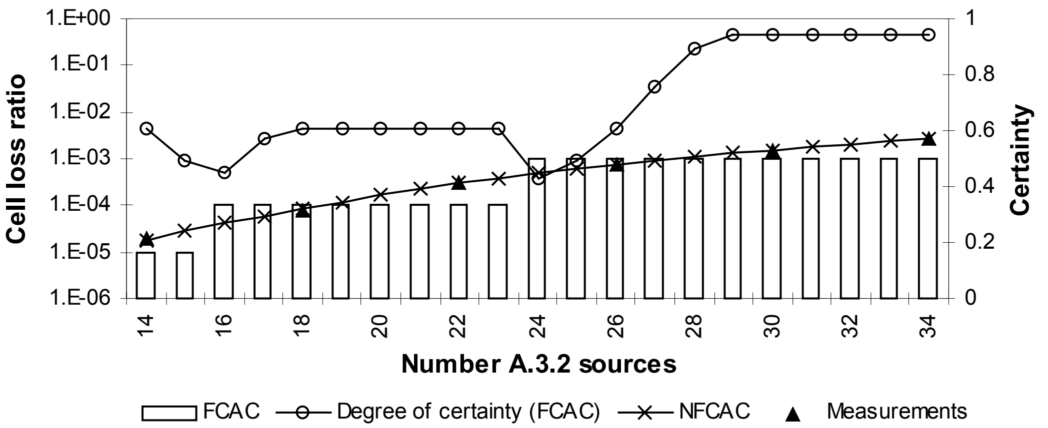

Figure 6.14 Comparison of CLR predictions for traffic mixes with A.3.2 traffic sources.

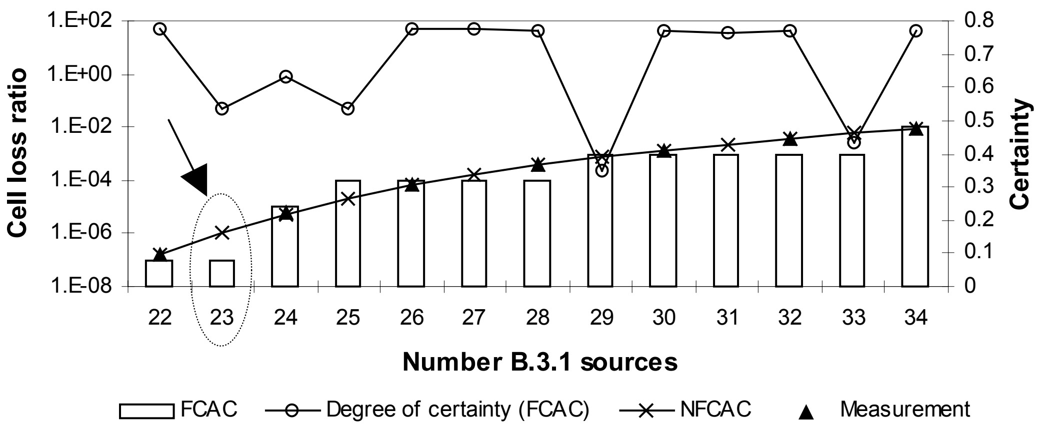

Figure 6.15 Comparison of CLR predictions for traffic mixes with B.3.1 traffic sources.

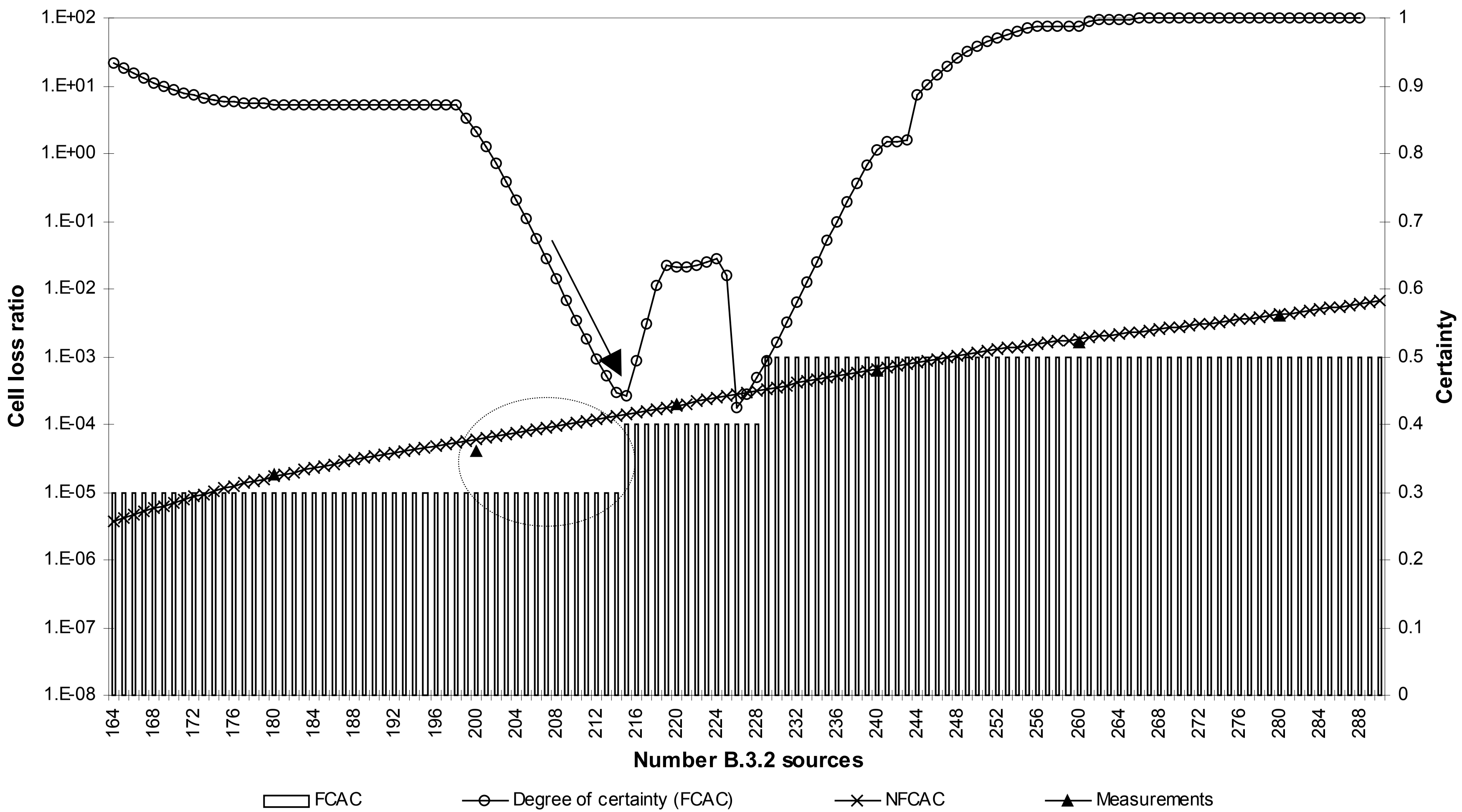

Figure 6.16 Comparison of CLR predictions for traffic mixes with B.3.2 traffic sources.

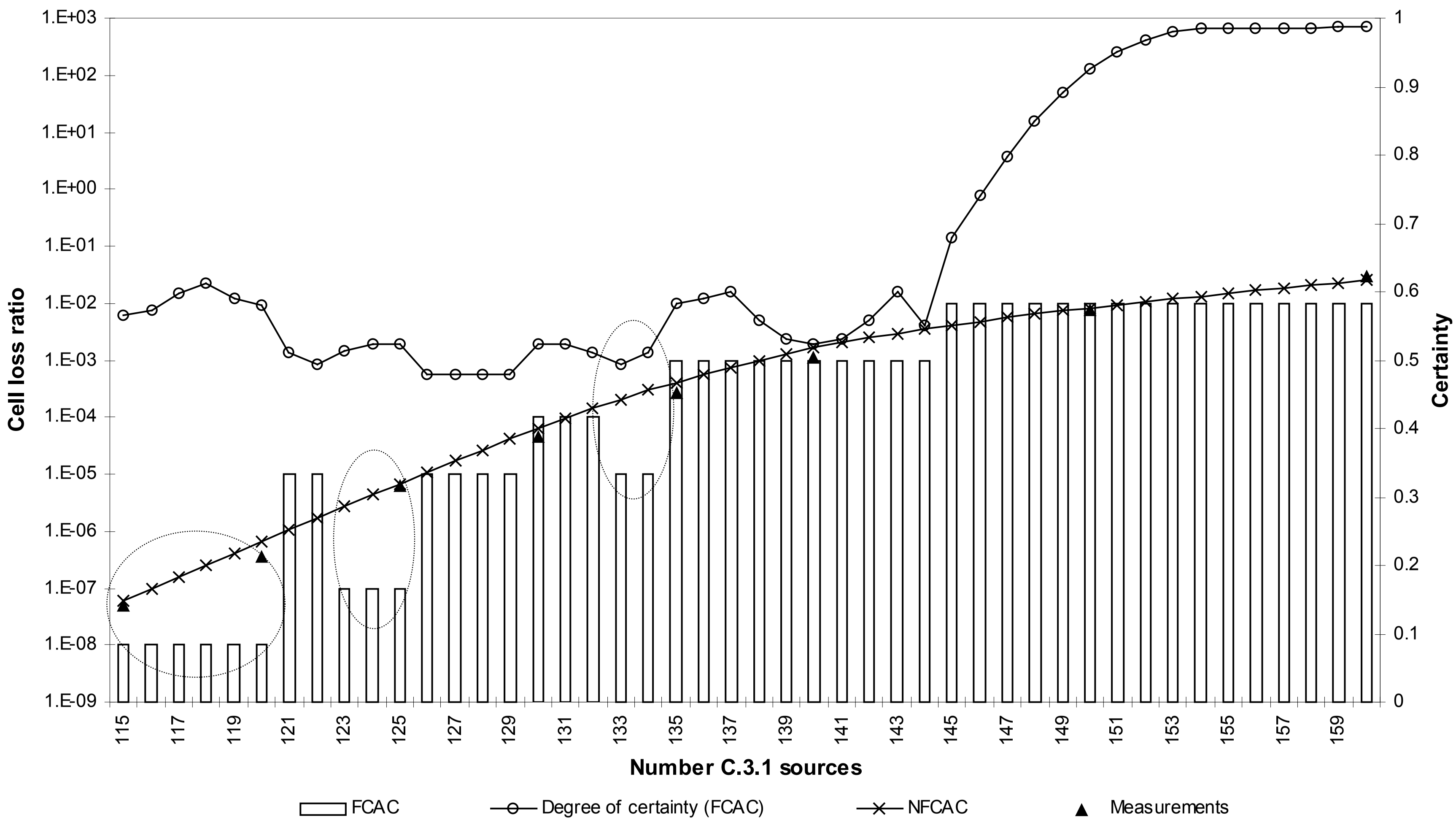

Figure 6.17 Comparison of CLR predictions for traffic mixes with C.3.1 traffic sources.

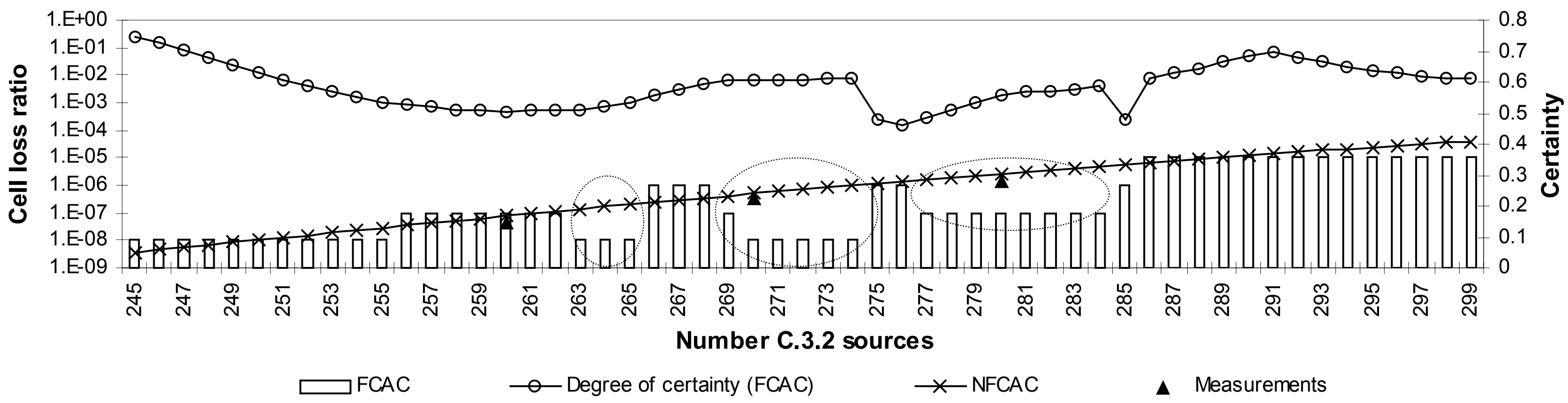

Figure 6.18a Comparison of CLR predictions for traffic mixes with low loads of C.3.2 traffic sources.

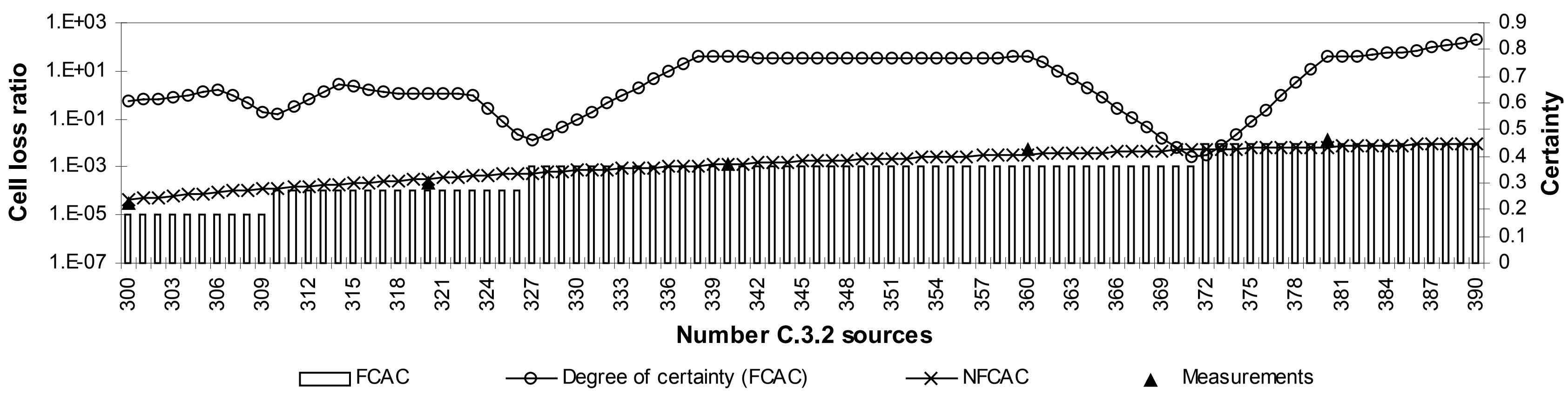

Figure 6.18b Comparison of CLR predictions for traffic mixes with high loads of C.3.2 traffic sources.

Figures 6.18a and 6.18b show the CLR predicted by FCAC and NFCAC for traffic mixes corresponding to low and high loads of C.3.2 sources, respectively. For high loads of C.3.2 sources, both the CLRs predicted by FCAC and NFCAC are within the expected value obtained when extrapolating from the result obtained via on-line measurements. For low loads, the CLR predicted by FCAC is optimistic for traffic mixes with a number of C.3.2 sources in the range of 263 to 265, 269 to 274 and 277 to 285. The degree of certainty associated with such CLR prediction is also between 0.5 and 0.6 revealing a small confidence in the accuracy of the predicted CLR value.

The CLR predicted by NFCAC is close to the CLR obtained via on-line measurements for experiments B.1. to B.6. The CLR prediction given by FCAC is closer to the CLR given by measurements for CLR predictions in the range of 10^{-2} to 10^{-4}. This is because the percentage of training examples corresponding to traffic mixes with a CLR in the previously mentioned range is much higher than the percentage of examples for traffic mixes with a CLR less than 10^{-4}.

6.2.2 Heterogeneous scenarios

The CLR predicted by FCAC and NFCAC for heterogeneous traffic mixes with two different types of sources (experiments C.1 to C.8) are plotted in figures 6.19 to 6.26.

In figure 6.19, FCAC failed to give a prediction for traffic mixes with 16, 20 and 21 B.3.1 sources in the mix. The difference between the CLR predicted by FCAC and the value predicted by NFCAC is over 10^{-1} for traffic mixes with 22, 23 and 24 B.3.1 sources. The degree of certainty associated with the CLR predicted by FCAC for these traffic mixes is always between 0.5 and 0.6, indicating that FCAC predicts with lack of knowledge.

The difference between the CLR predicted by FCAC and NFCAC is also over 10^{-1}, in figure 6.21, for traffic mixes with a number of B.3.2 sources in the range 90 to 112. In this case, inconsistencies in the fuzzy rule base (which could not be eliminated given the limited number of training examples in use) are the cause of discrepancies. Note also that the degree of certainty associated with the predictions is over 0.5 which shows that as far as the FCAC is concerned, the prediction is accurate.

In figure 6.20, the CLR predicted by FCAC and NFCAC is as expected comparing the predictions with the results obtained from on-line measurements.

FCAC failed to give a prediction for three traffic mixes in the traffic scenario of figure 6.20: traffic mixes composed of 10, 14 and 18 B.3.1 sources and a fixed number of A.3.1 sources (2 sources). The same is observed in figure 6.22, for traffic mixes with 35 to 45 B.3.2 sources and 4 A.3.1. sources in the mix. The prediction given by FCAC is optimistic in figure 6.22, for traffic mixes with 25 to 34 B.3.2 sources and 4 A.3.1 sources in the traffic mix. All these results clearly stem from a lack of traffic knowledge about these patterns, as shown by the fact that the degree of certainty decreases as the number of B.3.2 sources added to the traffic mixes increases.

In figure 6.23, the CLR predicted by FCAC is not as expected for traffic mixes with 114 to 130 C.3.1 sources (2 A.3.1 sources in the mix). The predicted CLR suddenly decreases from 10^{-3} to 10^{-4} and remains at that level even if the number of C.3.1 sources in the mix increases (the number of A.3.1 sources is fixed for this experiment). The CLR value predicted by NFCAC for traffic mixes with a number of C.3.1 sources over 125 is also unexpected, since it decreases instead of increasing. The CLR predicted by FCAC behaves as expected for high loads, though.

C.3.1 sources have a very small peak to link ratio (see table 6.4). A.3.1 sources have a very high peak to link ratio. Thus, in order to achieve an accurate CLR prediction for the experiment shown in figure 6.23, both FCAC and NFCAC require more training examples that include these type of traffic mixes. For the experiment shown in figure

6.23, only four of the training examples contain information relative to traffic mixes with C.3.1 and A.3.1 sources. These are the examples for traffic mixes with 70, 80, 100 and 120 C.3.1 sources (the number of A.3.1 sources, 2, is fixed for all the traffic mixes).

In figure 6.24 the CLR predictions given by FCAC and NFCAC are as expected, except for very low loads in which case the CLR predicted by FCAC is in the order of 10^{-7} and the CLR predicted by NFCAC is in the order of 10^{-9}. The set of training examples does not contain examples for traffic scenarios characterised by very low cell loss ratios. This is the reason why FCAC takes a more conservative approach.

In figure 6.25, the CLR predicted by FCAC is optimistic for traffic mixes with 114 to 118 C.3.1 sources in the mix, when compared to the CLR prediction given by NFCAC. Considering that only two of the training examples describe traffic patterns with 4 B.3.1 sources and a variable number of C.3.1 sources, the predictions given by FCAC and NFCAC generalise the traffic knowledge obtained from similar traffic patterns contained in the example set, in order to predict the CLR for this experiment. B.3.1 and C.3.1 traffic sources have the same mean on and off periods (different peak bit rates). B.3.1 has the highest peak bit rate but the number of C.3.1 sources is always much higher than the number of B.3.1 sources. Therefore, FCAC and NFCAC interpret this traffic pattern as the pattern for an homogeneous traffic scenario and match it with one of the homogeneous traffic patterns for which there is traffic knowledge.

In figure 6.26, the homogeneity of the traffic pattern is broken by adding 4 more B.3.1 sources to the traffic mix (a total of 8 B.3.1 sources is considered). NFCAC can still provide an accurate generalisation but FCAC requires more training examples in order to be able to generalise for low loads. The CLR predicted by FCAC is too pessimistic for traffic mixes with 85, 86 and 92 to 97 C.3.1 sources in the traffic mix. FCAC also fails to give a CLR prediction for traffic mixes with 87 to 91 C.3.1 sources.

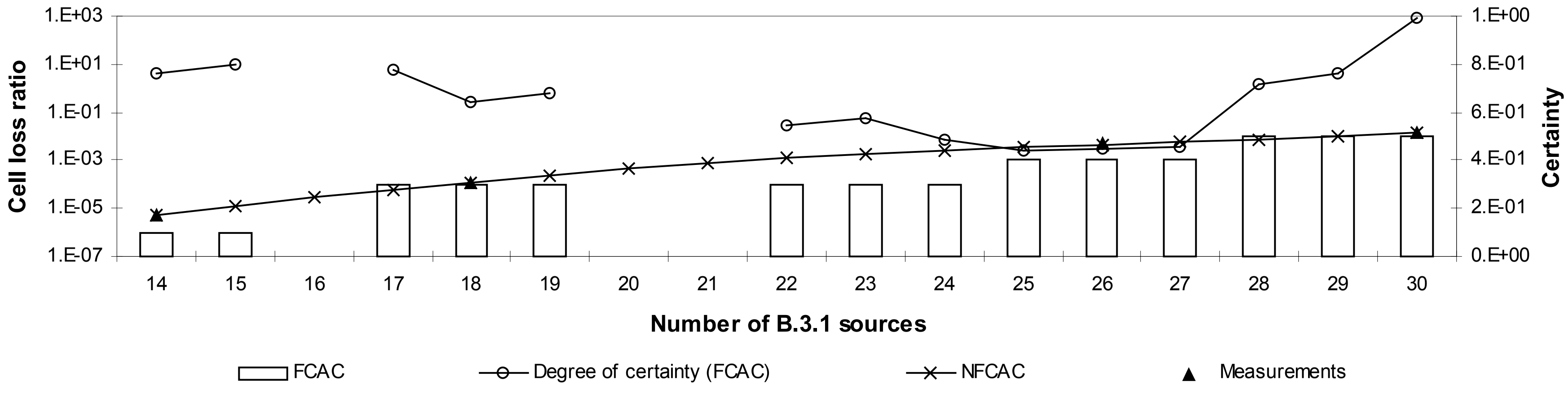

Figure 6.19 Comparison of CLR predictions for traffic mixes with 2 A.3.1 sources and a variable number of B.3.1 sources.

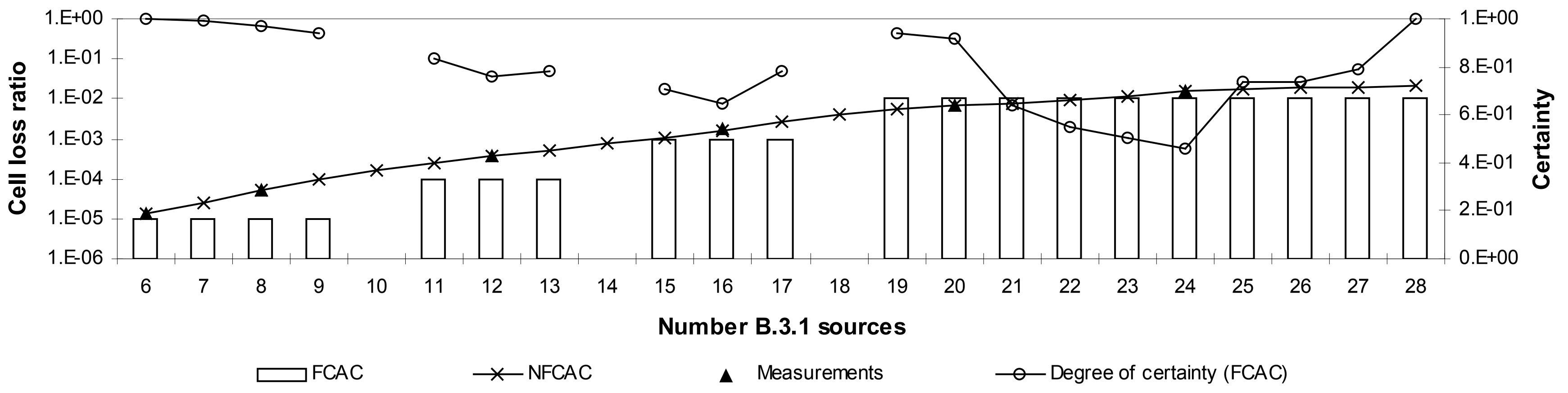

Figure 6.20 Comparison of CLR predictions for traffic mixes with 4 A.3.1 sources and a variable number of B.3.1 sources.

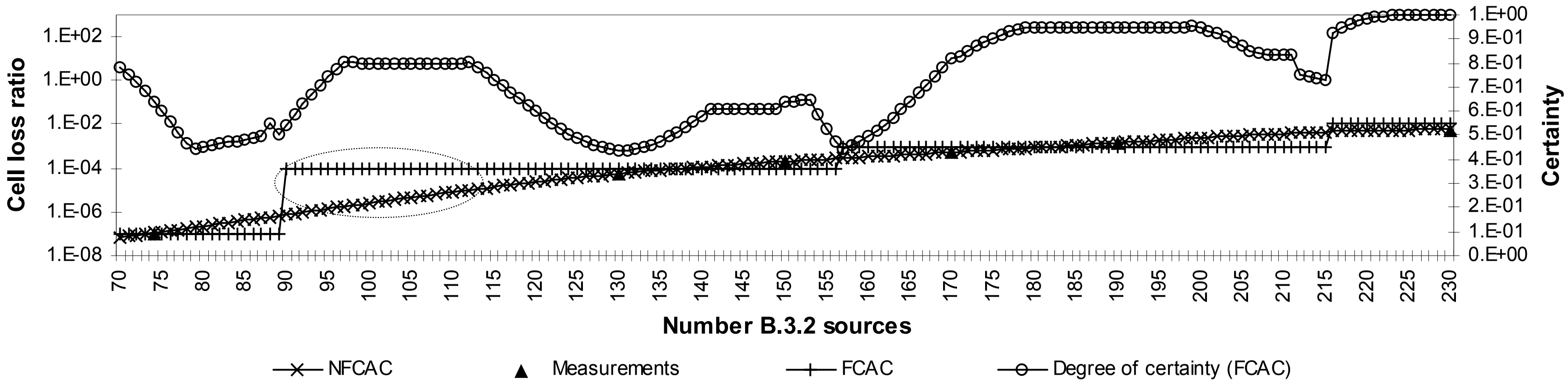

Figure 6.21 Comparison of CLR predictions for traffic mixes with 2 A.3.1 sources and a variable number of B.3.2 sources.

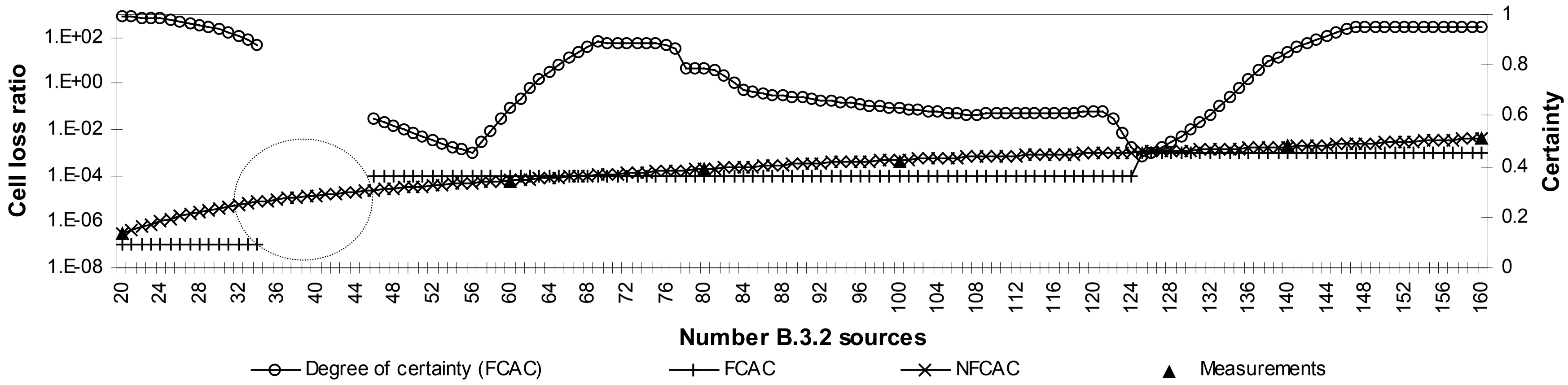

Figure 6.22 Comparison of CLR predictions for traffic mixes with 4 A.3.1 sources and a variable number of B.3.2 sources.

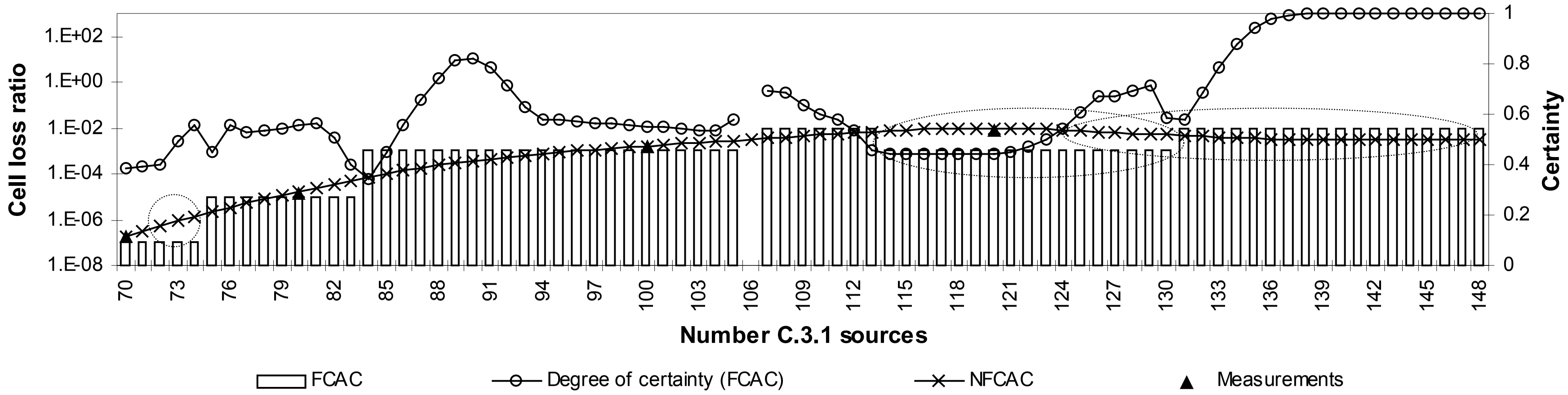

Figure 6.23 Comparison of CLR predictions for traffic mixes with 2 A.3.1 sources and a variable number of C.31 sources.

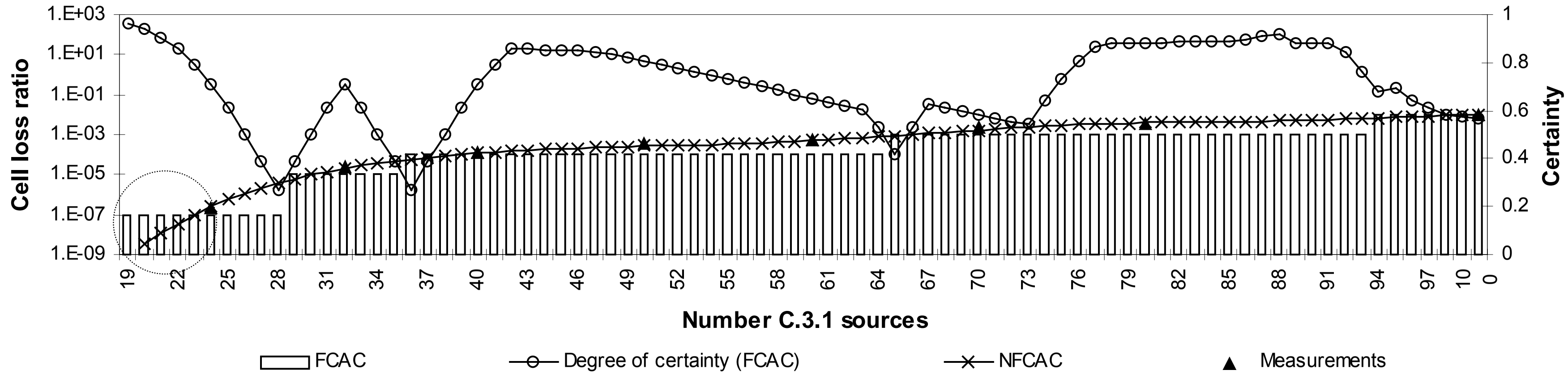

Figure 6.24 Comparison of CLR predictions for traffic mixes with 4 A.3.1 sources and a variable number of C.31 sources.

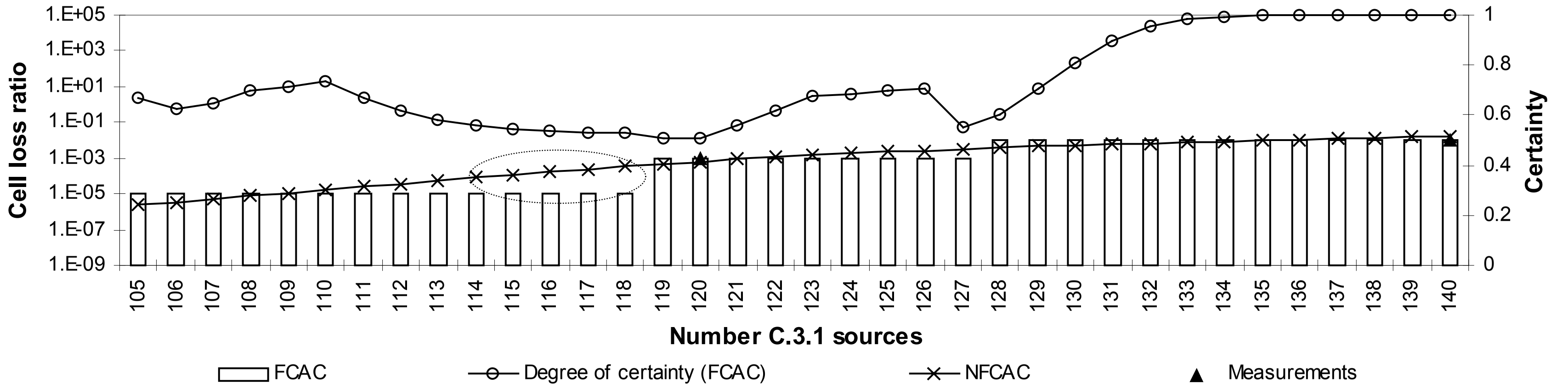

Figure 6.25 Comparison of CLR predictions for traffic mixes with 4 B.3.1 sources and a variable number of C.31 sources.

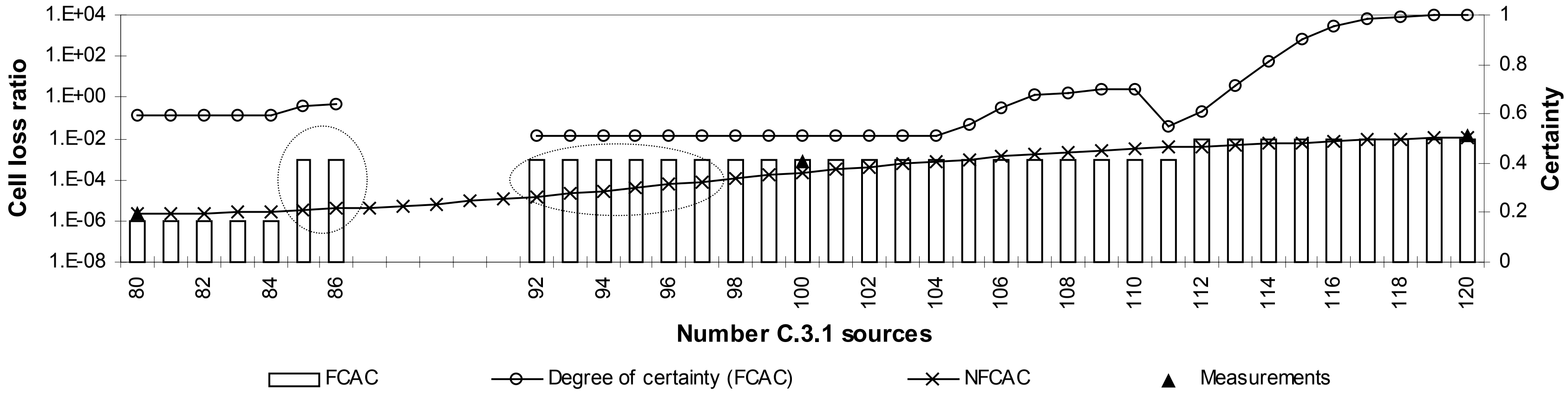

Figure 6.26 Comparison of CLR predictions for traffic mixes with 8 B.3.1 sources and a variable number of C.31 sources.

The source for these patterns [R2061 D.28, 1994] does not comment on the reason for the choice of the characteristics of the traffic sources shown in table 6.4 so, no mapping can be made between the results obtained in the experiments and real traffic situations. Nevertheless, from experiments B.1 to B.6 and C.1 to C.8 presented in this section, it can be concluded that:

- for traffic mixes such as the ones shown in figure 6.23, both FCAC and NFCAC require more training examples than the ones used in the experiment of this section in order to generalise the CLR prediction to traffic patterns with a different number of any of the traffic sources in the mix considered,
- for other traffic mixes such as the ones shown in figure 6.25, both approaches can easily generalise its CLR prediction even with a very reduced number of training examples.

The prediction given by FCAC is generally more accurate (compared with the values obtained by measurements) for traffic scenarios characterised by high load. This is because the percentage of training examples describing such traffic scenarios is higher than the number of training examples describing traffic scenarios with low load. The reason for the reduced number of training examples relative to traffic scenarios with low load is that such scenarios are associated with low CLRs and these ratios take a long time to monitor using on-line measurements (a very high number of cells needs to be generated in order to achieve statistical significance). Overall, the prediction given by NFCAC is consistent with the expected CLR when extrapolating from the results obtained via on-line measurements.

The switch used to perform the on-line measurements in the ATM test-bed in Basel has a very small output buffer size and, therefore, does not suit a CAC approach (such as FCAC) that takes into account the reduction of the CLR obtained by queueing part of the excess traffic. On the other hand, FCAC requires more knowledge than NFCAC to provide the same CLR prediction accuracy since the fuzzy system used has 4 input variables whereas NFCAC uses a fuzzy system with 3 input variables.

6.3 Experiments on adaptation of FCAC

The experiments reported in the following section were possible thanks to RACE project EXPLOIT having made available the equipment for traffic generation and analysis at the ATM test-bed in Basel for a week of experiments (experiments 1 to 9). The rest of the experiments reported have been obtained via simulations using LINKSIM.

The network configuration is the same as the one used in the experiments of section 6.2, that is, C=155.52 Mbit/s and K=27 cells. The sources used are VBR sources, modelled as On-Off sources with parameters: peak bit rate, mean "on" and "off" periods. The traffic mixes studied are composed of 6 different type of sources (heterogeneous mixes); see also table 6.6 (background traffic) and table 6.7 (foreground traffic).

Table 6.6 Traffic characteristics of On-Off sources (background traffic).

Traffic Class	*Peak Rate* (*Mbit/s*)	*Mean On period* (*s*)	*Mean Off Period* (*s*)
NTUA-PC1	37	0.000836	0.00336
NTUA-PC2	37	0.00126	0.00293
NTUA-PC3	37	0.00167	0.006715
NTUA-PC4	37	0.00252	0.005878

The experiments are performed using the RUM switch (made by AT&T) to multiplex the ATM traffic and the following traffic generators and analysers are used:

- PC-based ATM traffic generator developed by the National Technical University of Athens (NTUA- PC);
- UNIX-based ATM traffic generator and analyser, ATM-100, developed by Wandel & Goltermann;
- PC-based Alcatel A8643 traffic analyser and generator.

The experiment starts by switching on the 5 NTUA-PCs and the ATM100 as traffic generators. The A8643 is the traffic analyser for the traffic from the NTUA-PCs and the ATM-100 also performs the function of traffic analyser for the traffic that it generates.

The traffic is multiplexed using the RUM switch and switched via port 4 of the RUM. After a while, when the multiplexed traffic has reached a steady state, the ATM-100 analyser is switched on, and the analysis of the traffic generated by the ATM-100 starts. After receiving at the A8643 a pre-fixed number of cells for the 5 NTUA-PCs, both analysers (ATM-100 and A8643) are switched off and the experiment is concluded. The traffic analysis will allow the CLR for each of the connections to be obtained and from these ratios, the maximum CLR per connection to be determined.

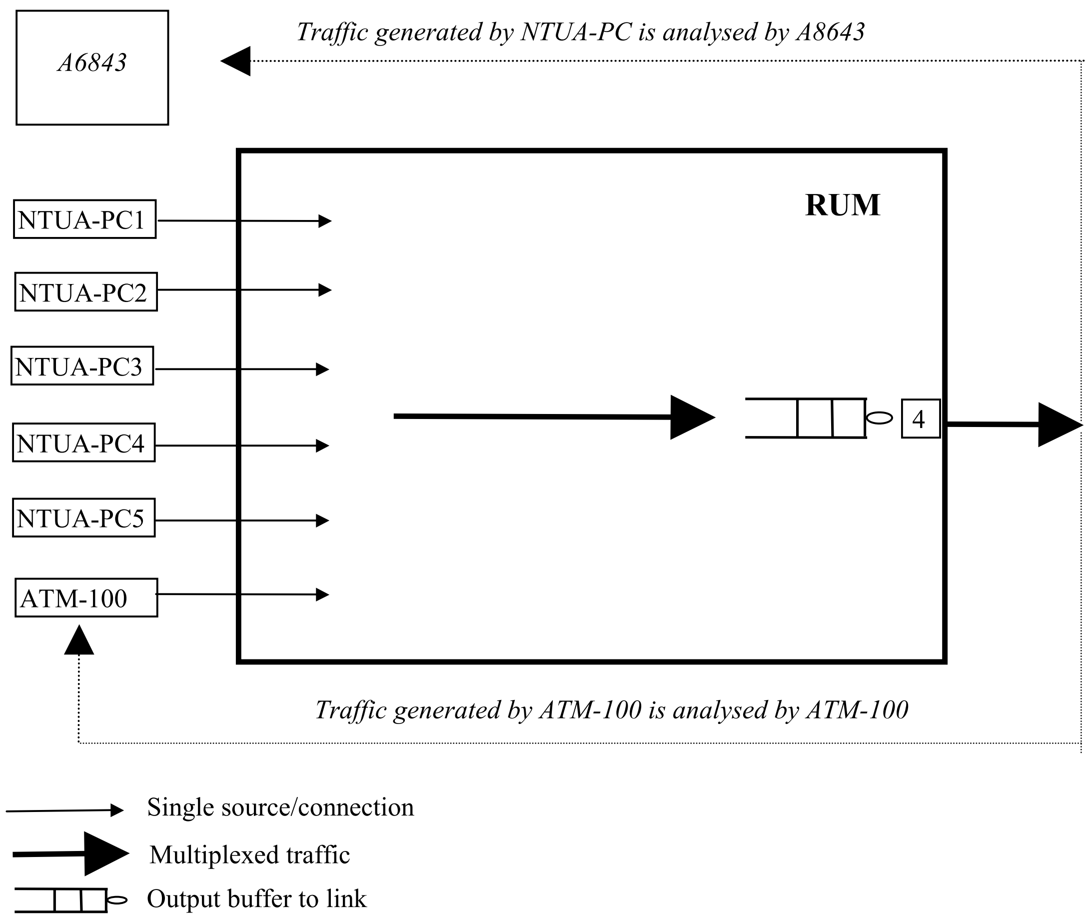

Figure 6.27 Description of the configuration for the experiments

Table 6.7 Foreground traffic characteristics (experiments D.1 to D.9)

	NTUA-PC5			ATM-100		
Experiment	Peak Rate (Mbit/s)	Mean On Period (s)	Mean Off Period (s)	Peak Rate (Mbit/s)	Mean On Period (s)	Mean Off Period (s)
D.1	37	0.007975	0.007975	8	0.02	0.08
D.2	37	0.007975	0.007975	8	0.04	0.08
D.3	37	0.007975	0.009568	8	0.02	0.08
D.4	37	0.007975	0.009568	8	0.04	0.08
D.5	37	0.007975	0.007975	7	0.04	0.08
D.6	37	0.007975	0.007975	6	0.04	0.08
D.7	37	0.007975	0.007975	8	0.02	0.04
D.8	37	0.007975	0.007975	8	0.06	0.12
D.9	37	0.007975	0.007975	8	0.01	0.02

The experiments performed on this section consist of multiplexing background traffic (table 6.6 presents the characteristics of the background traffic sources) with foreground traffic (table 6.7 presents the characteristics of the foreground traffic sources). The same background traffic is used throughout all the experiments presented in this section. The characteristics of the sources in the foreground traffic change according to table 6.7.

Tables 6.8, 6.9 and 6.10 allow a better understanding of the layout of the experiments performed in this section. Table 6.8 shows the differences between experiments D.1 to D.4 regarding the characteristics of the foreground traffic sources: the peak bit rate of the sources remains the same for the four experiments (8 Mbit/s for the ATM-100 and 37 Mbit/s for the NTUA-PC5) and only the mean on and off periods of one of the two foreground sources (ATM-100 or NTUA-PC5) will vary. The CLR results obtained for each of the experiments D.1, D.3 and D.4 will be compared with the CLR obtained for the reference experiment (experiment D.2).

Table 6.8 Layout of experiments D.1 to D.4

NTUA-PC5 vs ATM-100 (mean_on, mean_off) in sec.	ATM-100 (0.02, 0.08)	ATM-100 (0.04, 0.08)
NTUA-PC5 (0.0079, 0.0079)	Experiment D.1	Experiment D.2 (reference)
NTUA-PC5 (0.0079, 0.009568)	Experiment D.3	Experiment D.4

Table 6.9 Layout of experiments D.2, D.5 and D.6 for ATM-100 (same mean "on" and "off" periods)

Experiment	ATM-100 (Peak bit rate in Mbit/s)
D.2 (reference)	8
D.5	7
D.6	6

Table 6.9 presents experiments D.2, D.5 and D.6 as the experiments designed to test the influence of varying the peak bit rate of one of the foreground traffic sources (the ATM-100) on the CLR obtained per connection. Table 6.10 presents the experiments designed to test the influence of changing the mean on and off periods of the ATM-100 (the percentage of time "on" and the peak bit rate of the ATM-100 are constant for the four experiments) on the obtained maximum CLR per connection.

Table 6.10 Layout of experiments D.2, D.7, D.8 and D.6 for ATM-100 (same mean bit rate)

Experiment	ATM-100	
	Mean On Period (s)	Mean Off Period (s)
D.2 (reference)	0.04	0.08
D.7	0.02	0.04
D.8	0.06	0.012
D.9	0.01	0.02

Other traffic experiments were performed in order to study the influence of adding one more ATM-100 connection (with the same traffic characteristics as the ATM-100 connection specified for the corresponding experiment) on the maximum CLR to be expected for the background and foreground connections, for each of experiments D.1 to D.9.

Limitations of time and the number of connections that can be analysed by the traffic analysers at the ATM test-bed, made it impossible to make all of these measurements on-line and the cell rate simulator, LINKSIM (see [Pitts, 1993]) was used instead. Therefore, the CLR shown in figures 6.28 to 6.36 is obtained:

- via on-line measurements, for traffic mixes with only one ATM100 source in the mix;
- via simulation, for traffic mixes with more than one ATM100 traffic source in the mix.

The experiments described in table 6.7 have been performed in order to test the performance of the tuning algorithm in the presence of new test examples. The CLR predicted by FCAC was obtained after tuning the rule base obtained for the experiments in section 6.2 with a set of testing examples containing the on-line measured CLRs and also the CLRs obtained via simulation (see in section 5.2.1.1 how to construct the examples from the data). Some of the CLR values obtained via simulation were used to construct the test example set, the remaining values were used to evaluate the generalisation ability of FCAC when applied to traffic scenarios for which there was no previous knowledge. The symbol, ♣, is placed adjacent to the bar corresponding to the FCAC prediction in figures 6.33 to 6.36 for the traffic scenarios for which there is no previous knowledge.

Figures 6.28 to 6.36 show the CLR predicted by FCAC and the CLR obtained by on-line measurements (traffic mixes containing only one ATM-100 source) and simulations (traffic mixes containing more than one ATM-100 source). From the figures, it can be observed that FCAC is able to detect, for the majority of the cases, when the CLR changes from 10^{-4} to 10^{-3} with the increasing number of ATM-100 sources added to the background traffic. Only two cases show irregularities:

- in figures 6.35, FCAC failed to give a CLR prediction for a traffic scenario with 2 ATM100 sources in the traffic mix. The set of test examples does not contain data for this traffic pattern (note the ♣ symbol in the figure). The characteristics of the ATM-100 source for experiment D.8 are such that the mean on and off periods are the highest of all the experiments. Thus, FCAC cannot match the characteristics of this traffic mix with any of the previously studied.
- in figure 6.33, the CLR predicted by FCAC for the traffic mix with 2 ATM-100 sources (experiment D.6) is not as expected. Because the CLR obtained via simulation is less than $5x10^{-3}$, the CLR predicted by FCAC should have been 10^{-4} and not 10^{-3}. Again for this experiment the set of test examples does not contain knowledge for traffic mixes with 3 and 4 ATM-100 traffic sources. In experiment D.6, the peak bit rate of the ATM-100 is the lowest of the peak bit rates of the ATM-100 source for of all the experiments. Considering the reduced number of test examples for this traffic scenario, FCAC cannot make too much sense of the specific characteristics of this traffic mix and predicts a conservative CLR value.

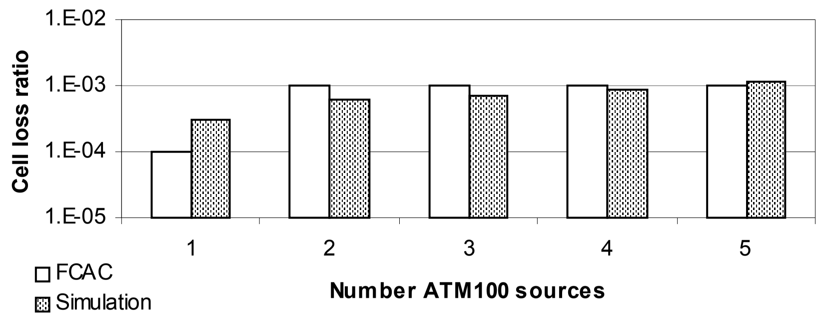

Figure 6.28. CLR for background traffic described in experiment D.1.

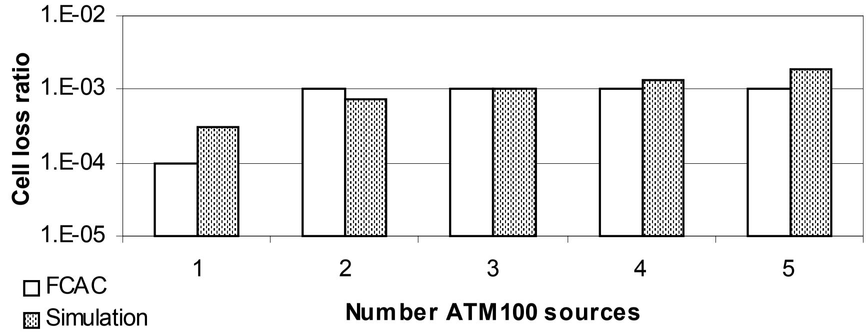

Figure 6.29. CLR for background traffic described in experiment D.2.

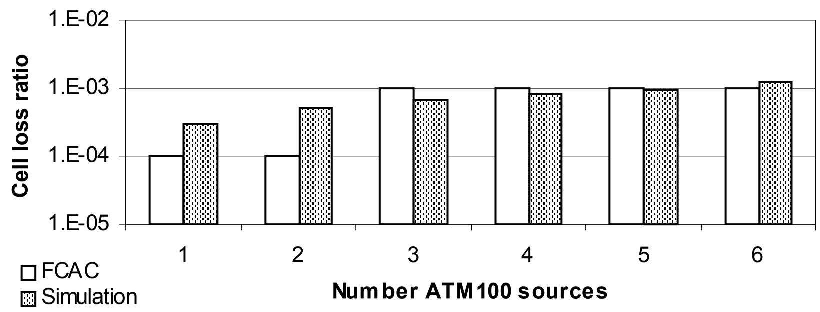

Figure 6.30. CLR for background traffic described in experiment D.3.

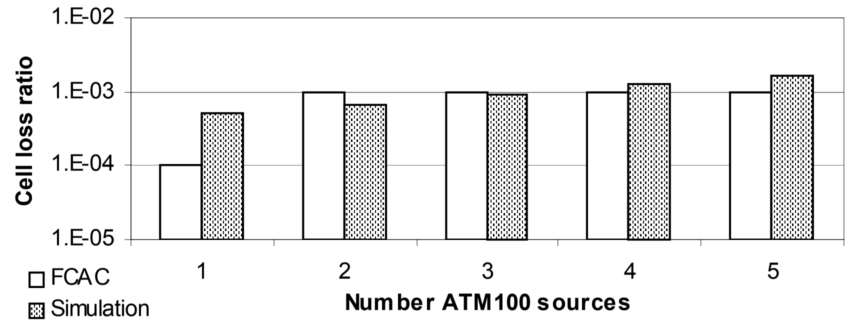

Figure 6.31 CLR for background traffic described in experiment D.4.

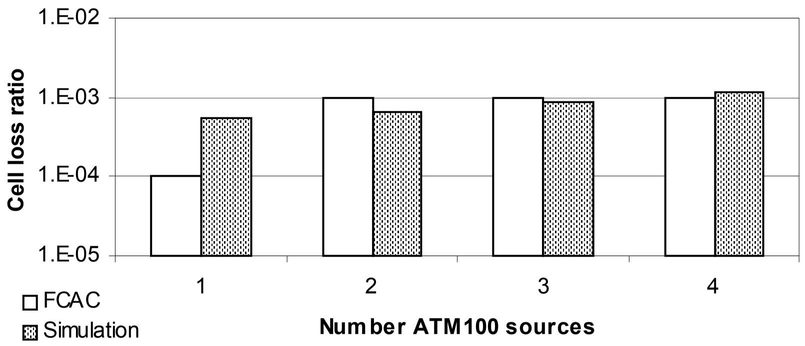

Figure 6.32 CLR for background traffic described in experiment D.5.

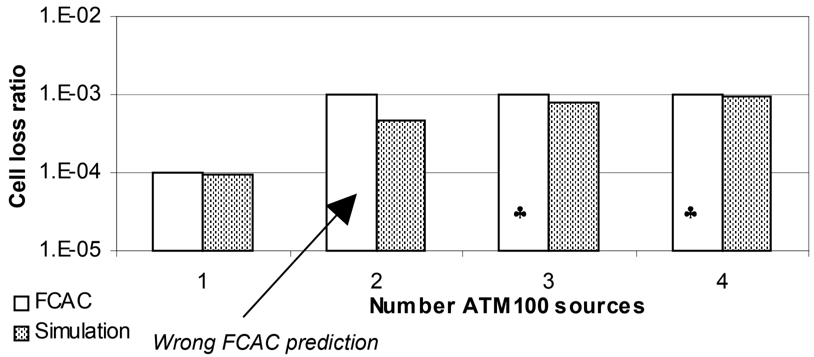

Figure 6.33 CLR for background traffic described in experiment D.6.

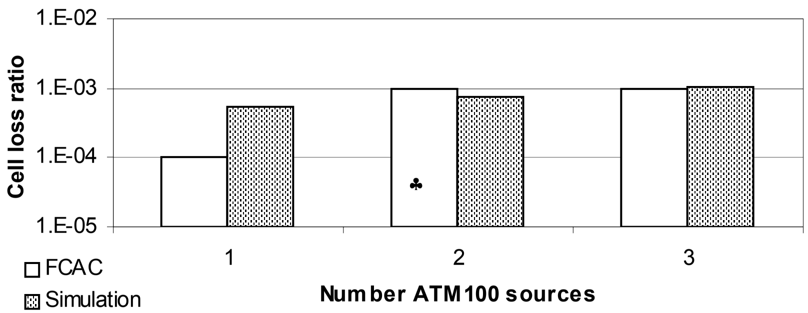

Figure 6.34 CLR for background traffic described in experiment D.7.

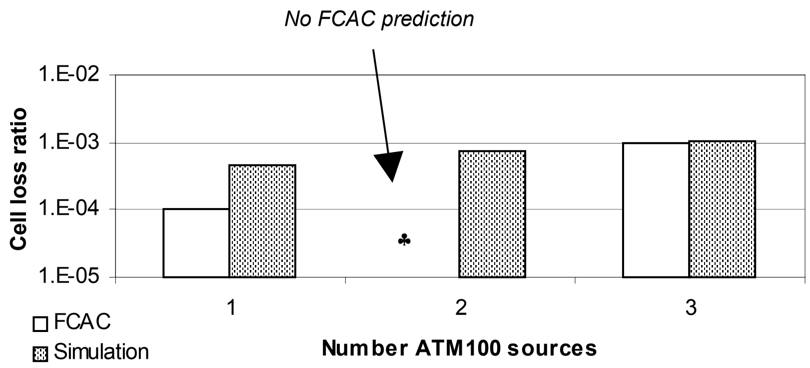

Figure 6.35 CLR for background traffic described in experiment D.8.

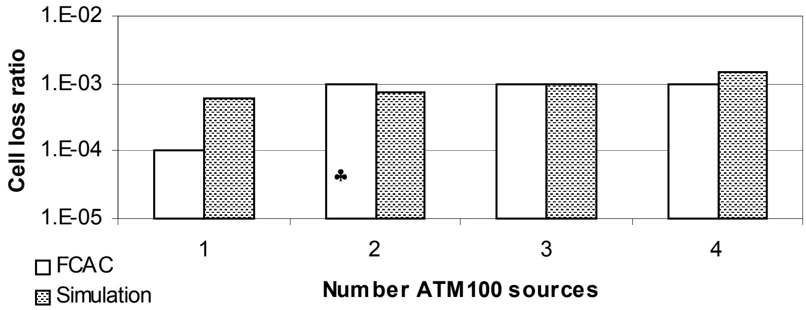

Figure 6.36 CLR for background traffic described in experiment D.9.

For the totality of the 39 traffic scenarios studied, FCAC had knowledge of 34 traffic scenarios. The prediction given by FCAC was not as expected for 2 of the studied traffic scenarios, comparing it with the value obtained either via on-line measurements or via simulation. The traffic scenarios for which FCAC failed to provide the expected CLR are scenarios for which there is no information in the set of test examples and, therefore, a reduced ability to generalise from previous knowledge.

For the experiments described in table 6.9, considering experiment D.2 as the reference experiment, the influence on the CLR of reducing the peak bit rate while keeping constant the mean "on" and "off" periods of the ATM-100 traffic source has been studied. The CLR increases more rapidly when adding ATM-100 sources to the traffic mix in the case of experiment D.2 (highest peak bit rate) than it does for the other two experiments (D.5 and D.6). This is so much so that, in the case of experiment D.6 (lowest peak bit rate), only two ATM-100 sources are allowed in the traffic mix for a cell loss requirement of 10^{-4}.

For the experiments described in table 6.10, considering experiment D.2 as the reference experiment, no influence has been observed on the maximum CLR of each of the multiplexed connections when varying the mean on and off periods of the ATM-100 source (the mean bit rate is kept constant).

The mean on and off periods of the ATM-100 source are similar for experiments D.7 and D.9 and this explains the similarity between the CLR obtained in figures 6.34 and 6.36, respectively. FCAC does not make any distinction for the CLR curves of these traffic patterns either.

7 Discussion of the results

Traffic control in ATM networks requires the development of traffic models (for source and queueing) that are able to predict the influence on the cell loss and cell delay of each of the multiplexed connections when traffic with different characteristics (e.g. data and voice) and different QoS requirements is multiplexed on the same ATM link. It is important to be able to cope with mixtures of delay sensitive and loss sensitive traffic.

Traffic models are the basis for the development of most CAC methods. CAC has to decide upon the admission or rejection of a new connection request, making sure that there are available network resources in order to be able to satisfy the QoS requirements of all admitted connections. Provision must be made for:

- bandwidth, if the multiplexed traffic is characterised by long range rate variations that cannot be queued in ATM nodes (e.g. video traffic);
- queueing space, if the traffic exhibits long bursts (e.g. file transfer).

This thesis proposes a new fuzzy logic based decision support system for CAC in ATM networks: FCAC. Fuzzy logic is used to model the relationship between the traffic characteristics of a multiplexed set of connections and the maximum cell loss ratio to be obtained for each of the connections at a steady state, that is, when cell scale influences (caused by the asynchronous arrival of cells from different connections) are no longer predominant and only the influences of variations in rates and arrival of bursts need to be considered.

The FCAC predicts the maximum cell loss ratio to be obtained for each of the multiplexed connections when the new connection is to be multiplexed with existing background traffic; if the predicted cell loss ratio violates the cell loss ratio requirements of any of the connections then the new connection is rejected, otherwise the new connection is admitted.

Fuzzy sets such as "high", "medium" or "low" allow the quantification of vague statements about the traffic characteristics that often come to mind to a traffic engineer when reasoning about traffic control, such as for example the fuzzy "rule-of-the-thumb":

> "if the mean load of the link is *high*, the degree of determinism[1] in the bit rate pattern is *low*, the peak rate of the multiplexed connections is *medium* and the connections are active for *long* periods, then the maximum cell loss ratio for the connections is *high*".

The use of fuzzy rules, such as the one above, to represent the knowledge about ATM traffic, constitutes a traffic model that is simple and more flexible than traffic models based on the statistical analysis of the traffic patterns. The "if-then" statement expressed in the fuzzy rule, combined with the use of fuzzy sets to describe each of the variables allows for adaptability to new traffic patterns by adjusting the values of the parameters that define each fuzzy set. Although the use of vague descriptions (such as "high", "medium" included in the "rule-of-the-thumb" above), might appear not to provide a precise quantification of the fuzzy variables, in fact the matching between a crisp (single-valued) input for a specific input variable and a particular fuzzy set is precisely quantified using the notion of "degree of membership" of the crisp input in that fuzzy set. Furthermore, the use of fuzzy sets makes it possible to cope with a certain degree of imprecision (noise) in numerical values obtained from on-line measurements. This is quite useful for parameters, such as the peak cell rate, that are declared subject to a certain margin of error due to cell delay variation (CDV).

The FCAC structures data obtained from on-line measurements on an ATM link in a framework composed of fuzzy sets and fuzzy rules that constitutes the traffic model itself. Given this, two further issues related to the use of a learning CAC approach as opposed to an analytical CAC approach are:

[1] Traffic behaviour close to CBR, for which the mean rate is close to the peak rate.

1. FCAC relies on a sampling system able to monitor the changes in traffic patterns and the resulting cell loss ratio, because the knowledge about ATM traffic is obtained from training data,.
2. FCAC can easily describe the effects on cell loss, not only for the traffic patterns known at present but also for traffic patterns resulting from the integration of services introduced in the future (adaptation ability). This is because FCAC is not based on an analytical source and queue model to calculate cell loss probabilities, but the fuzzy rule base constitutes the traffic model itself.

Regarding the first point, further studies are required in order to study how the selection of the examples used to train FCAC is going to be processed. Because of the limited number of examples available for this research, it was not possible to study this topic any further. However, some interesting points to be investigated further could be, for instance:

- Learning and tuning - Should the training examples contain *only* examples referring to (1) traffic scenarios for which FCAC did not provide an accurate CLR prediction and (2) traffic scenarios for which FCAC failed to provide a CLR prediction?
- Adaptation - Should the training examples *also* refer to traffic patterns that are representative of typical traffic scenarios observed for a certain time period (e.g. day, week)?

Another relevant issue related to the performance of FCAC is the choice of the time during which to measure the maximum cell loss ratio for each of the connections multiplexed on the same ATM link. Some suggestions on how long the measurements should last for are:

- the whole duration of the connection;
- for steady state periods in which no connections are initialised or concluded;
- a period of time depending on the type of service associated with the connection, that is, depending on whether it is an interactive or data transfer type of service.

In order to evaluate the implications of the time variable (associated with the time during which the cell loss ratio is monitored) on the performance of the FCAC, in terms of 1) the accuracy of the CLR prediction and 2) in terms of the multiplexing gain, further experiments need to be performed on a real ATM traffic environment.

Learning from examples has been chosen as the method for acquiring knowledge about the relationship between multiplexing factors (such as the mean to peak ratio, average load, peak to link ratio, burst length to buffer size ratio) and the maximum cell loss ratio obtained for each connection. In a traffic control context, the examples refer to pairs of numeric values for the input and output fuzzy variables (in the antecedent and consequent of the fuzzy rules, respectively) that can be obtained from on-line traffic measurements.

A method for assigning a measure of the upper and lower bounds of uncertainty (interval of uncertainty) to each of the rules has been described in section 3.2. This method is based on the method of identifying fuzzy logic systems from examples, proposed by Delgado and Gonzalez ([Delg, 1993]). The uncertainty measure enables an evaluation of the truthfulness of a rule given a set of examples or, in other words, how well the rule describes a set of examples. In order to infer from fuzzy rules with an associated interval of uncertainty, a fuzzy inference algorithm derived from the work by Campos et al ([Camp, 1993]]) has been proposed in chapter 3, section 3.3. The inference algorithm allows the contribution of each of the rules to the inferred output to be weighted, not only by the degree of matching between the input values and the antecedent of the rules but also by the uncertainty bounds associated with the rule.

The rule base of the fuzzy logic CAC proposed in this thesis, FCAC, is automatically designed given a set of examples, the number of fuzzy variables in the antecedent and consequent of the rules and the universe of the fuzzy sets for each of the fuzzy variables. The fuzzy rules and the definition of the fuzzy sets for each of these rules are obtained using an algorithm which is based on genetic algorithms and consists of three phases:

1. Generation: genetic algorithms are used to search the space of possible solutions for the set of fuzzy rules that "best" represents a set of training examples (*generation* of

the rules). The evaluation of the "best" set of rules is done using the notions of positive and negative examples derived from the method of identification of fuzzy logic systems from examples presented in chapter 3, section 3.2;

2. Simplification: a genetic algorithm is used to search the space of possible solutions for the subset of fuzzy rules among the generated rules that is still able to provide a cell loss ratio prediction, for the input values stated in the set of training examples, with an error less than a pre-specified threshold (*simplification* of the rule set). The mean square error is used to measure the difference between the output obtained through fuzzy inference for each of the inputs in the set of training examples and the corresponding output value stated in the examples.

3. Tuning: a genetic algorithm is used to search the space of possible solutions for the values of the parameters describing the fuzzy sets in the antecedent of the rules that enable FCAC to provide an accurate cell loss ratio prediction for a new set of testing examples (*tuning* of the simplified rule set). The mean square error is also used to measure the performance of FCAC for the new test set.

The tuning algorithm is an optimisation algorithm that searches for the "best" set of values of the parameters that define the fuzzy sets for each of the rules. The set of possible values that such parameters can assume are elements of a bounded interval; the selection of the "best" set of parameters is dictated by the accuracy of the cell loss ratio predicted by a fuzzy system (FCAC) defined by those parameters for a set of testing examples. Given this, the implementation of the tuning algorithm could have been based on another evolutionary algorithm such as, for example, evolution strategies (see [Schw, 1977], [Schw, 1981]). Due to limitations of time, this research could not compare the performance of tuning algorithms based on different evolutionary algorithms. This is, nevertheless, a very interesting topic to be further analysed.

The traffic experiments studied during this research (and presented in chapter 6) provide the basis for the subsequent comments on the applicability of FCAC to CAC in ATM networks, more precisely the accuracy of the cell loss ratio predicted by FCAC. The accuracy is measured calculating the error between the cell loss ratio obtained either via simulations or measurements and the cell loss ratio predicted by FCAC. The experiments have the following aims:

- comparison of FCAC with CAC approaches based on theoretical traffic models in order to visualise the traffic scenarios and network configurations where the use of a learning CAC approach, such as FCAC, allows to achieve a higher statistical multiplexing gain;
- comparison of FCAC with another fuzzy logic based CAC approach, NFCAC, that uses neural networks instead of genetic algorithms to automatically design the fuzzy rule base, in terms of (1) the ability to generalise the cell loss ratio prediction to traffic mixes constituted by the same type of traffic sources as the sources considered in the examples, but a different number of sources for each of the traffic types and (2) the number of examples required for training.

The experiments described in chapter 6, section 6.1, for homogeneous traffic mixes of voice, data or image traffic, and for various values of the link capacity (the size of the output buffer of the switch is assumed to be fixed), allow the conclusion that:

- the cell loss ratio predicted by FCAC, subject to the error of rounding to the nearest integer negative power of ten, is within a maximum error of 10^{-1} above or below the cell loss ratio given by simulations for the scenarios for which a cell loss prediction could be made;
- the predicted cell loss ratio does not favour any particular type of traffic or network scenario;
- the accuracy of the prediction is only limited by the size and information contained in the training data examples.

Referring to the last point, a total of 37 examples was available to train and tune the FCAC. The reason why such a small number of examples is used is that the process of obtaining the cell loss ratio for each of the traffic mixes considered in the examples using a simulator is time consuming. Nevertheless, in the experiments in section 6.1, a total of 482 traffic scenarios have been studied; for 5 of these, the cell loss ratio predicted by the FCAC was optimistic and for 11 traffic scenarios, FCAC failed to provide a cell loss ratio prediction.

Comparing thus with other techniques:

- the (M+1)-MMDP approximation is accurate for traffic scenarios where the connections exhibit long bursts (e.g. video and data traffic) and the buffer size is such that at least part of the whole burst can be queued (see chapter 6, figures 6.3);
- the convolution based approach, ECA, is accurate for scenarios for which the multiplexed traffic is characterised by variations in rates that cannot be smoothed by the queueing process at the output buffer (see chapter 6, figures 6.4); in this case, a bufferless model is accurate enough to describe the queueing process at the buffers of the ATM switch.

The results also show that the ECA does not provide an upper bound for the cell loss probability: in fact it is optimistic for traffic scenarios in which the connections present long bursts (e.g. image traffic, see figure 6.5). ECA is very pessimistic for a network scenario where the buffer size is very large (the case studied had a buffer size of 2000 cells) and the (M+1)-MMDP approximation cannot provide a prediction for such cases due to the complexity of the calculations involved (for 300 sources and a buffer of 2000 cells a system of linear equations for which the size of the coefficient matrix is $6.10^5 \times 6.10^5$).

The performance of FCAC in terms of its ability to generalise the cell loss ratio prediction to traffic scenarios for which it has no previous knowledge has been compared in chapter 6, section 6.2 with another fuzzy logic based CAC approach, the neuro-fuzzy CAC (NFCAC) proposed by Fontaine et al ([Font, 1996]). NFCAC uses neural networks instead of genetic algorithms to set-up the fuzzy rule base. FCAC and NFCAC have been trained with a set of examples for which the cell loss ratio values have been obtained from on-line measurements on an ATM test bed (see for details [R2061 D.28, 1994]). The cell loss ratio prediction given by NFCAC has a maximum error of approximately 10^{-1} for all the traffic scenarios. The prediction given by FCAC for the same traffic scenarios has a maximum error of approximately:

- 10^{-1}, for traffic scenarios for which the cell loss ratio is in the range 10^{-2} to 10^{-4};
- 10^{-2}, for traffic scenarios for which the cell loss ratio is in the range 10^{-5} to 10^{-8}.

The reason for different maximum error values is that, of the 80 training examples, only 30% describe traffic scenarios for which the cell loss ratio is in the range 10^{-5} to 10^{-8}, the remaining 70% refer to traffic scenarios for which a cell loss ratio between 10^{-2} and 10^{-4} can be observed.

One of the differences between FCAC and NFCAC is that NFCAC does not include the mean burst length value in the set of parameters that characterise the traffic behaviour of each connection. Because of this, NFCAC cannot distinguish between connections with the same mean and peak bit rates but different values for the mean active and inactive periods. Furthermore, NFCAC limits the maximum number of fuzzy rules by assigning a maximum of 3 fuzzy sets for the fuzzy domain of each of the 3 input variables (one input variable less than FCAC). This bounds the maximum number of fuzzy rules to 3^3. Because a smaller number of input variables is considered and the maximum number of rules is bounded, NFCAC requires a smaller number of training examples than FCAC to achieve the same degree of accuracy in the cell loss ratio prediction. However, in a real implementation, this does not constitute a drawback for FCAC since it is not likely that there will be a shortage of examples to learn from.

The knowledge about ATM traffic expressed in the fuzzy rule base of FCAC can be applied to network configurations that differ in the values of the link capacity. If FCAC has to predict cell loss ratios for a link with a bandwidth capacity different from the bandwidth capacity previously considered, the rule base of FCAC does not have to be re-designed for the new link. This is because the knowledge about ATM traffic expressed in the fuzzy rules is independent of the actual value of the link capacity (the link capacity is normalised to 1 cell per cell slot or 1 Mbit per second).

The network configuration (link capacity and output buffer size) considered in the experiments reported in chapter 6, section 6.2 is the same as the network configuration for the experiments in chapter 6, section 6.3. Thus, the rule base obtained for the experiments reported in section 6.2 can be tuned to enable FCAC to adapt to different traffic mixes composed of traffic types not "known" previously. The tuning algorithm is invoked for a new set of testing data, in order to adjust the values of the parameters that define the fuzzy sets in each of the fuzzy rules of the rule base obtained for the experiments in section 6.2. The ability of FCAC to adapt to new traffic patterns is tested for the same type of sources as the sources considered in the new set of testing examples, but varying the number of sources of one of the traffic types. For the totality of 39 traffic scenarios studied, FCAC failed to provide a cell loss ratio prediction for one of the traffic scenarios and for another the cell loss ratio predicted was not as expected comparing it with the cell loss ratio obtained via simulation. A total of 34 test examples have been used.

A CAC method is required to use a set of traffic (source) parameters that are simple to estimate, either from the statistical analysis of on-line measurements or from the parameters declared by the user at connection set-up. FCAC has chosen the latter option in order to make sure that the parameters used in the decision making process are being enforced by the UPC module. ECA follows the same approach as FCAC to estimate the parameters that define the traffic behaviour of a particular source. A CAC method based on the (M+1)-MMDP approximation requires the identification of the parameters that define a Markov Chain with (M+1) states, where M is the total number of sources in the traffic mix.

The set of input variables used by FCAC to identify the multiplexed traffic pattern might have to be revised if the set of traffic parameters used to describe the traffic generated by each of the multiplexed connections (more precisely the mean and peak bit rates and the mean burst length) does not allow traffic patterns for which different maximum cell loss ratios per connection are expected to be distinguished. The set of input variables can be extended to include additional traffic parameters which could be useful for the CAC decision making process, but this would imply a fuzzy system with a larger number of rules, longer training time and a larger set of training examples than the current fuzzy system. A better re-design strategy would be to substitute one (or more) of the four current input variables by another variable that represents a more influential factor on cell loss than the one being substituted in order to constrain the dimensionality of the rule base.

In principle, FCAC can be used to estimate other QoS parameters, such as cell delay, by extending the set of output parameters to include the cell delay output variable; the same algorithm used to design automatically the rule base of FCAC to enable the tool to predict cell loss ratios can be used to set up FCAC to predict also cell delay values. In practice, some further studies are required to evaluate whether this approach is viable in terms of the accuracy of the cell delay prediction.

On a real implementation, a CAC method is required to make a fast decision and this is determined by the number of connection set-up requests arriving at a node in one second. According to Hiramatsu ([Hiram, 1994]), the typical target for the speed of the calculations is 1 ms so as to handle 1,000 requests per second. These studies consider that a numerical comparison of the calculation speeds of the different CAC methods presented is not too meaningful, because it is bounded by the speed of the processor used. On the other hand, a qualitative comparison is useful:

- FCAC does not consider a fixed number of fuzzy sets per fuzzy variable. The fuzzy rule is defined by defining the fuzzy sets for each fuzzy variable and, thus, the number of fuzzy sets per variable is, in the worst case (if none of the fuzzy sets coincide), as many as the number of fuzzy rules. The maximum number of rules expected (after the generation phase) for a set of h training examples is:
  - h rules, if the covering parameter, ε, is between 0 and 1,
  - a maximum of $2h$ rules if ε is between 1 and 2.

Each rule contains the definition of 4 bell-shaped fuzzy sets (defined by 3 parameters) for the antecedent variables, the definition of the singleton for the consequent variable (defined by a single parameter) and the lower and upper bounds of uncertainty (2 parameters) associated with the rule. Therefore, in total, a maximum of $\lceil \varepsilon \rceil h \times (4 \times 3 + 1 + 2)$ parameters is considered, where $\lceil x \rceil$ stands for the smallest integer exceeding *x*. The inference algorithm of FCAC requires the calculation of the output fuzzy sets for each of the rules (using expressions (4.11) and (4.12)), combining the fuzzy sets (using (4.13)) and calculating the defuzzified output (using (4.15)). The computation complexity is only bounded by the number of rules considered and this number should not be more than 4^4 for a real implementation (the equivalent to 4 fuzzy sets for each input variable).

- NFCAC uses a neural network with 5 layers corresponding to input, fuzzification, rule base, defuzzification and output (3 hidden layers). Hence, the first hidden layer contains as many nodes as the number of membership functions. Since three input fuzzy variables are considered and the fuzzy domain of each variable contains 3 fuzzy sets, the number of neurons in the first hidden layer (representing the fuzzification process) is 3^3. The same reasoning applies for the number of neurons in the second (rule base) and third (defuzzification) hidden layers. The fuzzy sets considered are bell-shaped fuzzy sets, defined by three parameters. Thus, a maximum of 3×3^3 parameters needs to be defined.

- ECA assumes that connections are classified into traffic classes, defined by a GMDP process with *n* states. The bit rate of each of the traffic classes is given by a multinomial distribution function. When more than one traffic class is considered (heterogeneous traffic), the aggregated bit rate is calculated by convolving the partial bit rate distributions for each of the traffic classes. ECA reduces the complexity of the calculations by considering that CAC needs only to admit or reject a new connection demand. Thus, the complete calculation of the probability of congestion (and from this the probability of loss) is not always necessary, because the states of the aggregated process for which the sum of the rates, for each traffic class, is less than the link capacity do not need to be considered. This is because if the cell loss probability value partially convolved, already exceeds the required cell loss ratio, the calculations can be resumed and the new connection rejected. ECA further simplifies the actual convolution algorithm, in order to reduce the complexity of the calculations, by discarding very small cell loss probabilities, partially sorting of the aggregated bit rate for each traffic class and grouping states for those traffic classes with many possible states. The "enhancement" of the convolution algorithm given by ECA allows to have a fast response even when considering a large number of traffic sources, as long as the sources can be classified into a short number of traffic classes. If the number of traffic classes considered is high, the complexity of the calculations cannot be reduced any further.

- The (M+1)-MMDP approximation requires solving a system of linear equations where the size of the coefficient matrix is approximately (MK) x (MK), where M is the number of sources considered in the homogeneous traffic mix and K is the buffer size. The elements of the coefficient matrix are elements of a geometric sequence and, therefore, are easy to calculate. The coefficient matrix has a high sparsity (fraction of elements in the matrix whose value is zero), more precisely, $\frac{2M-1}{2M}$. Yang et al. ([Yang, 1995]) also refer to the fact that the (M+1)-MMDP method, although being robust, requires a lengthy computation and therefore "is appropriate for off-line applications such as network planning […] and other theoretical studies of traffic behaviour of on-off sources".

From all the CAC approaches presented above, FCAC and NFCAC, allow to predict the CLR for a specific traffic scenario in the shortest amount of time, because the speed of the calculations is not heavily depend on the number of multiplexed sources. Two further requirements associated with CAC methods based on learning techniques are:

- the time spent in real time training should be such that it allows CAC to adapt rapidly to changes in the traffic patterns. This is important only for real time training, since the automatic design is to be performed off-line. Hiramatsu states in [Hiram, 1994] that "even, in the real time training, the [required adaptation] speed is rather slow because the target curve [(the cell loss ratio curve)] is not expected to change dramatically in a short period".

- "safe-side control": the safe-side control policy requires CAC to estimate the cell loss ratio to be worse than the actual cell loss ratio when it makes wrong predictions. If CAC underestimates the cell loss ratio and admits more connections than it actually can admit while maintaining a specific cell loss ratio, then all multiplexed connections will experience a cell loss ratio worse than required [Hiram, 1994].

In order to fully test the accuracy of the cell loss ratio prediction given by FCAC, a wider set of training and testing examples would have been required. The enlarged example set would cover more traffic scenarios and different types of traffic than the traffic scenarios studied in chapter 6. Just to show how the task of constructing such a set of examples is not viable using an ATM simulator, let us assume we aim to construct a fuzzy domain with a maximum of 4 fuzzy sets per fuzzy variable. Hence, a minimum of 4^4 training examples would be required to cover all possible traffic patterns (assuming that each example covers a traffic scenario not covered by any other example in the training example set). To use a simulator to calculate the cell loss ratio for each of the traffic patterns represented in the examples is not a viable task to undertake due to time required to run each of the simulations, specially for the cases in which the cell loss ratio to be obtained is very low. But in a real network this does not constitute a problem, as explained earlier.

From the considered in the previous paragraph, it would be useful to have a pre-defined set of traffic scenarios or set of examples (both for training and testing purposes) in order to compare the CAC approaches proposed in the literature. Given that a reference set of training examples is not available at the moment, the number of training examples is very small and there is a lack of realistic training examples, the performance tests of FCAC done during this research comprise those presented in chapter 6.

The thoroughness to which each of the different CAC methods could be compared depends on the assessment of different aspects and it is constrained by further knowledge of the development of ATM based technology and the behaviour of “real” ATM traffic. The assessment of different CAC methods should focus on:

- satisfying the QoS requirements;
- maximisation of the statistical multiplexing gain;
- simplicity in the determination of the parameters that characterise the traffic behaviour of a specific source;
- speed of the decision making process for the admittance/rejection of a new connection;
- speed of the calculation of auxiliary parameters such as the cell loss ratio;
- ability to adapt to new services being introduced;
- simplicity of implementation.

With this criteria in mind, table 7.1 summarises the comparison of CAC approaches performed during this research.

The test of FCAC on a real ATM traffic environment would also allow the evaluation of whether:

- the segregation of ATM traffic with very different characteristics on different ATM links;
- the separation of ATM traffic in two output buffers, one large buffer used for data transfer services and a small buffer used for interactive real-time services,

enables a higher statistical multiplexing gain to be achieved than multiplexing all types of traffic on the same ATM link.

A further criteria for comparison of CAC methods is, as pointed by Kilkki ([Kilk, 1994]),

> “To evaluate the multiplexing efficiency for different source types could be another criteria of comparison of CAC methods.”

An attempt to evaluate the multiplexing efficiency of FCAC for very heterogeneous traffic mixes and to compare it with the efficiency of other CAC approaches has been tried during this research. For this purpose, simulations were executed in order to obtain the cell loss ratios for traffic patterns composed of the voice, data and image sources presented in chapter 6, section 6.1, and the artificially generated traffic sources presented in section 6.2. The calculated cell loss ratio values would then be used to construct a set of training and testing examples for FCAC. Unfortunately, none of these simulations reached a statistically acceptable cell loss ratio during a pre-specified period of two months.

Table 7.1 Comparison of CAC approaches

CAC methods	***Satisfaction of cell loss requirements***	***Improvement of network usage versus peak rate allocation***	***Simplicity of parameter determination***	***Speed calculation at network nodes***	***Ability to adapt to new services***	***Simplicity of implementation***
FCAC	yes, if there is previous knowledge for the traffic scenario for which the CLR prediction is to be made; otherwise either pessimistic or optimistic	yes	identifies the source traffic behaviour using the parameters declared by the user at connection set-up, that is, mean and peak rates and mean burst length	inference calculations are fast; the time spent in "on-line" training depends on required prediction accuracy and how fast the traffic scenarios change over time.	Yes, if the traffic characteristics of new services can be identified using the set of chosen traffic parameters; otherwise, easy to re-design to cope with modifications	I nference algorithm is simple, the learning algorithm is complex.
NFCAC	same as above	yes	uses only the mean and peak rates	same as above	same as above	same as above
ECA	yes, if size of the buffer is such that can cope with cell level fluctuations caused by the asynchronous arrival of cells (cell scale influence); otherwise, conservative	can be very conservative for network scenarios with large buffer size (the convolution algorithm assumes a bufferless service model)	same as above	fast if traffic sources can be grouped in classes and a smaller number of classes (less than 5) is considered	yes, if the traffic characteristics of new services can be captured using the peak and mean cell rates.	Simple, requires only CPU resources.
(M+1)-MMDP	optimistic for traffic scenarios that do not exhibit burst scale congestion	yes	sources are modelled as on-off sources defined by the rate in the active state, the mean on and off periods.	is not suitable for real time implementation due to complexity of calculations	only applicable to homogeneous traffic scenarios	simple, requires solving system of linear equations

8 Conclusions

The use of fuzzy logic techniques as a decision support tool for the CAC traffic control function has been proposed in this thesis. This is new work on methods for CAC. The FCAC uses a fuzzy inference method to estimate the cell loss ratio obtained when multiplexing several connections with different traffic characteristics on the same ATM link. This cell loss ratio value assists in the decision making of admitting or rejecting a new connection into an ATM network, in the sense that, if the estimated cell loss ratio value violates the cell loss ratio requirements of the new connection, then the connection is rejected; otherwise, it is admitted.

The fuzzy rule base and the definition of the fuzzy sets for each of the fuzzy rules is generated using genetic algorithms, making use of their efficient search properties. Another genetic algorithm is then used to simplify the generated rule and check for inconsistencies; this genetic algorithm uses the fuzzy inference output value in the fitness function expression. Finally, a genetic algorithm are also used to tune the fuzzy sets of each of the rules in order to enable the system to adapt to changes in the traffic patterns and their influence on the cell loss ratio of the multiplexed connections. This thesis focuses on the automatic design of the FCAC fuzzy rule base as opposed to the time consuming trial-and-error process of the manual design approach. The same method can, after the design phase, be used to adapt the FCAC to changes in the traffic environment caused by the introduction of new services in the ATM network. Measurements on the traffic characteristics and on the observed cell loss ratio of each of the connections multiplexed on a single ATM link (referred in this thesis as "examples”) are used to acquire knowledge on the cell loss ratio curve for an ATM link. The size of the output buffer of the switch is also taken into consideration to determine the cell loss ratio curve.

The CLR predicted by the FCAC has been compared with CLR results obtained from on-line measurements and from simulations. Subject to the rounding error, the predicted cell loss ratio value to the nearest integer negative power of ten and the availability of a reduced set of examples, the cell loss ratio predicted by FCAC is within (1) a maximum error of 10^{-1} for a prediction given with a degree of certainty over 0.6 and (2) the error is bounded to 10^{-2} for a prediction given with a degree of certainty less than 0.6. FCAC needs to be backed up by a conservative CAC algorithm, such as the peak rate allocation scheme, for the cases in which there is not enough knowledge in the fuzzy rule base to make a cell loss ratio prediction.

The cell loss ratio predicted by FCAC has been compared with predictions given by analytical CAC approaches. FCAC can capture the influence on cell loss caused by the asynchronous arrival of bursts (burst scale influences) whereas a CAC approach based on a convolution based algorithm (that only takes into account the influence of variations in the rate of the connections on the cell loss) can be very pessimistic for scenarios where the buffer size is such that part or the whole burst can be queued before being served. For network scenarios in which the output buffer of the switch is very large (e.g. 2000 cells) the complexity of the calculations prevents an approach such as the (M+1)-MMDP approximation to provide a fast cell loss ratio prediction. The speed of the calculations of the cell loss ratio prediction given by FCAC is only affected by the number of fuzzy rules in the fuzzy rule base and not by the network configuration (link capacity and output buffer size) or the number of multiplexed connections.

The FCAC has also been compared with another fuzzy based CAC approach, the NFCAC approach that uses neural networks to provide for adaptation and learning. The accuracy of the cell loss ratio predicted by NFCAC has been better than the same prediction given by FCAC for the scenarios tested. However, several aspects need to be considered:

- the traffic scenarios used in the comparison experiments are composed of artificially generated sources, not real ATM traffic;

- the output buffer size of the switch is relatively small (27 cells) and therefore does not do justice to an approach such as FCAC that takes into account the reduction on cell loss obtained by queueing the traffic bursts;
- FCAC uses 4 input variables and NFCAC uses only three. Thus, FCAC requires more training examples than NFCAC to have the same accuracy in the cell loss ratio prediction.

The learning method for FCAC based on genetic algorithms used in this research does not require that the number of fuzzy sets defined per input fuzzy variable be fixed. In fact, the fuzzy sets for each input variable are defined in the context of the fuzzy rule and in FCAC there is no static set of the fuzzy sets for each input fuzzy variable (fuzzy domain). The learning method does not require that a maximum number of rules be pre-specified but automatically generates the set of fuzzy rules and thus dictates the number of rules in the fuzzy rule base. This is in contrast with the learning method based on neural networks used in the case of NFCAC in which the maximum number of fuzzy sets per fuzzy variable and thus the maximum number of fuzzy rules to be obtained are pre-specified.

Although the number of traffic mixes studied does not allow the performance of FCAC for real ATM traffic to be exhaustively validated, the research enabled the conclusion that the use of a CAC approach such as FCAC is recommended for:

- links served by a switch with a large output buffer (e.g. 2000 cells) capable of queueing not only cells resulting from the asynchronous arrival of cells from active connections but also the whole or part of the traffic bursts.
- traffic mixes in which any of the multiplexed connections has a traffic behaviour close to the burst scale for which the obtained cell loss ratio is not only dependent on the availability of time resources (bandwidth) but also on the space resources (size of the output buffer of the switch).

Thus, FCAC is very useful for ATM services such as file transfer and image retrieval that are sensitive to loss but not to transmission delays.

The results obtained with the FCAC tool give confidence in proposing it as a valid alternative to other CAC approaches as long as the user requirements in terms of cell loss can expressed by a negative power of ten.

Appendix

A. Cell loss ratio results for experiments A.1 to A.6

The following tables contain the cell loss ratio results obtained for experiments A.1 to A.6 (see table 6.3) performed for a single ATM link and plotted in figures 6.1 to 6.6 of chapter 6, section 6.1. The description of the characteristics of the traffic sources (voice, data and image) is shown in table 6.2.

The tables contain the following values[1]:

- the average CLR predicted by the (*M*+1)-MMDP approximation and ECA approach (referred in the tables as $CLR_{\text{(M+1)-MMDP}}$ and CLR_{ECA}, respectively);
- the average CLR (CLR_{avg}) obtained via simulation,
- the maximum CLR per connection predicted by FCAC (CLR_{FCAC});
- the maximum CLR (CLR_{max}) obtained also via simulation.

Experiment A.1. Cell loss ratio obtained for *voice* sources (C= 7 Mbit/s, K=50 cells):

Number sources	$CLR_{\text{(M+1)-MMDP}}$	CLR_{avg} *(Simul.)*	CLR_{ECA}	CLR_{max} *(Simul.)*	CLR_{FCAC}
250	7.00E-06*	1.03E-05*	7.00E-06	1.09E-05*	1.00E-05
260	5.00E-05*	1.00E-04	4.10E-05	2.59E-04	1.00E-05
270	3.00E-04*	2.90E-04	1.90E-04	3.37E-04	1.00E-04
280	1.20E-03*	9.32E-04	6.90E-04	9.58E-04	1.00E-03
290	3.10E-03*	3.08E-03	2.00E-03	3.25E-03	1.00E-02
300	8.00E-03*	7.65E-03	5.10E-03	8.02E-03	1.00E-02

[1] The notation *x.yz*E-0*t* used in the values shown in the tables means $x.yz\times10^{-t}$, the asterisk means that the value has been read from [Yang, 1995], pp. 123-4 and not from simulation.

Experiment A.2. Cell loss ratio obtained for *voice* sources (C= 0.7 Mbit/s, K=50 cells):

Number sources	$CLR_{(M+1)\text{-}MMDP}$	CLR_{avg} *(Simul.)*	CLR_{ECA}	CLR_{max} *(Simul.)*	CLR_{FCAC}
15	3.00E-06*	5.00E-05*	8.50E-05	1.00E-04*	1.00E-04
17	3.00E-05*	3.08E-04	4.80E-04	3.48E-04	1.00E-04
19	2.00E-04*	9.09E-04	1.80E-03	9.43E-04	1.00E-03
21	1.00E-03*	2.16E-03	4.80E-03	2.18E-03	1.00E-03
23	3.50E-03*	5.08E-03	1.10E-02	5.12E-03	1.00E-03
25	1.00E-02*	1.12E-02	2.00E-02	1.13E-02	1.00E-02

Experiment A.3. Cell loss ratio obtained for *data* sources (C= 350 Mbit/s, K=50 cells):

Number sources	$CLR_{(M+1)\text{-}MMDP}$	CLR_{avg} *(Simul.)*	CLR_{ECA}	CLR_{max} *(Simul.)*	CLR_{FCAC}
160	1.00E-07*	1.00E-07*	2.70E-07	1.00E-07*	1.00E-07
180	2.30E-06*	2.48E-06	4.10E-06	2.60E-06	1.00E-06
200	2.40E-05*	2.71E-05	3.60E-05	2.73E-05	1.00E-05
220	1.50E-04*	1.56E-04*	2.10E-04	1.56E-04*	1.00E-04
240	5.80E-04*	7.02E-04	8.90E-04	7.02E-04	1.00E-03
260	2.00E-03*	2.58E-03	2.90E-03	2.79E-03	1.00E-02
280	6.00E-03*	6.93E-03	7.50E-03	7.42E-03	1.00E-02
300	1.50E-02*	1.58E-02	1.60E-02	1.73E-02	1.00E-02

Experiment A.4. Cell loss ratio obtained for *data* sources (C= 52 Mbit/s, K=50 cells):

Number sources	$CLR_{(M+1)\text{-}MMDP}$	CLR_{avg} *(Simul.)*	CLR_{ECA}	CLR_{max} *(Simul.)*	CLR_{FCAC}
8	1.00E-05*	1.66E-05	2.40E-05	1.66E-05	1.00E-05
10	5.00E-05*	3.06E-04	1.30E-04	3.90E-04	1.00E-04
12	2.00E-04*	1.61E-03	4.05E-04	2.16E-03	1.00E-03
15	6.90E-04*	3.68E-03	1.40E-03	4.14E-03	1.00E-03
18	2.00E-03*	5.62E-03	3.60E-03	6.07E-03	1.00E-02
22	5.50E-03*	1.00E-02	9.10E-03	1.07E-02	1.00E-02
25	1.20E-02*	1.51E-02	1.60E-02	1.59E-02	1.00E-02

Experiment A.5. Cell loss ratio obtained for *image* sources (C= 30 Mbit/s, K=50 cells):

Number sources	$CLR_{(M+1)\text{-}MMDP}$	CLR_{avg} *(Simul.)*	CLR_{ECA}	CLR_{max} *(Simul.)*	CLR_{FCAC}
80	1.00E-07*	5.78E-07*	1.10E-07	1.12E-06*	1.00E-06
113	1.00E-05*	4.16E-05*	8.80E-06	5.16E-05*	1.00E-05
150	2.00E-04*	7.63E-04*	2.10E-04	9.63E-04*	1.00E-03
187	1.50E-03*	4.00E-03*	1.70E-03	5.00E-03*	1.00E-03
220	5.80E-03*	6.32E-03	6.60E-03	6.55E-03	1.00E-02

Experiment A.6. Cell loss ratio obtained for *image* sources (C= 7 Mbit/s, K=50 cells):

Number sources	$CLR_{(M+1)\text{-}MMDP}$	CLR_{avg} *(Simul.)*	CLR_{ECA}	CLR_{max} *(Simul.)*	CLR_{FCAC}
4	9.00E-06*	2.53E-05*	1.00E-05	2.53E-05*	1.00E-05
8	3.00E-04*	4.70E-04	3.40E-04	6.60E-04	1.00E-03
12	1.40E-03*	2.58E-03	1.50E-03	3.17E-03	1.00E-03
18	5.00E-03*	6.76E-03	5.40E-03	7.68E-03	1.00E-02
22	1.00E-02*	9.44E-03	9.90E-03	9.98E-03	1.00E-02

B. Cell loss ratio results for experiments B.1 to B.6 and C.1 to C.9

The following tables contain the cell loss ratio values obtained via on-line measurements in the Exploit ATM test-bed (see [R2061 D.28, 1994]) for homogeneous traffic and heterogeneous traffic mixes. The characteristics of the multiplexed traffic sources are shown in table 6.4. The ATM link has a capacity of 155.52 Mbit/s and the output buffer size of the switch is 27 cells long. The cell loss ratio measurements were used to construct a set of examples for training the FCAC and NFCAC heuristic CAC approaches. The cell loss ratio predictions given by FCAC and NFCAC based on the knowledge obtained from the examples are plotted in figures 6.13 to 6.18b of chapter 6, section 6.2.

The tables contain the following values:

- the maximum CLR per connection, CLR_{max}, obtained via on-line measurements;
- the number and type of each of the sources in the traffic mix.

Experiment B.1. Cell loss ratio obtained for *A.3.1* sources (C= 155.52 Mbit/s, K=27 cells).

N. sources	7	8	10	12	14	16
CLR_{max}	2.02×10^{-4}	5.79×10^{-4}	2.89×10^{-3}	7.53×10^{-3}	1.58×10^{-2}	2.15×10^{-2}

Experiment B.2. Cell loss ratio obtained for *A.3.2* sources (C= 155.52 Mbit/s, K=27 cells).

N. sources	14	18	22	26	30	34
CLR_{max}	1.87×10^{-5}	7.77×10^{-5}	3.21×10^{-4}	7.38×10^{-4}	1.49×10^{-3}	2.71×10^{-3}

Experiment B.3. Cell loss ratio obtained for *B.3.1* sources (C= 155.52 Mbit/s, K=27 cells).

N. sources	22	24	26	28	30	32	34
CLR_{max}	1.59×10^{-5}	5.87×10^{-6}	6.96×10^{-5}	3.81×10^{-4}	1.38×10^{-3}	3.83×10^{-3}	8.85×10^{-3}

Experiment B.4. Cell loss ratio obtained for *B.3.2* sources (C= 155.52 Mbit/s, K=27 cells).

N. sources	180	200	220	240	260	280
CLR_{max}	1.84×10^{-5}	4.01×10^{-5}	1.93×10^{-4}	6.34×10^{-4}	1.58×10^{-3}	3.95×10^{-3}

Experiment B.5. Cell loss ratio obtained for *C.3.1* sources (C= 155.52 Mbit/s, K=27 cells).

N. sources	115	120	125	130	135	140	150	160
CLR_{max}	4.81×10^{-8}	3.53×10^{-7}	6.34×10^{-6}	4.61×10^{-5}	2.56×10^{-4}	1.137×10^{-3}	7.71×10^{-3}	2.93×10^{-2}

Experiment B.6. Cell loss ratio obtained for *C.3.2* sources (C= 155.52 Mbit/s, K=27 cells).

N. sources	260	270	280	300	320	340	150	160
$CLR_{\max}$	4.39×10^{-8}	3.11×10^{-7}	1.32×10^{-6}	2.82×10^{-5}	2.30×10^{-4}	1.35×10^{-3}	4.99×10^{-3}	1.41×10^{-2}

Experiment C.1. Cell loss ratio obtained for a traffic mix composed of a variable number of *B.3.1* sources and 2 *A.3.1* sources (C = 155.52 Mbit/s, K=27 cells).

Number B.3.1 sources	14	18	26	30
$CLR_{\max}$	5.50×10^{-6}	1.18×10^{-4}	4.92×10^{-3}	1.48×10^{-2}

Experiment C.2. Cell loss ratio obtained for a traffic mix composed of a variable number of *B.3.1* sources and 4 A.3.1 sources (C = 155.52 Mbit/s, K=27 cells).

Number B.3.1 sources	6	8	12	16	20	24
$CLR_{\max}$	1.31×10^{-5}	5.03×10^{-5}	3.81×10^{-4}	1.71×10^{-3}	6.91×10^{-3}	1.63×10^{-2}

Experiment C.3. Cell loss ratio obtained for a traffic mix composed of a variable number of *B.3.2* sources and 2 A.3.1 sources (C = 155.52 Mbit/s, K=27 cells).

Number B.3.2 sources	74	130	150	170	190	230
$CLR_{\max}$	1.00×10^{-7}	4.83×10^{-5}	1.96×10^{-4}	5.57×10^{-4}	1.29×10^{-3}	5.59×10^{-3}

Experiment C.4. Cell loss ratio obtained for a traffic mix composed of a variable number of *B.3.2* sources and 4 A.3.1 sources (C = 155.52 Mbit/s, K=27 cells).

Number B.3.2 sources	20	60	80	100	140	160
$CLR_{\max}$	2.87×10^{-7}	5.98×10^{-5}	2.10×10^{-4}	4.13×10^{-4}	1.87×10^{-3}	3.90×10^{-3}

Experiment C.5. Cell loss ratio obtained for a traffic mix composed of a variable number of *C.3.1* sources and 2 A.3.1 sources (C = 155.52 Mbit/s, K=27 cells).

Number C.3.1 sources	70	80	100	120
CLR_{max}	1.72×10^{-7}	1.38×10^{-5}	1.61×10^{-3}	8.25×10^{-3}

Experiment C.6. Cell loss ratio obtained for a traffic mix composed of a variable number of *C.3.1* sources and 4 A.3.1 sources (C = 155.52 Mbit/s, K=27 cells).

Number C.3.1 sources	24	32	40	50	60	70	80	100
CLR_{max}	2.16×10^{-7}	2.49×10^{-5}	1.20×10^{-4}	3.40×10^{-4}	5.38×10^{-4}	2.21×10^{-3}	3.69×10^{-3}	1.05×10^{-2}

Experiment C.7. Cell loss ratio obtained for a traffic mix composed of a variable number of *C.3.1* sources and 4 B.3.1 sources (C = 155.52 Mbit/s, K=27 cells).

Number C.3.1 sources	120	140
CLR_{max}	8.32×10^{-4}	8.88×10^{-3}

Experiment C.8. Cell loss ratio obtained for a traffic mix composed of a variable number of *C.3.1* sources and 8 B.3.1 sources (C = 155.52 Mbit/s, K=27 cells).

Number C.3.1 sources	80	100	120
CLR_{max}	2.33×10^{-6}	7.49×10^{-4}	1.41×10^{-2}

C. Cell loss ratio results for experiments D.1 to D.9

The following tables contain the cell loss ratio values obtained via on-line measurements (Exploit ATM test-bed) and simulation for heterogeneous traffic mixes. The characteristics of the multiplexed traffic sources are shown in table 6.4. The ATM link has a capacity of 155.52 Mbit/s and the output buffer size of the switch is 27 cells long. The cell loss ratio measurements were used to construct a set of test examples. The tuning process was executed for the new set of test examples and the cell loss ratio predicted by FCAC for the new traffic mixes was evaluated. The cell loss ratio predictions given by FCAC are plotted in figures 6.28 to 6.36 of chapter 6, section 6.3.

The tables contain the following values:

- the maximum CLR per connection, CLR_{max}, obtained via on-line measurements and simulation;
- the number of ATM-100 sources in the traffic mix.

For a description of the characteristics of the traffic sources for each of the experiments see table 6.6 (background traffic) and table 6.7 (foreground traffic).

Experiment D.1:

Number ATM-100 sources	1	2	3	4	5
CLR_{max}	3.04×10^{-4}	5.96×10^{-4}	7.04×10^{-4}	8.59×10^{-4}	1.12×10^{-4}

Experiment D.2:

Number ATM-100 sources	1	2	3	4	5
CLR_{max}	3.08×10^{-4}	7.31×10^{-4}	1.04×10^{-3}	1.38×10^{-3}	1.89×10^{-3}

Experiment D.3:

Number ATM-100 sources	1	2	3	4	5	6
CLR_{max}	2.99×10^{-4}	5.21×10^{-4}	6.53×10^{-4}	7.97×10^{-4}	9.10×10^{-4}	1.25×10^{-3}

Experiment D.4:

Number ATM-100 sources	1	2	3	4	5
CLR_{max}	5.14×10^{-4}	6.52×10^{-4}	9.04×10^{-4}	1.28×10^{-3}	1.68×10^{-3}

Experiment D.5:

Number ATM-100 sources	1	2	3	4
CLR_{max}	5.42×10^{-4}	6.44×10^{-4}	8.88×10^{-4}	1.18×10^{-3}

Experiment D.6:

Number ATM-100 sources	1	2	3	4
CLR_{max}	9.59×10^{-5}	4.55×10^{-4}	7.79×10^{-4}	9.54×10^{-4}

Experiment D.7:

Number ATM-100 sources	1	2	3
CLR_{max}	5.54×10^{-4}	7.54×10^{-4}	1.04×10^{-3}

Experiment D.8:

Number ATM-100 sources	1	2	3
CLR_{max}	4.59×10^{-4}	7.28×10^{-4}	1.05×10^{-3}

Experiment D.9:

Number ATM-100 sources	1	2	3	4
CLR_{max}	6.08×10^{-4}	7.63×10^{-4}	9.96×10^{-4}	1.49×10^{-3}

References

[Ram, 1994] M. F. Ramalho and E. M. Scharf, "Fuzzy Logic Based Techniques for Connection Admission Control in ATM Networks", Proc. of the 11th UK Teletraffic Symposium, Cambridge, March 1994, paper 12A.

[Ram, 1995] M. F. Ramalho and E. M. Scharf, "Developing a Fuzzy Logic Tool using Genetic Algorithms for Connection Admission Control in ATM Networks", Proc. of the 6th Int'l Fuzzy Systems Association World Congress, Vol. 1, pp. 281-284, S. Paulo, 22-28th July 1995.

[Ram, 1996] M. F. Ramalho and E. M. Scharf, "The application of Fuzzy Logic Techniques and Genetic Algorithms for Connection Admission Control in ATM Networks", in: Genetic Algorithms and Soft Computing, F. Herrera, J.L. Verdegay (Eds.), Physica-Verlag (Studies in Fuzziness, Vol. 8), pp. 615-640, 1996.

[Ram, 1996] M. F. Ramalho and E. M. Scharf, "Fuzzy Logic Tool and Genetic Algorithms for CAC in ATM Networks" in *Electronic Letters*, vol. 32, no. 11, pp. 973-974, 1996.

[Aag, 1993] Aagesen F. A., "A flow Management Architecture for B-ISDN", Proc. IBCN&S, Copenhagen, pp. 14.3.1-14, 1993.

[Ant, 1989] H.J. Antonisse, "A new interpretation of schema notation that overturns the binary encoding constraint", in [Schaf, 1989], pp. 86-91.

[Ackl, 1987] D.H. Ackley, "An empirical study of Bit Vector Function Optimisation", in [Davis, 1987], pp. 170-204.

[Andr, 1994] Andrade J., Martinez-Pascua M.J., "Use of the IDC to characterise LAN Traffic", *Proc. of the IFIP Workshop on Performance Modelling and Evaluation of ATM Networks*, U. Bradford, paper 15, July 1994.

[Andr, 1991] Andrade J., Burakowski W., Villen-Altamirano M., "Characterisation of Cell Traffic by an ATM source", Proc. of 1991 I TC-13, pp.545-550, Elsevier Science Publishers B.V. (North-Holland), Editors A. Jensen and V-B. Iversen, IAC 1991.

[Arak, 1992] S. Araki, H. Nomura, I. Hayashi, N. Wakami, "Self-generating method of fuzzy inference rules", *Int'l Fuzzy Engineering Symposium* (IFES'92), 1992, pp. 1047-1058.

[ATM-F, 1993] ATM-Forum, "ATM-User Network Interface Specification", Version 3.0, 1993.

[Bäck, 1991] T. B Bäck, F. Hoffmeister, “Extended Selection Mechanisms in Genetic Algorithms”, in [Bel, 1991], pp. 92-99.

[Bäck, 1992] T. B Bäck, “A user’s guide to GENEsYs 1.0”, University of Dortmund, Department of Computer Science, Systems Analysis Research Group, PO Box 50 05 00, D-4600 Dortmund 50, July 1992.

[Bak, 1985] J.E. Baker, "Adaptive Selection Methods for Genetic Algorithms”, in [Gref, 1985], pp. 101-111..

[Bak, 1987] J.E. Baker, "Reducing Bias and Inefficiency in the Selection Algorithm”, in [Gref, 1987]. Pp. 14-21.

[Bel, 1991] R. Belew, L. Booker (Editors), Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kafmann Publishers, Los Altos, CA, 1991.

[Berg, 1995] A. W. Berger, “Traffic control to support new telecommunication services based on ATM: a review of activities of the ITU and ATM Forum”, proceedings International Teletraffic Conference, San Petersburg, Russia Fed., pp. 383-392, 25 June to 2 July, 1995.

[Beth, 1976] A. D. Bethke, “Comparison of genetic algorithms and gradient-based optimizers on parallel processors: Efficiency of use of processing capacity (Technical Report no. 197). Ann Arbor: University of Michigan, Logic of Computers Group.

[Beth, 1981] A. D. Bethke, “Genetic algorithms as function optimisers. (Doctoral dissertation, University of Michigan). Dissertation Abstracts International 41 (9), 3503B. (University Microfilms no. 8106101).

[Blond, 1992] C. Blondia, O. Casals, "Performance analysis of statistical multiplexing of VBR sources", *Proc. of the 2nd Race Workshop on traffic and perform. aspects in IBCN*, Aveiro, Portugal, Jan. 1992, session 4.

[Bon, 1995] F. Bonomi, “Progress towards the definition of ABR in the Traffic Management Subworking Group at the ATM Forum”, AT&T Bell Labs, ATM Forum, *Proc. of ATM Hot Topics on Traffic and Performance: from RACE to ACTS*, Milano, paper 1, June 1995.

[Boy, 1990] P. Boyer, "A congestion control for the ATM", *7th ITC Specialist Seminar*, Morristown, October 1990, paper 4.3.

[Book, 1987] L.B.Booker, “Improving search in Genetic Algorithms” in [Davis, 1987], pp.61-73.

[Brad, 1969] P.T. Brady, “A model for generating on-off speech in two-way conversations”, Bell Syst. Tech. Journal, vol.48, Sept. 1969.

[Camp, 1989] L.M. Campos and M.G. Bolanos, “Representation of fuzzy measures through probabilities”, *Fuzzy Sets and Systems*, vol.. 31, pp. 23-36, 1989.

[Camp, 1992] L.M. Campos and S. Moral, “Propagating uncertain information forward”, *Int. Journal of Intelligent Systems*, vol. 7, pp. 15-24, 1992.

[Camp, 1993] L.M. Campos and A. Gonzalez, “A fuzzy inference model based on an uncertainty forward propagation approach”, *Int. Journal Approximate Reasoning*, vol. 9, pp. 139-164, 1993.

[CFS, 1992] CFS D510, "General Aspects of Quality of Service and System Performance in IBC", *RACE Common Functional Specifications*, Document 3: Networking and Internetworking 2, December 1992.

[Choq, 1953] Choquet G., "Theory of capacities", Ann. Inst. Fourier 5, pp. 131-295, 1953.

[COST, 1991] COST 224 Final Report, Management Committee of the COST 224 Project, "Performance Evaluation and Design of Multiservice Networks", Edited by J.W. Roberts, October 1991.

[Cuth, 1993] Sapanel J.C., Cuthbert L.G., "ATM The Broadband Telecommunication Solutions", IEE Telecommunications Series 29, 1993.

[Davis, 1987] L. Davis (Editor), "Genetic Algorithms and Simulated Annealing", Morgan-Kaufmann Publishers, Los Altos, CA, 1987.

[Davis, 1989] L. Davis, "Adapting Operator Probabilities in Genetic Algorithms", in [Schaf, 1989], pp.61-69, 1989.

[Delb, 1981] L.E.N. Delbrouck, "A unified approximate evaluation of congestion functions for smooth and peaky traffic", *IEEE Trans. Comm*., vol. Com-29, pp 85-91, Feb. 1981.

[Delg, 1993] Delgado M. and Gonzalez A., "An inductive learning procedure to identify fuzzy systems", *Fuzzy Sets and Systems*, 55 , pp.121-132 , 1993.

[Delg, 1992] Delgado M. and Gonzalez A., "A frequency model to assess Dempster/Shafer's evidences", *Int. Journal Approximate Reasoning*, vol. 11, pp. 159-174, 1994.

[DeJo, 1975] K. De Jong, "An analysis of the behaviour of a class of genetic adaptive systems", *PhD Thesis*, University of Michigan, 1975. Diss. Abstr. Int. 36 (10), 54140B, University Microfilms no. 76-9381.

[DeJo, 1985] K. De Jong, "Genetic Algorithms: A 10 year perspective", in [Gref, 1985], pp. 169-177.

[Demp, 1967] Dempster A.P., "Upper and lower probabilities induced by a multivalued mapping", Ann. Math. Statistics 38, 325-339, 1967.

[Dub, 1988] Dubois D., and Prade H., "Possibility Theory: An approach to computerized processing of uncertainty", Plenum Press, New York, 1988.

[Dub, 1991] Dubois D., and Prade H., "Fuzzy sets in approximate reasoning", Part 1: Inference with possibility distributions, *Fuzzy Sets and Systems* 40, pp. 143-202, 1991.

[Dub, 1988] D. Dubois, R. Martin-Cllouaire, and H. Prade, "Practical Computing in Fuzzy Logic" in *Fuzzy Computing*, Editors: M.M. Gupta and T. Yamakawa, Elsevier Science Publishers B.V. (North-Holland), 1988.

[Eck, 1989] A.E.Eckberg, D.T.Luan and D.M.Lucantoni, "Meeting the challenge: Congestion and flow control strategies for broadband information transport", in *Proc. GLOBECOM'89*, 1989, pp. 49.3.1-49.3.5.

[Eck, 1991] A.E.Eckberg, D.M.Lucantoni and P.K. Prasana, "Congestion control issues and strategies associated with B-ISDN / ATM Access and Network Transport", in *Proc. Int'l Symp. on Subscriber Loops and Services*, Amsterdam, 1991.

[Ekl, 1988] B. Eklundh, K. Sallberg, B. Stavenow, "Asynchronous Transfer Mode-Options and Characteristics", *12th International Teletraffic Congress*, Torino, Italy, June 1988.

[Esh, 1993] L.J. Eshelman, J.D. Schaffer, "Real-coded Genetic Algorithms and Interval-Schemata", in *Foundations of Genetic Algorithms* 2, L. Darell Whitley (Ed.), Morgan Kaufmann Publishers, San Mateo, pp.187-202, 1993.

[Fabr, 1994] R.Fabregat-Gesa, J. Sole-Pareta J.L.Marzo-Lazaro, J. Domingo-Pascual, "Bandwidth Allocation Based on Real Time Calculations Using the Convolution Approach", GLOBECOM'94, pp. 788-793, 1994.

[Fabr, 1995] R.Fabregat-Gesa, J.L.Marzo-Lazaro and Pere Ridao, "Resource dimensioning aspect of heterogeneous traffic with different service requirements: integration versus segregation" 12th IEE UKTS, 1995.

[Fodor, 1993] J. Fodor, T. Keresztfalvi, "Non-standard Connectives in Fuzzy Logic", Proceedings IEEE Conference, ISBN 0-7803-0614-7/93, pp. 1055-1058, 1993.

[Fogel, 1966] L.J.Fogel, A.J.Owens and M.J.Walsh, "Artificial Intelligence through Simulated Evolution", John Wiley, New York, 1966.

[Font, 1996] M. Fontaine and D. G. Smith, "A neuro-fuzzy approach to connection admission control in ATM networks", 13th IEE UKTS, 18th-20th March, Glasgow, 1996, session 1, paper 2, pp.2/1-2/8.

[Foul, 1993] L. Foulloy, S. Galichet, "Fuzzy Controllers Representation", *Proc. EUFIT'93*, Aachen, Sept. 1993, pp. 142-148.

[Fowl, 1991] H.J. Fowler, W.E. Leland, "Local Area Network Traffic Characteristics with implications for Broadband Network Congestion Management", *IEEE Journal on Selec. Areas on Communications*, Vol. 9 no. 7, Sept. 1991.

[Fuhr, 1991] S. Fuhrman and J-Y Le Boudec, "Burst and cell models for ATM buffers", in Proc. ITC-13, Copenhagen, Denmark, June 1991, pp. 975-980.

[Gihr, 1991] O. Gihr, P. Tran-Gia, "A layered descrition of ATM cell traffic streams and correlation analysis", *Proc. IEEE INFOCOM '91*, Florida, April, 1991.

[Gold, 1989] D. Goldberg, *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley Publishing Company, Inc., 1989.

[Gold, 1991] D.E. Goldberg, The Theory of Virtual Alphabets, in *Parallel Problem Solving from Nature*, P.H. Schwefel, R. Männer (Ed.), Springer-Verlag, Berlin, pp 13-22, 1991.

[Gon, 93] Gonzalez A., Perez R. and Verdegay J.L., "Learning the structure of the rule" in Proc. of Enfit Conference, 1993.

[Gon, 1995] A. Gonzalez, R. Perez, "Completeness and consistency conditions for learning fuzzy rules", Technical Report DECSAI-95103 (1995).

[Gref, 1985] J.J. Grefenstette ed., "Proceedings of the First International Conference on Genetic Algorithms and their Applications", Hillsdale, NJ: Lawrence Erlbaum Associates, 1985.

[Gref, 1987] J.J. Grefenstette ed., "Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms", Hillsdale, NJ: Lawrence Erlbaum Associates, 1987.

[Gref, 1987a] J.J. Grefenstette, "A user's guide to GENESIS", Navy Centre for Applied Research in Artificial Intelligence, Washington, D.C., 1987.

[Guer, 1991] R. Guerin, H. Ahmadi, M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks", *IEEE J. on Sel. areas in com.*, vol. 9, no.7, Sept. 1991.

[Gup, 1988] M. Gupta, T. Yamakawa, Fuzzy Logic in Knowledge-Based Systems, Decision and Control, Elsevier Science Publishers, 1988.

[Hand, 1989] R. Handel, "Evolution of ISDN towards broadband ISDN", *IEEE Net.*, pp. 7-13, 1989.

[Heff, 1986] H. Heffes, D.M. Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and related statistical multiplexer performance", *IEEE J. on Selected areas in communications.*, vol. SAC-49, no.6, pp.856-868, Sept. 1986.

[Hell, 1992] H. Hellendoorn, "The generalised modus ponens considered as a fuzzy relation", *Fuzzy sets and systems* 46, pp. 29-48, 1992.

[Hemm, 1989] H. Hemmer, P.T. Huth, "Police functions in ATM networks", *RACE contrib. NTA_123_0012_CD_CC*, Sep. 1989.

[Hemm, 1991a] H. Hemmer, P.T.Huth, "Combinations of policing functions in ATM networks", *RACE contrib. NTRD_123_0018_CD_CC*.

[Hemm, 1991b] H. Hemmer, "Usage parameter control and cell delay variation", *RACE contrib. NRT_123_0022_CD_CC*, Aveiro, September, 1991.

[Hemm, 1992] H. Hemmer, "Further results on UPC functions and Cell Delay Variation", *RACE contrib. NTR_123_0026_CD_CC*, June 1992.

[Herr, 1993] F. Herrera, M. Lozano, J. Verdegay, "Crossover Operators and Offspring Selection for real coded Genetic Algorithms", Technical Report DECSAI-93113, October, 1993.

[Herr, 1994] F. Herrera, M. Lozano, J. Verdegay, "The use of Fuzzy Connectives to design real-coded Genetic Algorithms", *Mathware and Soft Computing*, vol. 3 pp. 239-251, 1994.

[Herr, 1995] F. Herrera, M. Lozano, J. Verdegay, “A Learning Process for Fuzzy Control Rules using Genetic Algorithms”, Technical Report DECSAI-95108 (1995).

[Hiram, 1990] A. Hiramatsu, "ATM communications network control by neural networks", *IEEE Trans. on Neural Networks*, vol. 1, no.1, pp. 122-130, March 1990.

[Hiram, 1995] A. Hiramatsu, "Training techniques for neural network applications in ATM”, IEEE Communications Magazine, October, pp. 63-67, 1995.

[Holl, 1968] J.H. Holland, Hierarchical descriptions of universal spaces and adaptive systems, (Technical Report ORA Projects 01252 and 08226). Ann Arbor: University of Michigan, Department of Computer Sciences, 1968.

[Holl, 1975] J.H. Holland, Adaptation in natural and artificial systems, The University of Michigan Press, Ann Arbor, 1975.

[Hui, 1988] J.Y.Hui, "Resource Allocation for broadband networks", *IEEE J. on Sel. A. in Com.*, vol. SAC-6, no.9, pp. 1598-1608, Dec. 1988.

[Ichik, 1992] R. Ichikawa, K. Nishimura, M. Kunugi, K. Shimada, "Auto-tuning method of fuzzy membership functions using Neural Networks learning algorithm", *Proc. of the 2nd Int'l on Fuzzy Logic and Neural Networks* (IIZUKA'92), pp. 345-348.

International Telecommunication Union (ITU) - Telecommunications Recommendations (see also http://www.itu.ch):

* Descriptions of ISDNs
[ITU, 1984] ITU COM XVIII-228-E, Geneva, March, 1984.
[ITU I121, 1991] ITU *Rec. I.121*, "Broadband aspects of B-ISDN”, Rev. 1, Geneva, April 1991.

* General description of asynchronous transfer mode
[ITU I150, 1995] ITU *Rec. I.150*, "B-ISDN asynchronous transfer mode functional characteristics", November 1995.

* General aspects of services in ISDN
[ITU I211, 1993] ITU *Rec. I.211*, “B-ISDN service aspects”, March 1993.

* Network functional principles
[ITU I311, 1993] ITU *Rec. I.311*, "B-ISDN general network aspects", March 1993.

* General network requirements and functions
[ITU I371, 1995] ITU: *Rec. I.371*, *Study Group 13*, “Traffic Control and Congestion Control in B-ISDN”, Geneva, 1995.
[ITU I371 A1, 1995] ITU *Rec. I.371*, *Appendix* 1, “Examples of application of the Equivalent Terminal for the Peak Cell Rate definition”, 1995.

* Performance objectives
[ITU I350, 1993] ITU *Rec. I.350*, “General aspects of quality of service and network performance in digital networks, including ISDNs”, March, 1993.

[Ivers, 1987] V.B. Iversen, "The exact evaluation of multi service systems with access control", *Teleteknik*, English Ed., 1987, no. 2, pp. 56-61. The program ATMOS implements the algorithm.

[Ivers, 1990] V.B. Iversen and Liu Y., "The performance of convolution algorithms for evaluating the total load in an ISDN system", 9th Nordic Teletraffic Seminar, Norway, pp. 14, Aug. 1990.

[Ivers, 1990a] V.B. Iversen, A.B. Nielsen, "Policing based on a counting mechanism", *Ninth Nordic Teletraffic Seminar*, Aug. 1990.

[Jacob, 1990] S.B. Jacobsen, K. Moth, L. Dittmann, "Load control in ATM networks", *XIII International Switching Symposium*, Stockholm, Sweden, May 27-June 1, 1990, Proc. vol. V, pp. 131-138.

[Jang, 1992] R. Jang, "Fuzzy Controller design without domain experts", *Proc. IEEE Int. Conf. on Fuzzy Systems* (FUZZ-IEEE'92), 1992, pp. 289-296.

[Jani, 1990] C. Janikow, Z. Michalewicz, "Specialized Genetic Algorithms for Numerical Optimization Problems", *Proceedings of the International Conference on Tools for AI*, pp.798-804, 1990.

[Karr, 1991] C. Karr, "Applying Genetics to Fuzzy Logic", AI Expert, Vol. 6, No.2, 1991, pp. 26-33.

[Karr, 1992] C. Karr, E. Gentry, "A genetics-based adaptive pH Fuzzy Logic Controller", *Proc. of the Int'l Fuzzy Systems and Intelligent Control Conf.* (IFSICC'92), Louisville, KY, 1992, pp. 255-264.

[Katay, 1992] R. Katayama, Y. Kajitani, Y. Nishida, "A self generating and tuning method for Fuzzy Modeling using interior penalty method", *Proc. of the 2nd Int'l on Fuzzy Logic and Neural Networks* (IIZUKA'92), pp. 349-352, 1992.

[Kawa,1988] M. Kawarazaki, T.Okada and M. Sasakawa, "Perspective of ATM communication technique-Evolution to broadband communication network", *J.EICE*, vol.71, no. 8, pp. 806-814, 1988 (in japanese).

[Kawash,1990] M. Kawashima and H. Saito, "Teletraffic issues in ATM networks", *Comput. Net. ISDN.*, vol. 20, pp. 369-375, 1990.

[Kell, 1992] J. Keller and R. Krishnapuram, "Fuzzy set methods in computer vision", in An Introduction to Fuzzy Logic Applications in Intelligent Systems", R. Yager and L. Zadeh, Eds. Boston: Kluwer, 1992, pp. 121-146.

[Khal, 1994] I. Khalil, B. M. Ali, A. R. Bidin, M. R. Mukerji and S. Ahmed, "A neural based call admission control in ATM networks to provide multiple QoS", Singapore ICCS'94, 14-18 Nov., pp.510-514, 1994.

[Kick, 1967] Kickert W.J.M. and Mamdani E.H., "Analysis of a Fuzzy Logic Controller", in *Readings in Fuzzy Sets for Intelligent Systems*, Edited by D. Dubois, H. Prade, Yager, R., Morgan Kaufmann Publ., pp. 290-297, 1967.

[Kilk, 1994] Kilkki L., “Traffic Characterisation and Connection Admission Control in ATM Networks”, Phd Thesis, Helsinki Univ. of Technology, Espoo 1994.

[Klir, 1988] G. Klir, T. Folger, "The role of uncertainty measures and principles in AI", in Fuzzy Sets Uncertainty and Information, Prentice Hall, 1988.

[Kosco, 1992] K. Kosko, "Neural Networks and Fuzzy Systems: a dynamical systems approach to Machine Intelligence", Prentice Hall International Editions, 1992.

[Kron, 1991] Kroner H., Heburterne G., Boyer P. and Gravey A., "Priority management in ATM switching nodes", *IEEE J. Select. Areas Commun.*, vol. 9, no.3, pp. 418-427, Apr. 1991.

[Lam, 1989] M.T. Lamata, S. Moral, “Classification of fuzzy measures”, *Fuzzy Sets and Systems* vol. 33, pp. 243-253, 1989.

[Lee, 1991] C.C. Lee, “A self-learning rule-based controller employing approximate reasoning and neural net concepts”, *Int. Journal of Intelligent Systems*, vol. 6, pp.71-93, 1991.

[LeeM, 1993] M.A. Lee, H. Takagi, “Integrating Design Stages of Fuzzy Systems using Genetic Algorithms”, Proceedings of an IEEE Conference, ISBN-0-7803-0614-7/93/$03.00, pp.612-617, 1993

[Lel, 1993] W.E. Leland et al., “On the Self-Similar Nature of Ethernet Traffic”, *Computer Communication Review*, ACM-SIGCOM, vol. 23, no. 4, October 1993.

[Liao, 1990] K.-Q. Liao and L.G. Mason, "A heuristic approach for performance analysis of ATM systems", *Proc. GLOBECOM'90*, San Diego, Dec. 1990, pp. 1931-1935.

[Lind, 1989] K.Lindberger, "Traffical analysis of statistical multiplexing in ATM networks", Contrib. to COST 224 and *8th Nordic Telletraffic Seminar*, Otnas, Finland, Aug. 1989.

[Louv, 1988] J.-R. Louvion, P. Boyer, A. Gravey, "A Discrete-time single server queue with Bernoulli arrivals and constant service time", ITC 12, Torino, June 1988.

[Lutt, 1984] J.L. Lutton, J.W. Roberts, "Traffic performance of multi-slot call routing strategies in an integrated service digital network", *ISS '84* Florence, Session 22.6.

[Magl, 1988] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, J. D. Robbins, “Performance Models of Statistical Multiplexing in Packet Video Communications”, *IEEE Transac. on Communications*, vol. 36, no. 7, pp. 834-843, July 1988.

[Magr, 1989] Magrez P., and Smets P., "Fuzzy modus ponens: a new model suitable for applications in knowledge-based systems", *International Journal of Intelligent Systems* 4, pp. 181-200, 1989.

[Mamd, 1981] E. Mamdani, B. Gaines, *Fuzzy Reasoning and its Applications*, Academic Press, 1992.

[Mamd, 1973] E.H. Mamdani and S. Assilian, "An experiment in Linguistic Synthesis with a Fuzzy Logic Controller", in *Readings in Fuzzy Sets for Intelligent Systems*, Edited by D. Dubois, H. Prade, Yager, R., Morgan Kaufmann Publ., pp. 283 289, 1973.

[Mand, 1985] N.J. Mandic, E.M. Scharf and E.H. Mamdani, " Practical Application of a Heuristic Fuzzy Rule-Based Controller to the Dynamic Control of a Robot Arm", Proc. IEE, Part D, Vol 132, No 4, July 1985.

[Marzo, 1993] J.L.Marzo, R. Fabregat, J. Domingo, J. Sole. "Fast Calculation of the CAC Convolution Algorithm, using the Multinomial Distribution Function". Tenth UK Teletraffic Symposium, pp.23/1-23/6, 1993.

[Mase, 1991] K. Mase and S. Shioda, "Real-Time Network Management for ATM Networks", in J.W. Cohen and C.D. Pack editors, *Queueing, Performance and Control in ATM* (ITC-13), Feb. 1991.

[Mich, 1991] Z. Michalewicz, G.A. Vignaux, M. Hobbs, "A non-standard Genetic Algorithm for the Nonlinear Transportation Problem", ORSA Journal on Computing, Vol. 3, no. 4, pp.307-316, 1991.

[Mich, 1992] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Springer Verlag, 1992.

[Mich, 1992a] Z. Michalewicz, C. Janikow, J. Krwczyk, "A modified Genetic Algorithm for Optimal Control problems", *Computers & Mathematics with Applications*, vol. 23, no. 12, pp. 83-94, 1992.

[Mich, 1992b] Z. Michalewicz, C. Janikow, "GENOCOP: A Genetic Algorithm for numerical optimisation problems with linear constraints", accepted for publication in the Communications of the ACM series, 1992.

[Mitr, 1992] N.M. Mitrou and K.P. Kontovasilis, "Fast analysis of on-off heterogeneous traffic multiplexing with fluid-flow models and its applications to ATM", *Proc. of the 2nd Race Workshop on traffic and perform. aspects in IBCN*, Aveiro, Portugal, Jan. 1992, session 2.

[Miyao, 1993] Y. Miyao, "Bandwidth allocation in ATM Networks that guarantee multiple QoS requirements", IEEE GLOBECOM'93, pp. 1398-1403, 1993.

[Moral, 1991] S. Moral and L.M. Campos, "Updating uncertain information", in: B.Bouchon-Meunier, R.R. Yager, L.A. Zadeh Eds, *Uncertainty in Knowledge Bases*, Lecture Notes in Computer Science 521, Springer Berlin, pp. 58-67, 1991.

[Mur., 1991a] T. Murase, H. Suzuki, S. Sato, T. Takeuchi, "A call admission control sheme for ATM networks using a simple quality estimate", *IEEE J. on Sel. areas in com.*, vol. 9, no. 9, Dec. 1991.

[Mur, 1991b] Murase T., Suzuki H. and Takeuchi T., "A call admission control for ATM networks based on individual multiplexed traffic characteristics", in proceedings IEEE ICC'91, 1991, paper 6.3.

[Ng, 1994] K. C. Ng, Y. Li, "Design of Sophisticated Fuzzy Logic Controllers Using genetic Algorithms", Proc. *3rd IEEE Int. Conference on Fuzzy Systems*, vol. 3, IEEE World Congress on Computational Intelligence, Orlando FL (1994) 1708-1712.

[Nguy, 1993] H. T. Nguyen, V. Kreinovich, D. Tolbert, "On robustness of Fuzzy Logics", Proceedings IEEE Conference, ISBN 0-7803-0614-7/93, pp. 543-547, 1993.

[Nish, 1992] T. Nishiyama, T. Takagi, R. Yager and S. Nakanishi, "Automatic generation of fuzzy inference rules by Genetic Algorithms", 8th Fuzzy System Symposium, 1992, pp. 237-240 (in Japanese).

[Niest, 1990] G. Niestedge, "The 'leaky bucket' policing method in the ATM Network", *Int. J. of Digital and Analogue Communications Systems*, vol. 3, 187-197, 1990.

[Nom, 1992a] Nomura H., Hayashi I. and Wakami N., "A learning method of fuzzy inference rules by descendent method", *Proc. of the IEEE International conference on Fuzzy systems*, pp. 203-210, 1992.

[Nom, 1992b] Nomura H., Hayashi I. and Wakami N., "A self-tuning method of fuzzy reasoning by Genetic Algorithm", *Proc. of the Int. Fuzzy Systems and Intelligent Control Conf.* (IFSICC'92), Louisville KY, pp. 236-245, 1992.

[Norros, 1991] Norros I., Roberts J.W., Simonian A. and Virtamo J.V., "The Superposition of Variable Bit Rate Sources in an ATM Multiplexer", *IEEE Journal on Selected Areas in Communication*s, vol. 9, no. 3, pp. 378-387, April 1991.

[Nov, 1993] V. Novak, “Logical Analysis of Max-Min Rule of Inference”, EUFIT’ 93, Aachen, Sept 7-10, pp.1311-1317.

[Onvur, 1994] Onvural R. O., “Asynchronous Transfer Mode Networks: Performance Issues”, Artech House, Boston, London, 1994.

[Park, 1992] D. Park, Z. Cao, A. Kandel, “Investigations on the applicability of fuzzy inference”, *Fuzzy Sets and Systems* 49, pp.151-169, 1992.

[Pedr, 1984] W. Pedrycz, “An identification algorithm in fuzzy relational systems”, *Fuzzy Sets and Systems* 13, pp. 153-167, 1984.

[Pedr, 1993] W. Pedrycz, “Fuzzy neural networks and neurocomputations”, *Fuzzy Sets and Systems*, vol. 56, pp.1-28, 1993.

[Pitts, 1993], Pitts J.M., "Cell-rate simulation modelling of Asynchronous Tranfer Mode Telecommunications Networks", *Ph.D. Thesis*, Queen Mary and Westfield College, July 1993.

[Raedt, 1992] L. Raedt, M. Bruynooghe, “A unifying framework for concept-learning algorithms”, *The Knowledge Engineering Review*, Vol. 7, no. 3, pp. 251-269, 1992.

[R1022 D.120, 1992] RACE 1022 TGIII, Del.120, “Updated results of the Traffic Simulation of the Policing Experiment”, Dec. 1990.

[R1022 D.126, 1992] RACE 1022 TGIII & IV, Del.126, "Final recommendation on Connection Admission Control, Usage Monitoring and Validation of control schemes", Nov. 1992.

[R2061 D.28, 1994] EXPLOIT Del. no. R2061/EXP/SW3/DS/P/028/B1, “Results of Experiments on Traffic Control using Real Applications”, Work Package 3.1, December 1994.

[Rama, 1990] G. Ramamurthy, B. Sengupta, “Modelling and Analysis of a Variable Bit Rate video multiplexer”, 7th ITC Specialist Seminar, Morristown, NJ, 1990.

[Rasm, 1991] Rasmussen C., Sorensen J.H. and Kvols K.S., "Source-Independent Call Acceptance Procedures in ATM Networks", *IEEE Journal on Selected Areas in Communication*s, vol. 9, no. 3, pp. 351-358, April 1991.

[Rath, 1989] E.P. Rathgeb, T.H. Theimer, "The policing functions in ATM networks, *RACE contrib. UST_123_0026_CD_CC*, June 1992.

[Rath, 1991] E.P. Rathgeb, "Modeling and comparison of policing mechanisms for ATM networks", *IEEE J. on Select. A. in Com.*, vol. 9, no. 3, April 1991.

[Rech, 1973] I. Rechenberg, Evolutionstrategie: Optimierung technischer Systems nach Prinzipien der biologischen Evolution, Frommann-Holzboog Verlag, Stuttgard, 1973.

[Redd, 1992] P. Venkata Subba Reddy and M. Syam Babu, “Some methods of reasoning for fuzzy conditional propositions”, *Fuzzy Sets and Systems* 52, pp. 229-250, 1992.

[Rob, 1988-89] J.W. Roberts, J.T. Virtamo, "The superposition of periodic cell arrival streams in an ATM Multiplexer", submitted to the *IEEE Trans. in Comm*.

[Roub, 1985] M. Roubens and Ph. Vincke, “Preference Modelling”, in Springer-Verlag, Berlin, 1895.

[Saito, 1989] H. Saito, "A simplified dimensioning method of ATM Networks", *IEICE tech. rep.* SSE89-112, 1989.

[--, 1990] --, "New dimensioning concept for ATM networks", presented at the *7th Int. Teletraffic Congr.*, Specialist Sem.., 1990.

[Saito, 1991a] Saito H., "Dynamic Call Admission Control in ATM Networks", *IEEE J. on Select. A. in Com.*, vol. 9, no.7, pp. 982-989, September 1991.

[Saito, 1991b] Saito H., M.Kawarasaki, and H.Yamada, "An analysis of statistical multiplexing in an ATM transport network", *IEEE J. on Select. A. in Com.*, vol. 9, no. 3, 1991.

[Saito, 1992] Saito H., "Call admission control in an ATM network using upper bound of cell loss probability", *IEEE Transactions on Communications*, Vol. 40, No. 9, September 1992.

[Saito, 1994] Saito H., "Teletraffic Technologies in ATM Networks", Artech House, Boston, London 1994.

[Sest, 1991] S. Sestito and T. Dillon, “Using single-layered neural networks for the extraction of conjunctive rules and hierarchical classifications”, *Journal Applied Intelligence*, vol. 1, pp.157-173, 1991.

[Scha, 1985] E.M. Scharf and N.J. Mandic, " The Application of a Fuzzy Controller to the Control of a Multi-Degree of Freedom Robot Arm ", in M. Sugeno (ed), Industrial Applications of Fuzzy Control, Elsevier Science Publications B.V., North Holland, Amsterdam, 1985.

[Schaf, 1989] J. Schaffer (Editor), Proceedings of the Third International Conference on Genetic Algorithms", Morgan Kaufmann Publishers, Los Altos, CA, 1989.

[Schw, 1981] H.-P. Schwefel, "Numerical Optimisation for Computer Models", Wiley, Chichester, U.K., 1981.

[Schw, 1977] H.-P. Schwefel, "Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie", volume 26 of Interdisciplinary systems research, Birkhäuser, Basel, 1977.

[Shaf, 1976] Shafer G., "A mathematical theory of evidence", Princeton University Press, Princeton, 1976.

[Stall, 1995] W. Stallings, "ISDN and B-ISDN with Frame Relay and ATM", Prentice-Hall, 1995.

[Sug, 1983] M.Sugeno, T. Takagi, "A new approach to design a fuzzy controller", in *Advances in Fuzzy Sets, Possibility Theory and Applications*, Wang, P.P. (Ed.), Plenum Press: New York, 325-334, 1983.

[Sug, 1991] Sugeno M. and Tanaka K., "Successive identification of a fuzzy model and its applications to prediction of a complex system", *Fuzzy Sets and Systems* 42, pp. 315-334, 1991.

[Sug, 1993] M.Sugeno, T. Yasukawa, "A Fuzzy Logic based approach to Qualitative Modelling", *IEEE Trans. Fuzzy Systems*, vol. 1, no. 1, January 1993.

[Surm, 1993] H. Surmann, A. Kanstein, K. Goser, "Self-organising and Genetic Algorithms for an Automatic Design of Fuzzy Control and Decision Systems",EUFIT'93, Aachen, pp. 1097-1104, 1993.

[Tans, 1988] Tanscheit R. and Scharf E M, "Experiments with the Use of a Rule-based Self-Organising Controller for Robotics Applications", *Fuzzy Sets and Systems*, Vol 26, No 2, 1988.

[Tak, 1985] T. Takagi, M.Sugeno, "Fuzzy identification of systems and its application to modeling and control", in *IEEE Trans. on Syst. Man and Cybernetics* 15, 116-132, 1985.

[Tak, 1991] T. Takagi, I. Hayashi, "NN-driven Fuzzy Reasoning", in *Int'l J. of Approximate Reasoning* (Special Issue of IIZUKA'88), Vol.5, No. 3, 1991, pp. 191-212.

[Taka, 1990] T. Takahashi, A. Hiramatsu, "Integrated ATM traffic control by distributed neural networks", , in *Proc. ISS '90*, Stockolm, Vol. 3, pp.59-65, 1990.

[Ter, 1992] T. Terano, K. Asai, M. Sugeno, *Fuzzy Systems Theory and its Apllications*, Academic Press, 1992.

[Thri, 1991] P. Thrift, "Fuzzy Logic Synthesis with Genetic Algorithms", in Proceedings of the International Conference on Genetic Algorithms, pp. 509-513, 1991.

[Turn, 1986] J.S. Turner, "New directions in communications (or which way to the information age)", *IEEE Commun. Mag.*, vol. 24, no.10, pp 8-15, 1986.

[Tran, 1991] P. Tran-Gia, O. Gropp, "Structure and performance of neural nets in broadband system admission control", Report no. 37, University of Wurzburg, Institute of Computer Sciences, *Research Report Series*, Dec. 1991.

[Yag, 1991] R. Yager, L. Zadeh, "An introduction to Fuzzy Logic Applications in Intelligent Systems", edited by Kluwer Academic Publishers, 1991.

[Yag, 1992] R. Yager, D. Filev, T. Sadeghi, "Analysis of Flexible Structured Fuzzy logic Controllers", Technical Report MII-I232, SMC 092-09-0916.

[Yam, 1992] N.Yamanaka, Y. Sato, K.Sato, "Performance limitation of Leaky Bucket algorithm for usage parameter control and bandwidth allocation methods", *IEICE Trans. on Comm.*, vol. E75-B, no. 2, Feb. 1992, pp. 82-86.

[Yang, 1995] T. Yang, D.H.K. Tsang, “A novel approach to estimating the Cell Loss Probability in an ATM Multiplexer loaded with Homogeneous On-Off sources”, *IEEE Trans. on Communications*, vol. 43, no.1, pp. 117-126, January 1995.

[Wood, 1988] G.M. Woodruff, R.G.H.Rogers and P.S.Richards, "A congestion control framework for high-speed integrated packetized transport", in *Proc. GLOBECOM'88*, 1988, pp. 7.1.1-7.1.5.

[Wors, 1992] T. Worster, "Neural network based Controllers for Connection Acceptance", *2nd RACE Workshop on Traffic and Performance Aspects in IBCN*, Aveiro, Portugal, Jan. 1992.

[Verd, 1984] J.L. Verdegay, “A dual approach to solve the Fuzzy Linear Programming problem”, *Fuzzy Sets and Systems* vol. 14, pp. 131-141, 1984.

[Vak, 1991] F. Vakil, H. Saito, "On congestion Control in ATM Networks", *IEEE LTS*, pp. 55-65, Aug. 1991.

[Zad, 1965] Zadeh L. A., "Fuzzy sets" in *Inf. Control*, vol. 8, pp. 338-353, 1965.

[Zad, 1973] Zadeh L. A., “Outline of a new approach to the analysis of complex systems and decision process”, IEEE Trans. In Systems, Man and Cybernetics” 3, pp. 159-176, 1993.

[Zad, 1978] Zadeh L.A., "Fuzzy sets as a basis for a theory of possibility", *Fuzzy Sets and Systems* 1, 3-28, 1978.

[Zad, 1979] Zadeh L. A., "A theory of approximate reasoning", in *Machine Intelligence* 9 (J. E. hayes, D. Michie and L.I. Mikulich, Eds.), Halstead Press, New York, 149-194, 1979.

[Zad, 1983] Zadeh L. A., "The role of fuzzy logic in the management of uncertainty in expert systems", *Fuzzy Sets and Systems* 11, 199-227, 1983.

[Zad, 1993] Zadeh L. A., "The role of fuzzy logic and soft computing in the conception and design of intelligent systems", *Fuzzy Logic in Artificial Intelligence*, 8th Austrian AI Conf. FLAI'93 Linz, Austria, June 1993.

[Zimm, 1987] H. Zimmermann, *Fuzzy sets, Decision Making and Expert Systems*, Kluwer Academic Publishers, 1987.

[Zimm, 1991] H. Zimmermann, *Fuzzy Set Theory - and Its Applications*, Kluwer Academic Publishers, Second Revised Edition, 1991.