國 立 交 通 大 學

電信工程學系

博 士 論 文

多媒體高速網路之服務品質保證式智慧型訊務控制機制

QoS-provisioning Traffic Control Schemes for Multimedia High-speed Networks Using Intelligent Techniques

研 究 生：林 立 峰

指導教授：張 仲 儒　　博士

中　華　民　國　九十五　年　七　月

多媒體高速網路之服務品質保證式智慧型訊務控制機制

QoS-provisioning Traffic Control Schemes for Multimedia High-speed Networks Using Intelligent Techniques

研究生：林立峰　　Student: Li-Fong Lin

指導教授：張仲儒　博士　　Advisor: Dr. Chung-Ju Chang

國立交通大學

電信工程學系

博士論文

A Dissertation
Submitted to Institute of Communication Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in
Communication Engineering
Hsinchu, Taiwan

2006 年 7 月

多媒體高速網路之服務品質保證式智慧型訊務控制機制

研究生：林立峰　　指導教授：張仲儒 博士

國立交通大學電信工程學系博士班

摘　要

為了支援多媒體服務之叢集性傳輸(bursty-transmission) 和異質性服務品質(quality of services, QoS) 之要求，我們需要一套精確設計的訊務控制(traffic control)機制，藉以在滿足連線頻寬與服務品質要求的狀況下，進一步有效地提升系統資源的使用效率。現有文獻已指出，多媒體服務其多樣化且多變的訊務特性、以及各式各樣的傳輸與服務品質上的需求，已經使得整體網路的行為更加複雜，因此有必要應用智慧型技術(intelligent techniques) 來解決多媒體高速網路環境中的訊務控制課題。在本論文中，我們探討在 ATM 和 IP 這兩種多媒體高速網路環境下，應用類神經/乏晰智慧型技術之訊務控制機制，並著重在「連線允諾控制 (connection admission control, CAC)」與「訊務監控調節(traffic policing)」這兩個主要的訊務控制功能。

我們首先探討在 ATM 網路中，採用「時域(time-domain)」訊務參數與類神經乏晰技術來進行連線允諾控制決策的「類神經乏晰連線允諾控制(neural fuzzy connection admission control, NFCAC)」方法。該允諾控制方法由於採用了類神經乏晰控制器做為控制決策的核心，因此也可以視為是一種整合性的技術：結合了乏晰邏輯系統的高階語意控制功能，以及類神經網路的自我學習能力。透過挑選適當的輸入信號（包含時域訊務參數與部分系統性能量測統計值）和設計較佳的控制法則，該允諾控制方法的類神經乏晰控制器即可以從中習得既有的連線允諾控制的專家知識，並提供一套快速、精確而有效的計算程序來實現足夠強健(robust) 的連線允諾控制功能。模擬結果顯示，與傳統的幾種（同樣採用時域訊務參數的）連線允諾控制機制相較起來，我們在此所提出的類神經乏晰連線允諾控制方法可以在滿足服務品質保證的基本條件下，達到最好的系統資源使用率；同時，若再與傳統方法中具有類神經網路學習功能的方法比較起來，此類神經乏晰連線允諾控制方法也有較快的學習收斂速度。

接著，在前述採用時域訊務參數的連線允諾控制方法之外，我們還另外探討採用「頻域(frequency-domain)」訊務參數與類神經網路技術來進行連線允諾控制決策的「功率頻譜基礎之類神經網路連線允諾控制(power-spectrum-based neural-net connection admission control, PNCAC)」方法。該允諾控制方法採用類神經網路控制器作為控制決策的核心，並以呼叫連線訊務源和既有連線訊務源的「功率頻譜密度(power spectral density, PSD) 參數」作為該控制器的（部分）輸入信號來進行連線允諾的決策控制。其理論基礎主要源自於，已知一個訊務源的功率頻譜密度函數(PSD function) 包含該訊務源的（時域）自我相關與叢集特性，並且已經有研究文獻證實，功率頻譜密度函數的確可以完全掌握並描述一個訊務源在一個佇列系統(queueing system) 的行為表現特性，因此其特徵參數(也就是前述簡稱的「功率頻譜密度參數」)

自然也就可以視為是該訊務源的一種訊務參數，並且可以對應到其一定的佇列行為特性。此研究發現，開啟了我們以訊務源的頻域訊務參數，亦即「功率頻譜密度參數」，來進行連線允諾控制的初始構想和概念。隨後，透過我們所提出的一套功率頻譜密度函數合成演算法，可以很快速地由兩個訊務源各自的功率頻譜密度參數，求得其合成訊務源的功率頻譜密度參數（的合理近似值），如此也使得採用「功率頻譜密度參數」來進行連線允諾控制成為實際可行且適合的方案。模擬結果顯示，我們所設計並提出採用「功率頻譜密度參數」的類神經網路連線允諾控制方法，可以在變動的網路訊務特性的環境下，達到相當穩定而強健的控制效能。

另外，我們在連線允諾控制機制的研究外，也對於和連線允諾機制息息相關且為其必備輔助功能的訊務監控調節機制進行相關的研究。我們首先針對 ATM 網路定義的訊務監控調節機制，也就是所謂的使用參數控制(usage parameter control, UPC) 功能，提出兩種智慧型使用參數控制器，分別是「乏晰使用參數控制器」以及「類神經乏晰使用參數控制器」。「乏晰使用參數控制器」是在傳統 ATM 標準所建議並用來實現使用參數控制功能的「漏水桶方法(leaky bucket algorithm)」上，引進一個「乏晰水增量控制器(fuzzy increment controller, FIC)」；同理，「類神經乏晰使用參數控制器」即是在傳統漏水桶方法上增加了一個「類神經乏晰水增量控制器(neural fuzzy increment controller, NFIC)」。這兩個智慧型控制器皆是選用相同的兩個系統量測統計值，即受監控訊務源之長期平均封包速率和短期平均封包速率，來做為其輸入語意變數，並據此對受監控訊務源做出適應性的動態水增量調整決策。模擬結果顯示，我們所提出的兩種智慧型使用參數控制器，皆可比傳統的漏水桶方法具有較高的違法封包檢出正確率、較短的訊務違法反應時間以及較快的違法封包檢出速率；對於合法連線於用戶終端設備輸出處的訊務塑型器(traffic shaper, TS) 所造成的佇列延遲上，也有比較小的數值表現。而在兩智慧型使用參數控制器方法彼此的比較方面，類神經乏晰使用參數控制器也在上述幾個方面的表現上比乏晰的方法有稍佳的效能，特別是在訊務違法反應時間及違法封包檢出速率上有較明顯的差異。

最後，我們繼續針對 IP 網路環境上的訊務監控調節機制提出最佳化設計。而 IP 網路的訊務監控調節機制，主要是以定義在差異化服務(differentiated services, DiffServ) 模型中的「訊務調節器(traffic conditioner)」所執行的訊務調節功能為主，因此我們便針對訊務調節器中的核心關鍵性組件－訊務封包標記器(traffic maker)，以網際網路標準組織 IETF 所建議的「雙速率三顏色封包標記器(two-rate-three-color-marker, TRTCM)」方法為基礎，提出一「效能增強型訊務封包標記器(enhanced traffic marker)」方法，在既有（消極）的監控性管制調節功能之外，再增加公平標記與積極性封包合法位階促升(promotion) 功能。模擬結果顯示，我們所設計的「效能增強型訊務封包標記器」的確可以在一合成訊務流中，對其各組成分流的封包合法位階標記做到精確檢出及公平標記的目的。同時，其不僅可以在系統資源足夠的時候，把過去曾經因為「局部(local)」網路壅塞或較嚴格的訊務合約(traffic contract) 導致合法位階被調降的封包再度提升至其原有的合法位階標記，還能夠進一步積極地盡可能提升其他封包的合法位階以享有較佳品質的封包處理動作，如此可在不違反訊務合約的狀況下，達到充分利用系統資源並增進受監控訊務連線的上層應用程式服務品質的目的。因此模擬結果也顯示，我們所設計的「效能增強型訊務封包標記器」比起傳統單純的「雙速率三顏色封包標記器」方法，在各合法位階的訊務上皆有較高（但仍合於訊務合約）的封包輸出速率(throughput)。

QoS-provisioning Traffic Control Schemes for Multimedia High-speed Networks Using Intelligent Techniques

Student: Li-Fong Lin Advisor: Dr. Chung-Ju Chang

Department of Communication Engineering
National Chiao Tung University

Abstract

To support bursty-transmission and heterogeneous quality of services (QoS) requirements for multimedia services, a suite of well-designed sophisticated traffic control scheme is required to effectively enhance the system utilization. The non-stationarity of work-loads, together with heterogeneous traffic characteristics and QoS constraints of multimedia services, indeed constitute the necessity for applying intelligent techniques in multimedia high-speed networks. In this dissertation, the traffic control functions involving the connection admission control (CAC) and the traffic policing for multimedia high-speed networks by neural/fuzzy intelligent techniques are studied. Both ATM and IP networks which can be utilized to construct the multimedia high-speed networks are considered in this dissertation.

Firstly, a *neural fuzzy connection admission control* (NFCAC) scheme which based on the *time-domain* traffic parameters and provides QoS guarantees for ATM networks is proposed. The NFCAC scheme is an integrated method that combines the linguistic control capabilities of a fuzzy logic controller and the learning abilities of a neural network. With properly choosing input variables which involve the measured statistics of network performances, and well designing the rule structure for the NFCAC scheme, it can not only provide a robust framework to mimic experts' knowledge embodied in existing connection admission

control techniques but can also construct precise and efficient computational algorithms for connection admission control. Simulation results show that, compared with conventional CAC schemes, the proposed NFCAC can achieve superior system utilization, high learning speed, and simple design procedure, while keeping the QoS contract.

And then, as contrast to the CAC scheme based on the time-domain traffic parameters discussed above, a *power-spectrum-based neural-net connection admission control* (PNCAC) scheme is proposed for also the multimedia high-speed ATM networks. It employs a neural network controller to handle the CAC function according to the *frequency-domain* power spectral density (PSD) parameters of the traffic sources. Since the PSD function of an input traffic contains the correlation and burstiness properties of the traffic, and it has been proven capable to characterize the queueing performances of the input traffic, the PSD parameters describing the PSD function can well correspond to the queueing performances also. With a composition algorithm to easily obtain the three PSD parameters of an aggregate traffic, it is suitable to adopting PSD parameters for CAC accordingly. Simulation results show that, after well training the neural network, an optimal CAC decision hyperplane based on the input variables is constructed to provide an efficient and robust admission control under dynamic network environments, while the QoS requirements are strictly assured.

After that, in addition to the studies about CAC, a traffic policing mechanism is necessary to ensure that all established connections conform to their respective traffic contracts, so that the CAC can perform correctly. Therefore, two intelligent usage parameter controllers are first proposed to implement the traffic policing function, usage parameter control (UPC), for multimedia transmissions in ATM networks. One is the *fuzzy usage parameter controller* realized by the fuzzy leaky bucket algorithm, in which a fuzzy increment controller (FIC) is incorporated with the conventional leaky bucket algorithm; the other is the *neural fuzzy usage parameter controller* base on the neural fuzzy leaky bucket algorithm, where a neural

fuzzy increment controller (NFIC) is added to the conventional leaky bucket algorithm. Both of FIC and NFIC properly choose the measured long-term and short-term mean cell rates, as input variables to adaptively determine the optimal increment value with respect to the traffic dynamics. Simulation results show that both intelligent leaky bucket algorithms have significantly outperformed the conventional leaky bucket algorithm, by higher selectivity and shorter responding time when taking control actions against a non-conforming connection, while reducing the queueing delay experienced by a conforming connection. Also, the neural fuzzy leaky bucket algorithm outperforms the fuzzy one in all aspects especially the responsiveness.

Finally, since the UPC is the traffic policing function defined in ATM networks, the traffic conditioner defined in the differentiated services (DiffServ) model is employed to handle the traffic policing function, namely the traffic conditioning, for the IP networks. An *enhanced traffic marker* (ETM) based on the Two-Rate-Three-Color-Marker (TRTCM) scheme is then proposed for the traffic conditioner to perform traffic policing by properly determining the conforming level of the incoming packet and making a corresponding color notation on the packet. The proposed ETM scheme introduces the features of aggressive promotion and fair share marking, and incorporates them into the existing traffic policing function. Simulation results show that the ETM scheme can fairly allocates the color notations among connections within an aggregate one. It also enhances the throughput of each conforming level for the aggregate connection to achieve as high rate as possible by not only restore the conforming levels of the previously demoted packets, but also aggressively promote the packets to higher conforming levels, so that the end-to-end QoS of the applications would be substantially improved while the traffic contract is still be respected. It can be concluded that the ETM scheme outperform the conventional TRTCM scheme in both aspects of marking fairness and traffic throughput of each conforming level under congested and under-loaded networks.

感言 與 誌謝
Acknowledgement

在這個實驗室經歷了十個寒暑，有歡笑也有淚水，不過好在的是歡樂的時光還是多於難過的時刻。在這十個年頭，感受了實驗室氣氛的轉變，從比較呆板的、純粹屬於（交大）男孩子的生活，到比較活潑而多采多姿的樣貌，外校系以及女生成員的加入，的確是這一股轉變的動力來源，為實驗室帶來很不一樣的朝氣和活力，而我也似乎隨著一屆屆碩士班學弟妹們的來來去去，而得以始終保持在大學剛畢業時的年輕 :)

十年了，不算短的一段時間，我彷彿伴隨著這個實驗室一起成長，又或者是說，實驗室帶著我、照顧著我茁壯。實驗室如同我的第二個家。從面向交大棒球場的這扇窗，我看到了春夏秋冬、晨昏午夜所有時刻的交大風景，也見證了一棟棟拔地而起的館舍蠶食鯨吞地佔領珍貴的校園綠地，而這原是莘莘學子和外來訪客最佳的休憩場所，內心覺得不捨卻又無可奈何，只希望能將這些景象永遠地烙印在腦海裡，永遠保持鮮綠明亮！而正如每個孩子在長大後都必須離開家庭獨立生活一般，此刻，也是我展開人生另一段旅程的時候了。感謝老師和整個實驗室這些年來的照顧和磨練，豐富了我的羽翼，也蓄積了我向前飛行的力量！

要感謝的人真的很多，特別是十年的光陰串起來的人事物，全部寫來，即便只有名字，也需要不少的篇幅。但在此，我還是想要把這些在每個階段同我一起經歷、分享實驗室一切的夥伴們的名字再寫一遍，並同時在腦海中再度細數與回憶我們曾一起經歷的時光。

首先最要衷心感謝的，自然是帶領我進入實驗室這個大家庭的恩師－張仲儒教授，除了在論文研究上的指導之外，其在生活和待人處事上的嚴謹態度與自我要求，也讓我在整個研究生涯中獲益良多，特別是幫助我看清自己的能力和障礙。或許我最終無法完全克服自己的障礙，但是可以更加認識自己，掌握自我特質的優缺點，也必將對於我的未來有很深遠的助益和影響，而我也真的很衷心感激老師對於我有時莫名而執拗的任性所給予的莫大包容。

再來要感謝瑞光、古博、界和、芳慶、義昇、勇志、又壬等幾位學長，每每不厭其煩地對於我的疑惑給予適切的引導和解答，在茅塞頓開的豁然開朗之外，除了提供我的論文許多寶貴的建議，並讓我學習到不少做研究的經驗和技巧，也開拓了我的生活視野和思想，明白一件事物是可以有很多面向與截然不同的風貌。此外，也感謝有宗勳、家慶、俊雄、騰元以及青毓幾位好同學的陪伴，互相鼓勵、加油打氣，才能夠讓我在論文研究的低潮期重新振作起精神和動力繼續走下去。另外還要感謝的是所有可愛的學弟妹們，文成、智勝、伯偉、俊賓、樹佑、宗益、尚逸、詠翰、嘉瑯、良正、鏗銘、崇光、慶喜、信宗、永宏、柏翰、照旗、易霖、玉葵、寧佑、至永、凱盟、駿元、崇禎、皓棠、俊憲、志明、朕逢、宗軒、立忠、凱元、同昊、家源、俊帆、文祥、煖玉、琴雅、建興、建安、佳璇、佳泓、世宏、正昕、尚樺，和你們一起從事的所有活動以及你們所帶來的歡樂，豐富了我的生活經驗，也讓我保持永遠年輕的心境和生活上的朝氣與活力。當然，也不

能忘記對總是默默在背後給予實驗室許多幫助的秀鷹、玉琇、雅雯和惟娟說聲謝謝，沒有妳們的貼心和幫忙，我可能沒辦法擺脫許多瑣事和煩人的行政工作的羈絆，實驗室（計畫）的運作也會因而多受阻礙而踟躕難前吧。最後，再次感謝所有實驗室的夥伴們，因為有你們大家的陪伴和幫助，充實了我人生中最精華的十年時光，我會帶著這些回憶和感動，化成源源不絕的動力，繼續往前走去！

謝謝～～也說聲～～再會了，我親愛的 701 實驗室～～～

最後，在實驗室之外要感謝的，自然是我最親愛的家人－特別是爸爸和媽媽，沒有他們的開明和耐心、安慰和鼓勵、體諒和包容，以及在信心和經濟上的支持，我是決不能走到現在這一步的。感謝一起成長的弟弟和妹妹，在我長年負笈在外的求學生涯裡幫我分擔許多為人子該有的責任。此外，也要謝謝女友芝宇一路體貼的陪伴和支持，讓我可以偶爾有機會跳脫研究上的煩悶枯燥或是牛角尖，再以一個全新的心情和精神來重新面對問題。而來自高中死黨裕忠和孟芳持續不斷的關心和鼓勵，也讓我倍感溫馨並永遠感謝在心。來自家人與好友的支持、鼓勵和祝福，永遠是最讓人感到窩心與幸福的一件事，也是我漫長的研究生涯中最強而有力的後盾，真的，很謝謝你們！

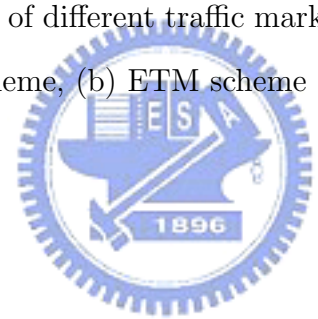
林立峰 謹誌

民國 九十五 年 九 月

（于交通大學電信系 ED701 寬頻網路實驗室）

Contents

List of Figures



List of Tables

Chapter 1

Introduction

1.1 Motivation

Over the past decades, the development of communication networks has been astonishing. So is the evolution of the service provisioning in the field of high-speed network. With the breakthrough of advanced semi-conductor and computer technologies, numerous transmission and networking techniques associated with broadband capability have gotten dramatic developments. Also, the costs for network usage are continually getting cheaper and thus more people enroll in the network due to the low cost and convenience. All kinds of applications or services have been developed over the high-speed communication networks. Most of them, especially the emerging ones, are the kind of *multimedia* services, which have various quality-of-service (QoS) and bandwidth requirements associated with the high-volume, high-burstiness and variable-rate traffics. Henceforth, high-bursty and high-volume services over high-speed networks are no longer scientifically fictional, but real. With the proliferation of various applications and the emergence of multimedia and real-time services, the high-speed network supporting multimedia services has to be capable of handling high-volume bursty traffic and providing guarantees to various QoS and bandwidth requirements. This is absolutely not an easy job because the abundant and diverse multimedia traffics have drastically

complicated the network environments. Henceforth, *sophisticated* and *efficient* (real-time) traffic control mechanisms are necessary to support diverse multimedia services/applications with different QoS and bandwidth requirements for high-speed networks while achieving high system utilization. Besides, the traffic control mechanisms are also required to be *adaptive* enough to handle the traffic and network dynamics.

Nowadays, both ATM (asynchronous transfer mode) and IP (Internet Protocol) technologies can be employed to implement multimedia high-speed networks to accommodate versatile services and the subsequent diverse traffic types and characteristics. In order to support a set of QoS classes sufficient for all feasible multimedia services, several traffic control mechanisms are proposed in both ATM and IP networks, such as connection admission control (CAC), traffic policing and shaping, congestion control, buffer management, (feedback) flow control, priority control, traffic identification/classification, and traffic scheduling. [31], [32], [50], [51]. Among these traffic control mechanisms, the dissertation concentrates on the studies of connection admission control (CAC) and traffic policing functions for ATM and IP network systems. As shown in Fig. 1.1, where a generic architecture of a network access node is illustrated and the conceptual operation locations of several traffic control mechanisms are also depicted, the CAC and traffic policing functions can be regarded as two critical mechanisms performing access control upon the network-incoming traffics.

Connection admission control (CAC) is performed in a network access node at the call setup phase and is defined as "a set of actions taken by the network in order to determine whether a connection can be accepted or not" [31]. Specifically, the set of actions taken by the network (in the access node) is to estimate the network resources against the requirements of the incoming connections. A new connection is accepted and allowed to begin its traffic transmission only if sufficient network resources are available and its required QoS can be afforded while the QoSs of existing connections can still be maintained. In addition, the

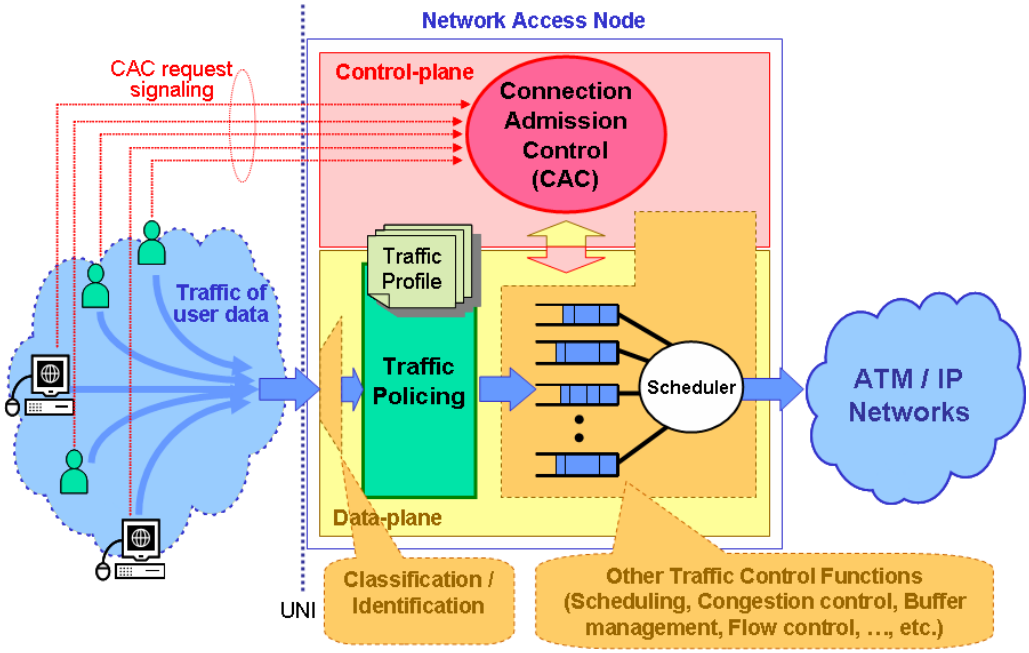


Figure 1.1: The generic architecture of a network access node

network utilization would also be expected to increase through CAC as high as possible. The challenge of CAC is the complexity because the characteristics of heterogeneous source are hard to precisely estimate. For CAC to perform correctly, all the established connections can not violate their respective traffic contracts.

To make sure that the established connections conform to their traffic contracts, a traffic policing mechanism should be employed. Traffic policing is performed at the user-network interface (UNI) during the data transfer phase, and is defined as the set of actions taken by the network to police the offered traffic of a connection so that the associated traffic contract is respected. That is, some portion of the traffic of a connection would be dropped or shaped (by introducing queueing effect) to enforce the resultant traffic compliant with the traffic profile negotiated in the traffic contract during the call setup phase. Sometimes, the non-conforming portion of a connection would be tagged rather than directly dropped, so that the residual traffic satisfy the contract and some future processing would be performed

upon the tagged non-conforming traffic to attain some operation objectives. The main purpose of the traffic policing function is to protect network resources from malicious as well as unintentional misbehavior which can affect the QoS of other already established connections. The wide variety of multimedia services with different traffic characteristics and QoS requirements makes traffic policing a difficult job. The difficulty lies in finding a simple, universal, and effective scheme which is able to police any type of traffic to meet its (usually long-term and call-level statistic-oriented) traffic contract by making (short-term and packet-level) processing decision upon each incoming packet.

Several approaches have been designed and proposed to deal with the traffic control problems. Most of the conventional approaches, which usually based on the analytic parametric models, suffer from serious shortcomings: some are simple but have many approximations and assumptions that are hard to justify, which makes those approaches impractical and leads to poor network resource utilizations because of the over- or under-estimations; others contain complicated mathematical solutions that may not be feasible to operate in real-time for high-speed multimedia networks. Besides, these conventional approaches provide optimal solutions only under a steady state with the assumption of a stationary system, and some parameters of the approaches are designed to be a pre-defined static value. This makes it difficult for the conventional approaches to handle the traffic control over non-stationarily dynamic network systems, because of not being able to react or respond to the highly varying network conditions.

Alternatively, some researchers turn to incorporate the measured system statistics and apply intelligent techniques to deal with the traffic control problems. Intelligent techniques, such as fuzzy logic inference systems, neural networks and neural fuzzy systems have been widely applied to deal with problems in numerous fields. They have replaced conventional technologies in many scientific applications and engineering systems including the network

control systems. Both fuzzy systems and neural networks are numerical model-free estimators and dynamical systems [1], which have the capability of modeling complex non-linear processes to arbitrary degrees of accuracy and efficiently adapting to the system dynamics. Recent research results have proven that these intelligent computations are capable of producing better results than parametric models or other conventional algorithmic approaches when applied to dynamic, non-linear complex systems. Researches have also shown that the non-stationarity of work-loads, together with heterogeneous traffic characteristics and QoS constraints of multimedia services, constitute the necessity for applying intelligent techniques in multimedia high-speed networks. Henceforth, in this dissertation, it is motivated to exploit the merits of intelligent techniques applying to traffic control schemes for multimedia high-speed networks.

1.2 Paper Survey

1.2.1 CAC using Time-domain Traffic Parameters

Two kinds of CAC schemes are discussed in this dissertation for ATM networks. They make the CAC control decision according to the traffic parameters of different aspects which are the *time-domain* [8]–[23] and *frequency-domain* [33]–[39] parameters, respectively. For the CAC schemes based on time-domain parameters, the conventional ones usually apply a parametric model of the traffic being offered, either by requiring each connection to provide an accurate description of its traffic behavior (via traffic parameters such as the peak rate, mean rate, and the peak rate duration), or by measuring the observed traffic and fitting it to a model, and then infers the cell loss ratio (CLR) (and other network performance measures) from this model. For this scheme, when a new connection is requested, the network examines either the required bandwidth [8], [9], [10], [11], [12] or the QoS requirements [13], [14] to decide whether to accept the new connection or not. In most of the approaches disclosed in

the literature, complicated mathematical equations were derived and approximations were required to meet the real-time operation requirement for CAC. Some conventional CAC schemes based on the time-domain parameters are briefly described below.

Guèrin, Ahmadi, and Naghshineh [8] proposed an "equivalent capacity" method for individual and multiplexed connections, based on their statistical characteristics defined by traffic parameters and the desired QoS. A unified metric is then obtained to represent the effective bandwidth used by connections and the corresponding effective load on network links. Although the paper can provide an exact approach to the computation of the equivalent capacity, the associated complexity makes it infeasible for real-time calculation. Hence, an approximation is introduced and it results in the degradation in utilization. Chang and Thomas [9], [10] used the large deviation theory and the Laplace method of integration to provide a simple intuitive overview of the recently developed theory of effective bandwidth for ATM networks. A simple priority scheme and a cut-off threshold scheme for implementing multiple QoS were discussed. Four parameters of the average rate, the asymptotic variance, the peak rate, and the average burst duration were employed as traffic descriptors for approximating the effective bandwidth functions. They introduced the use of envelope processes and conjugate processes that could be used for fast simulation and bounds. Another effective bandwidth approach is proposed by Elwalid, Mitra, *et al.* in [11], [12] for generic Markovian traffic sources rather than typical two-state on-off sources models. It also based on the large deviation theory and derive the effective bandwidth with an approximation techniques according to Chernoff's theorem. Fast and effective techniques for the computation of the approximation are given. The additive form in the effective bandwidth has simplifying consequences for connection admission problem with multiple heterogeneous classes of sources.

Saito [13] proposed a call admission scheme by inferring the upper bound of cell loss

probability from the traffic parameters specified by users (i.e. the maximum number of cells arriving during a fixed interval, and the average and variance of the number of cells arriving during a fixed interval). The QoS requirement is guaranteed to be satisfied under this control without assumptions of a cell arrival process. In [14], Murata *et al* modeled an ATM switch as a discrete-time single server queueing system, and an exact analysis is developed to obtain the waiting time distribution and cell loss probability for a new call and all existing calls. According to the results, how the network performances depend on the statistics of a new call (burstiness, sojourn time of a call in active or inactive state, etc.) is investigated, and the effectiveness of admission control and traffic smoothing is also demonstrated.

As noted in section 1.1, the conventional CAC approaches, based on the analytical parametric models, maybe simple but not feasible in practice. Many approximations and assumptions in these approaches simplify the models and this represents that networks are forced to make control decisions based on incomplete or imprecise information. Besides, the conventional CAC approaches provide optimal solutions only under a steady state with the assumptions of a stationary system. And some parameters of them are designed to be *static* values associated to the *a-priori* statistical knowledge. Henceforth, it is difficult to deal with the traffic control problems over modern and future communication networks, which are expected to be highly complicated and non-stationarily dynamic.

Some literatures [15]–[23] had proposed schemes to adopt a number of measured statistics of the network system to help provide a better or optimal traffic control decision. The statistics can be obtained by collecting network performance information such as the network load, the occupancy of the buffers, the data rate and the data loss ratio. All of them were proven to effectively and greatly improve the network performances. The reason lies on that the measured statistics of network performance indeed provide a more *insight* information of the system: the measured values represent the *real* conditions which can substitute

the parameters that base on *a-priori* knowledge in the model-based conventional schemes; the continual measurements of the network statistics would further provide the controller *adaptive* capabilities responding to network dynamics. Besides, the measurement forms a close-loop control system capable of adjusting itself to correspond to the network conditions, which would consequently provide stable and robust operations.

Additionally, more network statistics can be collected than what are needed in the model-based schemes to provide more *comprehensive* information about the system. These additional measurements may provide intuitive information about the traffic control. To full utilize those measurements, numerous model-free approaches based on the intelligent techniques are proposed. This is because the intelligent techniques could accommodate all information without any assumption about the systems. With the learning capability or the human experiences about the system, the intelligent techniques can extract the knowledge about the CAC and construct a robust admission controller. These approaches are briefly depicted as follows.

Fuzzy logic systems have been widely employed to deal with CAC-related problems in ATM networks [17], [18]. Bonde and Ghosh [17] used fuzzy mathematics to provide a flexible, high-performance solution to queue management in ATM networks. In [18], a fuzzy traffic controller which simultaneously incorporates CAC and congestion control was proposed. It is a fuzzy implementation of the two-threshold congestion control method and the equivalent capacity admission control method extensively studied in the literature. Comparative studies have shown that the proposed fuzzy approaches significantly improve system performance compared with conventional approaches.

Aamadi, Tarraf, Habib, and Saadawi [19] introduced intelligent traffic control for ATM networks. They surveyed some of the recent applications of NNs in high-speed networks. NNs could be used to measure and predict the traffic characteristics, to determine the acceptance

of connections as a CAC controller, to detect violation of negotiated parameters as a UPC controller, and to control the traffic flow via feedback control signal to prevent network from congestion. Performance results show that the NN approaches achieve better results, much simpler and faster than conventional approaches. Hiramatsu [20] proposed a neural-net based connection admission controller. The proposed ATM network controller used multilayer feedforward neural networks for learning the relations between the offered traffic and the service quality. In the proposed method, the declared traffic parameters were used only to divide calls into several bit-rate classes. The neural network in the controller actually learns the relationship between the numbers of existing connections in each bit-rate class and their corresponding QoSs according to the statistical characteristics of each bit-rate class. Morris and Samadi [21] described the application of neural networks to the CAC and the switch control problems. Key network performance parameters are observed while carrying various combinations of calls, and their relationship is learned by a neural network structure. The neural network model chosen has the ability to interpolate or extrapolate from the past-experienced results, and it also has the ability to adapt itself to the new and changing conditions. In [22], Youssef, Habib, and Saadawi proposed a call admission controller for ATM networks. A neural network is trained to compute the effective bandwidth required to support MPEG-1 VBR video calls with different QoS requirements. They showed that the adaptability of the neural network controller to new traffic situations had been achieved by adopting a hierarchical approach to the design. We have also proposed a neural network connection admission control (NNCAC) scheme [23] for ATM networks. Simulation results reveal that call admission control with neural networks can improve significantly system utilization, under QoS constraint.

1.2.2 CAC using Frequency-domain Traffic Parameters

All the studies about CAC schemes mentioned above were conducted mainly on the basis of traffic parameters in *time domain*. On the other hand, Li and Hwang [33] and Sheng and Li [34] have studied the queueing performance of a high-speed network from the point of view in the *frequency-domain traffic parameters*. The process of input traffic inherently contains a *power spectral density* (PSD) function, which is the Fourier transform of the input traffic process's autocorrelation function. From their studies, two characteristics of PSD are concluded: (i) The PSD can be well characterized by three main parameters such as the *DC component*, the *average power*, and the *half-power bandwidth*. (ii) The low-frequency band of the input PSD has a dominant impact on queueing performance, while the high-frequency band can be neglected to a large extent. This is because the low frequency component of PSD contains the correlation and burstiness of the input process. The more the low-frequency components are, the burstier the input traffic will be [35]. Therefore, according to the above two PSD characteristics from Li's studies, it can be conducted that these three PSD parameters can well characterize the input traffic and correspond to its queueing performances, and thus this reveals a chance to employ the PSD parameters for CAC.

A composition algorithm is proposed in [36] to obtain the three PSD parameters of an aggregate traffic source from the given PSD parameters of these individual traffic sources which build the aggregate one. The computation process of the composition algorithm is just through some simple arithmetic operations. It can then be concluded that PSD parameters possess *additive* property; this makes the PSD parameters more suitable for admission control, no matter how many types of traffic sources there are, because the PSD parameters of the virtually aggregated total traffic enrolling the new call could be easily estimated as the new call request arrives and maintain the same (three) reference variables, which can correspond to the queueing performances, for the admission control decision making. The

design of the CAC algorithm based on PSD parameters can be made accordingly and this indeed greatly reduce the complexity for admission control.

An intuitive and simple CAC method using PSD parameters, the power-spectrum-based table-lookup CAC method, was studied for multimedia communications in ATM networks, where the table content was the cell loss probability indexed by the PSD parameters of voice/video calls and arrival rates of data calls [36]. The table can be constructed through several explorative simulations. This method, according to the simulation results, is efficient enough, however, since the table is constructed based on the original three PSD parameters: DC component, half-power bandwidth, and average power [36], there is a drawback of large-dimensional CAC table. An "equivalent source" concept is consequently introduced to transform the PSD parameters of an offered traffic source into the so called "equivalent" PSD parameters which are corresponding to another (equivalent) traffic source [37]. The word "equivalent" exactly stands for almost the same queueing performances in some evaluation aspects. That is, the corresponding traffic source with the "equivalent" PSD parameters generated by the transformation is expected to have equivalent queueing performances with the offered traffic source characterized by original PSD parameters, so that the equivalent PSD parameters could substitute the original ones. A modified power-spectrum-based table-lookup CAC method was then proposed in [37] where the CAC lookup table is significantly reduced by one dimension than that proposed in [36], since the PSD parameters of each table entry in [36] can be transformed to the equivalent ones with the pre-defined half-power bandwidth value which is identical among all transformed entries, and thus only the DC component and the transformed equivalent average power have to be specified to characterize and distinguish each (voice/video) traffic source. The offered three PSD parameters of a new call request would also be transformed to the equivalent ones at first to adapt to the operations based on the dimension-reduced CAC table. Although the transformation may

introduce some degradations on performances, simulation results show that the modified power-spectrum-based table-lookup CAC scheme is still efficient enough and more feasible for practical implementations.

In order to get rid of the trade-off between efficiency and table size due to the quantization resolution of the PSD parameters indexing the CAC table, an enhanced power-spectrum-based CAC method based on the PSD parameters is designed by adopting the intelligent techniques to replace the lookup table and accommodate the index variables (including PSD parameters of voice/video calls and arrival rates of data calls) as the inputs for the intelligent controller. A continuous CAC decision hyperplane according to the input variables is then built to provide more precise admission control under the constraint of QoS requirements. Also, the learning capability of some intelligent techniques can bring adaptability to respond to the network dynamics. Two intelligent power-spectrum-based CAC schemes employing the neural network and the neural fuzzy controllers has been proposed in [38] and [39], respectively. Both of them further raise the performance improvements on network utilization of the power-spectrum-based CAC schemes as compared with the conventional equivalent capacity method [8].

1.2.3 Traffic Policing in ATM Networks

For CAC to perform correctly, a traffic policing mechanism is necessary to ensure that all established connections conform to their respective traffic contracts. Two traffic policing functions, the *usage parameter control* (UPC) [40]–[48] and *traffic conditioning* [49]–[56], are exploited in this dissertation for ATM and IP networks, respectively. The UPC is the traffic policing function defined in ATM networks [31], while the traffic policing in IP networks is performed through the traffic conditioning functions. For ATM networks, several UPC schemes such as the jumping window, triggered jumping window, moving window, exponentially weighted moving average, and leaky bucket algorithm were studied and compared [40],

[41], [42], [43]. The most popular and well-known policing scheme is the leaky bucket algorithm because of its simplicity and effectiveness. Three performance objectives have to be fulfilled by UPC and they can also be adopted as the criteria to evaluate the efficiency of the UPC in ATM networks: (i) *High selectivity (detection accuracy)*: UPC should detect and tag (drop) the non-conforming cells of a violating connection as many as possible, while being transparent when the connection conforms to its traffic contract. (ii) *High responsiveness*: the time for UPC to detect a violating connection should be rather short. (iii) *Low queueing delay*: cells of a non-violating connection should not experience too much queueing delay at the output shaper of customer premise equipment (CPE).

Some literature had also studied to utilize the intelligent techniques for the UPC [44], [45], [46], [47]. In [44], a fuzzy logic implementation of the leaky bucket algorithm that used a channel utilization feedback to manage voice cells in ATM networks was proposed. Simulation results showed that the fuzzy leaky bucket had performance improvement over the conventional leaky bucket algorithm. In [45], a neural network traffic enforcement mechanism using window-based scheme for ATM networks was presented. It is based upon an accurate estimation of the probability density function (pdf) of the traffic via a counting process, and the system performance is evaluated in terms of the pdf violation. It has scalability and convergence problems if the number of previous windows is required to be a large value. In [46], the paper designed a fuzzy policer based on window control scheme, which has the characteristic of simplicity and the capability to combine a fast responsiveness with a high-degree selectivity close to that of an ideal traffic policer. In [47], the proposed policing strategy integrated with a linear prediction filter is used to forecast the cell rate of the policed traffic source.

1.2.4 Traffic Policing in DiffServ IP Networks

The traffic policing function in IP networks is handled by a controller named as *traffic conditioner* defined in the Differentiated Services (DiffServ) model [50], [51]. The DiffServ model is a QoS-provisioning service architecture for traffic processing and delivery proposed by the Internet Engineering Task Force (IETF) [50] for the IP network, since the IP network is basically originated on the "best-effort" service model and can hardly provides QoS guarantees for any connection because the bandwidth resources are allocated in a competition sense among all connections. As contrary to the Integrated Services (IntServ) model [49] which is alternatively the other QoS-provisioning service model defined by IETF but has scalability problem because of the per-connection-based processing [51], DiffServ focuses on the QoS of the aggregate connections and supports only a set of finite number of predefined QoS classes in order to reduce the complexity and provide a promising solution to scalability. The connections that require a similar QoS level would be assigned to the same class, and thus (virtually) form an aggregate connection with a unique QoS processing including traffic conditioning.

The traffic conditioner, consisting of a meter, a marker and a shaper (or a dropper), would continually determine the conforming level of the incoming traffic of an aggregate connection according to the measured traffic flow and its traffic contract [50], [51]. After that, a notation would be made on the traffic packets by the *marker* to indicate the conforming level, and a corresponding processing action such as dropping, shaping and bypassing is then taken upon the packets. The packet notation assigned by the traffic marker in DiffServ networks is defined as three colors, denoted as green, yellow, and red, which are corresponding to three different pre-defined conforming levels for the packet with respect to the traffic contract. The packets assigned a green notation can be called as green packets for simplicity, and so do the packets marked a yellow or a red notation. The green packets stand for that these

packets belong to the best conforming level and have the lowest dropping precedence (or the shortest shaping delay); the red packets, on the contrary, represents that these packets are judged to be with the worst conforming level (e.g. the violation level) and have the highest dropping precedence (or the longest shaping delay).

In DiffServ IP networks, several traffic conditioning schemes such as Single-Rate-Three-Color-Marker (SRTCM) [52], Two-Rate-Three-Color-Marker (TRTCM) [53] and Time-Sliding-Window-Three-Color-Marker (TSWTCM) [54] were proposed in RFC to implement the traffic conditioner. The TRTCM, which is popular because of its simplicity and effectiveness, adopts a couple of token buckets to police two rate properties of a traffic source simultaneously. The output traffic rate of green packets as well as the aggregate output rate of green and yellow packets are both ensured individually to conform to the traffic profile, where the green traffic rate is usually corresponding to the policed mean (or sustainable) rate of the incoming traffic and the aggregated green and yellow traffic rate represents the policed peak rate of the incoming traffic.

In addition to the *color-blind* operation mode, where the color marking decisions are based on only the metering results against the traffic contract, the alternative *color-aware* operation mode of the TRTCM performs the color marking according to not only the metering results against the traffic contract, but also the existing color notation of the packets, simultaneously. The purpose and operation principle of the color-aware mode is to maintain the existing color notation of the policed packets as best as it can while still conforming to the traffic contract. This is because, as noted above, the color notation of the packets can represent the conforming level and correspond to the pre-defined QoS-provisioning packet processing behaviors. A packet may originally have its first color notation assigned by the output shaping function at the source node according to not only the metered results but also the *importance of the packet's content*. By properly allocating color notations represent-

ing higher conforming level and better QoS-provisioning packet processing behaviors to the packets with application critical contents, the QoS of each application is then expected to be quite improved while the traffic contract remains assured, since the packets with important application data are supported and served with better QoS. For example, the I-frame in the MPEG video is more vital than the other two coding frames, the B-frame and P-frame, because it serves as the base frame to reconstruct a series of video frames. The packets containing I-frame data can be assigned with the color notation representing higher conforming level and better QoS-provisioning packet processing behaviors so that the quality of the replayed video at the destination can be improved. Accordingly, the TRTCM operating in the color-aware mode can support better QoS for the applications than the TRTCM running in the color-blind mode.

As the TRTCM is a scheme to implement the traffic policing function in DiffServ IP networks, the packet *demotion* capability that re-marks a packet with a color notation corresponding to a lower conforming level than its existing one is inevitable and natural. However, a packet that is demoted due to occasionally short-term congestions or a locally stricter traffic profile may not have the chance to restore its existing or even the original conforming level. It has also been observed that the output rate of green packets might be impaired by the excessive incoming yellow packets: many packets with existing green notation are thus demoted to be with red color directly because the token resources are excessively consumed by the incoming yellow packets with the rate exceeding the traffic profile. These facts would result in the end-to-end QoS degradations for the applications since more packets carrying critical application data and originally denoted with a high conforming level maybe treated by worse packet processing behaviors due to the demotions. Also, the marking fairness among all connections within a (virtual) aggregate traffic is uncertain.

Similar performance objectives such as the selectivity, responsiveness and queueing delay

introduced in the UPC of ATM networks can also be employed to verify the efficiency of the traffic conditioner. In addition, because the processing of the traffic conditioner is based on the aggregate connection, the marking *fairness* for resource share among all connections within the (virtual) aggregate one could be taken into consideration as another performance objective. On the other hand, the IP network would be a world-wide network constituted by several interworking network systems which are hosted by different network service providers (NSPs). The network management policies of different NSPs may be varied and thus the definitions of a specific DiffServ QoS class can be distinct. Therefore, the traffic profile and the associated QoS-provisioning processing of an aggregate connection corresponding to the same QoS class may change from network domains to domains. As noted above in the TRTCM scheme, the packets might be demoted due to a locally stricter traffic profile and thus the end-to-end QoS of the applications would be degraded since the only demotion processing would make the traffic rate corresponding to the high conforming level decline along the communication route when the traffic traverse across several network hops or domains [55]. Consequently, a traffic promotion function is also considered as an objective for the traffic conditioner to not only restore the conforming levels of the previously demoted packets, but also aggressively promote the packets to higher conforming levels, if possible, for better application QoSs, while the traffic contract is still be respected. The aggressive promotion processing can then be equivalently regarded as fully utilizing the network resources to drive the traffic of each conforming level to achieve as high rate as possible by packet promotions while conforming to the traffic contract.

A random early demotion and promotion (REDP) technique [55] was proposed to overcome the unfair-marking problem. It implements a packet promotion function in addition to the demotion nature of the RED-In/Out (RIO) [56] marking mechanism, and achieves marking fairness by appropriately allocating the demotion/promotion probabilities among

packets during the packet demotion and promotion procedures. In order to fully utilize the network resources for better application QoSs and provide marking fairness among all connections within the (virtual) aggregate one for TRTCM, a TC_PFG marking scheme [57] was proposed. However, in TC_PFG, only the packets belonging to the yellow conforming level is allowed to be promoted and this limits its application. Moreover, TC_PFG has the problem of unjust-promotion that the previously demoted packets can not be guaranteed to be promoted first when the network resource condition is available to perform the packet promotion function.

1.3 Dissertation Organization

In this dissertation, the traffic control functions involving the connection admission control (CAC) and the traffic policing for multimedia high-speed networks by neural/fuzzy intelligent techniques are studied. Several types of service with different QoS requirements and various bandwidth demands have to be supported by the multimedia high-speed networks. Both ATM and IP networks which can be utilized to construct the multimedia high-speed networks are considered in this dissertation. The CAC schemes which make the admission control decisions according to the *time-domain* and *frequency-domain* traffic parameters are both discussed where the intelligent techniques are chosen to implement the CAC controllers. Also, the enhanced algorithms which implement the traffic policing function by incorporating the intelligent techniques and a elaborate computation procedure into existing algorithms for ATM and IP networks respectively are both well explored.

In Chapter 2, the basic concepts of fuzzy systems, neural networks, and integrated neural fuzzy systems are briefly reviewed. The architecture of a fuzzy inference system (FIS) and the most basic and popular fuzzy inference model to implement a fuzzy logic controller are stated. The neural networks and learning mechanism are presented along with two popular

architectures for implementing a neural network controller. The benefits of integrated neural fuzzy systems is described. Also a typical five-layer connectionist architecture to build a neural fuzzy controller are stated there. Additionally, the applications of these intelligent techniques to the traffic control functions over multimedia high-speed networks are given.

In Chapter 3, a neural fuzzy connection admission control (NFCAC) scheme which based on the time-domain traffic parameters and provides QoS guarantees for multimedia high-speed ATM networks is proposed. The NFCAC scheme adopts a neural fuzzy controller for admission control, which integrates the linguistic control capabilities of a fuzzy logic controller with the learning abilities of a neural network. We properly choose input variables which involves the measured statistics of network performances and the available network resources converted from the time-domain traffic parameters, and then well design the rule structure for the neural fuzzy controller. Accordingly, the NFCAC scheme can provide a robust framework to mimic experts' knowledge embodied in existing connection admission control techniques and can construct precise and efficient computational algorithms for connection admission control to achieve high system utilization while supporting QoS-guarantee.

In Chapter 4, a power-spectrum-based neural-net connection admission control (PNCAC) scheme for multimedia high-speed ATM networks is proposed. It employs a neural network controller to handle the connection admission control function according to the frequency-domain power spectral density (PSD) parameters of the traffic sources. With a composition algorithm to easily obtain the approximated three PSD parameters of the virtually aggregated total traffic enrolling the new call request, the neural network controller accommodate all the three PSD parameters as the inputs and generate the admission control decision. After well training the neural network, an optimal CAC decision hyperplane based on the input variables is constructed to provide an efficient and robust admission control even under dynamic network environments, while the QoS requirements are still satisfied and strictly as-

sured. Also, the learning capability of the neural network techniques can bring adaptability to respond to the network dynamics.

In Chapter 5, two intelligent usage parameter controllers are proposed to implement the traffic policing function for the sustainable-cell-rate (SCR) of multimedia transmissions in ATM networks. One is the fuzzy usage parameter controller realized by the fuzzy leaky bucket algorithm, in which a fuzzy increment controller (FIC) is incorporated with the conventional leaky bucket algorithm; the other is the neural fuzzy usage parameter controller base on the neural fuzzy leaky bucket algorithm, where a neural fuzzy increment controller (NFIC) is added to the conventional leaky bucket algorithm. The FIC and NFIC are exactly the fuzzy logic controller and the neural fuzzy controller, respectively, and both of them properly choose two measured statistics of the network performances, the long-term mean cell rate and the short-term mean cell rate, as the input variables to adaptively determine the optimal increment value with respect to the traffic dynamics. Accordingly, both of the proposed fuzzy and neural fuzzy usage parameter controllers can achieve better performances than the conventional leaky-bucket-based usage parameter controller because of the dynamic increment value by adaptive decisions.

In Chapter 6, an enhanced traffic marker (ETM) based on the Two-Rate-Three-Color-Marker (TRTCM) scheme is proposed for the traffic conditioner to perform traffic policing by properly determining the conforming level of the incoming packet and making a corresponding color notation on the packet. The proposed ETM scheme introduces the features of aggressive promotion and fair share marking, and incorporates them into the existing traffic policing function. One of the primary performance objectives is that it can fairly allocate the color notations among connections within an aggregate one. It is also anticipated to enhances the throughput of each conforming level for the aggregate connection to achieve as high rate as possible by not only restore the conforming levels of the previously demoted

packets, but also aggressively promote the packets to higher conforming levels if the network resource condition is available, so that the end-to-end QoS of the applications would be substantially improved while the traffic contract is still be respected. The performances of the proposed ETM scheme were verified via simulations and the simulation results were compared with the conventional TRTCM scheme.

Finally, some concluding remarks and future research topics are addressed in Chapter 7.

Chapter 2

An Overview of Intelligent Techniques

In this chapter, the basic concepts of fuzzy logic systems, neural networks, and integrated neural fuzzy systems are briefly reviewed. Fuzzy logic systems and neural networks are both numerical model-free estimators and dynamical systems [1], which have the capability of modeling complex nonlinear processes to arbitrary degrees of accuracy and efficiently adapting to the system dynamics. Also, the integrated neural fuzzy systems are integrating fuzzy systems and neural networks into a functional system to overcome their individual weaknesses by mutual compensation; that is, neural networks provide fuzzy systems with learning abilities and fuzzy systems provide neural networks with structural reasoning.

2.1 Introduction

In light of the recent developments of multimedia high-speed networks, future telecommunication networks will consists of heterogeneous access networks and comprise of content-rich services with diverse service characteristics and QoS requirements. Thus, the future multimedia high-speed networks will be highly dynamic communication environments, which require comprehensive and real-time traffic control mechanism. Traditional modelling and computation techniques are not well-suited to fulfill the requirements of future multimedia high-speed networks. On the other hand, intelligent techniques, such as fuzzy logic systems and neural networks, have attracted the numerous interests in various scientific and engineering areas. These intelligent techniques have the capabilities of soft-computing and adaptation, which are more flexible for network designers to cope with the network control problems. In this chapter, the concept of fuzzy logic system, neural network and neural fuzzy techniques will be briefly introduced.

Both fuzzy and neural network are mimicked the behaviors of human brain: fuzzy logic operates on the way the brain deals with vague information and neural networks are modelled according to the physical architecture of the brain [1]. There are a number of parallels that point out their similarities. Fuzzy systems and neural networks are both numerical model-free estimators and dynamical systems. Also, they have been shown to have the capability of modelling complex nonlinear processes to arbitrary degrees of accuracy. Although the two intelligent techniques are somewhat similar, some significant differences do exist. Fuzzy systems employ linguistic *if-then* fuzzy rules as a kind of expert knowledge to formalize insights about the structure of categories founding the real world. Fuzzy systems combine the mathematical theory of fuzzy sets with fuzzy rules to produce overall complex nonlinear behavior. On the other hand, neural networks are dynamical systems and are adaptively fitting the behavior of the real-world through their various connectionist structures and learning

techniques. Neural networks have a large number of highly interconnected processing elements (nodes or neurons) which demonstrate the ability to learn, recall and generalize from training patterns or data; these simple processing elements also collectively produce complex nonlinear behavior.

Alternatively, an innovative concept of interests of intelligent techniques is to merge or combine fuzzy systems and neural networks into a functional system to overcome their individual weaknesses. This innovative concept of integration reaps the benefits of both fuzzy systems and neural networks. That is, neural networks provide fuzzy systems with learning abilities, and fuzzy systems provide neural networks with a structural framework with high-level fuzzy *if-then* rule thinking and reasoning. Consequently, the two technologies can complement each other.

The rest of this chapter is organized as follows. The concept of fuzzy inference system (FIS) and the most basic and popular architectures of a fuzzy logic controller are stated in section 2.2. The neural networks and learning mechanism are presented in section 2.3, along with two popular architectures for implementing a neural network controller. In section 2.4, the concept of the integrated neural fuzzy system is described. Also a typical five-layer connectionist architecture to build a neural fuzzy controller are stated there. The applications of these intelligent techniques (such as fuzzy logic system, neural networks and integrated neural fuzzy system) in the following chapters of this dissertation are briefly previewed in section 2.5. Finally, the concluding remarks are given in section 2.6.

2.2 Fuzzy Logic Controller

2.2.1 Fuzzy Inference System (FIS)

Fuzzy logic is based on the concepts of linguistic variables and fuzzy sets theory. A *fuzzy set* in a universe of discourse U is characterized by a membership function $\mu(\cdot)$ which

takes values in the interval [0, 1]. A fuzzy set F is represented as a set of ordered pairs, each made up of a generic element $u \in U$ and its degree of membership $\mu(u)$. A *linguistic variable* x in a universe of discourse U is characterized by $T(x) = \{T_x^1, ..., T_x^i, ..., T_x^K\}$ and $M(x) = \{M_x^1(u), ..., M_x^i(u), ..., M_x^K(u)\}$, where $T(x)$ is the fuzzy term set, i.e., the set of linguistic values' names T_x^i the linguistic variable x can take, and $M_x^i(u)$ is a membership function with respect to the term T_x^i. If, for instance, x indicates the temperature, $T(x)$ could be the set as {*Low, Medium, High*}, and each element in $T(x)$ is associated with a membership function.

The *fuzzy inference system* (FIS) is a popular computing framework based on the concept of fuzzy logicand fuzzy reasoning. As shown in Fig. 2.1, a fuzzy inference system consists of four fundamental blocks [4]: *fuzzifier*, *fuzzy rule base*, *inference engine*, and *defuzzifier*. The fuzzifier performs a mapping function from the observed value of each input linguistic variable x_i to a fuzzy term set $T(x_i)$ with associated set of membership degree $M(x_i)$, $i = 1, \ldots, m$. The fuzzy rule base is a knowledge base characterized by a set of linguistic statements in a form of *"if-then"* rules that describe a fuzzy logic relationship between the m-dim input linguistic variables $\{x_i\}$ and the n-dim output linguistic variables $\{y_j\}$. The inference engine performs an implication function according to the pre-condition of the fuzzy rule with the input linguistic terms. It is a decision-making logic that acquires the input linguistic terms of $T(x_i)$ from the fuzzifier and uses an inference method to obtain the output linguistic terms of $T(y_j)$ [3]. The defuzzifier adopts a defuzzification function to convert $T(y_j)$ into a non-fuzzy (crisp) value that represents the decision y_j. Several implementation ways have been introduced to build a fuzzy inference system as a fuzzy logic controller, such as the *Mamdani fuzzy model*, *Tsukamoto fuzzy model*, and *Sugeno fuzzy model* [2]. Briefly speaking, these fuzzy models (or said implementation ways) differ on the high-level linguistic expression form of the fuzzy rule and the consequent reasoning way. Because the Mamdani fuzzy model

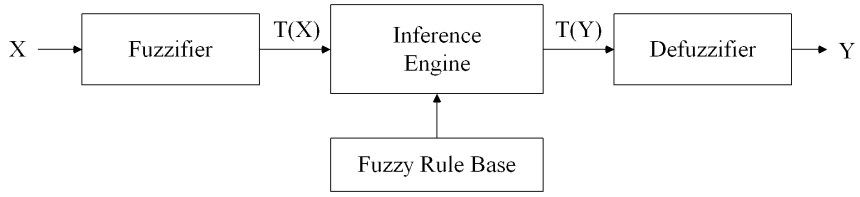


Figure 2.1: The basic structure of fuzzy inference system

is the most basic and popular one, some descriptions about the Mamdani fuzzy model are given in the following subsection.

2.2.2 Mamdani Fuzzy Model

The *Mamdani fuzzy model* is a way to implement a fuzzy inference system to serve as a

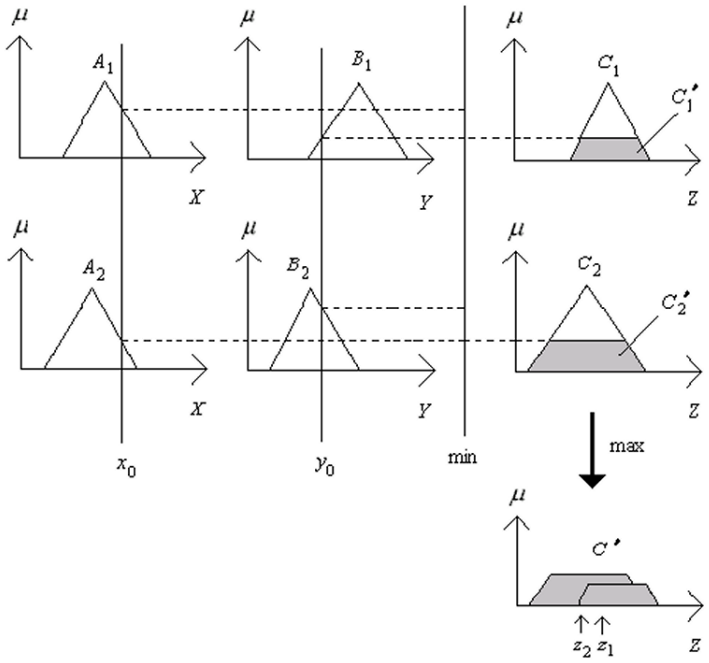


Figure 2.2: An example of Mamdani fuzzy model

controller. It was proposed as the first attempt to control a system by a set of linguistic control rules obtained from experienced human knowledge. Fig. 2.2 shows an example of

Mamdani fuzzy model, where the overall output Z is derived from two linguistic variables X and Y. Here, the fuzzy rule is expressed by

if X is A_i and **Y** is B_i, **then** output **Z** is C_i with $\mu(C_i)$, $\quad i$=1 and 2

where A_i, B_i and C_i are all fuzzy terms, and $\mu(C_i)$ is the membership value on C_i. In the Mamdani model, each input linguistic variable is firstly fuzzified by the membership function $\mu(\cdot)$. Then, the inferred value of the output of each fuzzy rule is determined by a pre-defined inference method. In this example, the *min-max* method is applied. That is, the inferred value of each fuzzy rule is obtained by *min* operator and the inferred value of the same fuzzy term is obtained by *max* operator. Finally, the overall crisp output is derived by a pre-defined defuzzification method. There are diverse defuzzification methods such as: centroid of area (COA), bisector of area (BOA), mean of maximum (MOM), smallest of maximum (SOM), and largest of maximum (LOM), among which COA is the most popular one.

Additionally, the membership functions for terms in the term set should be defined with the proper shape and position. In general, a triangular function $f(x; x_0, a_0, a_1)$ or a trapezoidal function $g(x; x_0, x_1, a_0, a_1)$ is chosen as the membership function because of the advantage of simple computational complexity. This feature makes these functions are suitable for real-time application [3]. As shown in Fig. 2.3, $f(x; x_0, a_0, a_1)$ and $g(x; x_0, x_1, a_0, a_1)$ are given by

$$f(x; x_0, a_0, a_1) = \begin{cases} \frac{x-x_0}{a_0} + 1 & \text{for } x_0 - a_0 < x \le x_0 \\ \frac{x_0-x}{a_1} + 1 & \text{for } x_0 < x \le x_0 + a_1 \\ 0 & \text{otherwise,} \end{cases} \tag{2.1}$$

$$g(x; x_0, x_1, a_0, a_1) = \begin{cases} \frac{x-x_0}{a_0} + 1 & \text{for } x_0 - a_0 < x \le x_0 \\ 1 & \text{for } x_0 < x \le x_1 \\ \frac{x_1-x_0}{a_1} & \text{for } x_1 < x \le x_1 + a_1 \\ 0 & \text{otherwise,} \end{cases} \tag{2.2}$$

where x_0 in $f(\cdot)$ is the center of the triangular function; x_0 (x_1) in $g(\cdot)$ is the left (right) edge of the trapezoidal function; and a_0 (a_1) is the left (right) width of the triangular or the trapezoidal function.

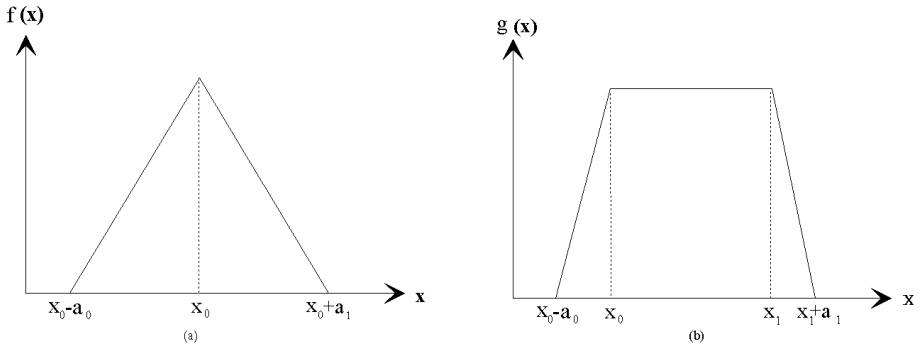


Figure 2.3: Definitions for functions $f(\cdot)$ and $g(\cdot)$

2.3 Neural Network Controller

2.3.1 Neural Networks and its Learning Capability

Neural networks are inspired by modeling networks of real (biological) neurons in the brain. It has a large number of highly interconnected processing elements which correspond to biological neurons and thus also be called *artificial neurons*, or simply *neurons*. The nodes are configured in regular architectures and can usually operate in parallel to make the whole network as a parallel distributed information processing structure. The collective behavior of an neural network, like a human brain, demonstrates the ability to learn, recall, and generalize from training pattern or data. The building blocks of neural network consists of three basic entities: *neurons model*, *connectionist structures* (among neurons) and *learning rules* [1]. Neurons are the basic information processing elements and can be viewed as consisting of two parts in the mathematical model: input part and output part. Associated with the input of a neuron is an *integration function* f which serves to combine information, activation, or evidence from an external source or other neurons into a *net input* to the neuron. The integration function f is usually a linear function of the input. A second action of each neuron is to output an *activation value* as a function of its net input through an *activation function* or *transfer function* $a(f)$. The step function, unipolar sigmoid function

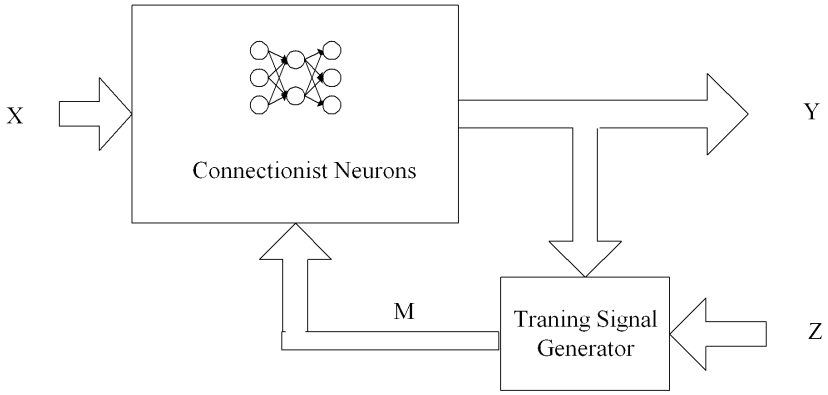


Figure 2.4: The basic structure of neural network

and bipolar sigmoid function are commonly used examples of the activation function. The connectionist structures are then applied to link neurons to mimic how the human brain works while the learning rules are applied to adaptively modify the behavior of the neural networks through past experience. Fig. 2.4 shows the basic concept of neural network. In the figure, X is the input signal, Y is the actual output, Z is the reference signal, and M is the training signal. The *connectionist neurons* block computes the output signal Y for input signal X and then the *training signal generator* block will generate a training signal according to a specified learning rules. The training signal is used to update the weighting of the nodes in the neural networks.

Generally speaking, the learning rules can be classified into three kinds of categories: *supervised learning*, *reinforcement learning*, and *unsupervised learning*. For different learning rules, there are different sets of Z and M. In the following, the main concepts of three learning rules are briefly described.

• Supervised Learning

In supervised learning, each input signal X has its own *desired* output D. Here, the reference signal Z is equal to desired output D. When the actual output Y is different from

reference signal Z, an error occurs. Then, the training signal will be generated to adjust the weighting of the nodes in the neural network such that the actual output will approach the reference signal. Therefore, the supervised learning can be considered as a input/output mapping machine or a function approximation tool.

• **Reinforcement Learning**

In the reinforcement learning, there is *no desired* output, only a reinforcement signal R. The reinforcement signal is an evaluation value of the actual output Y. For example, in the control problems, the reinforcement signal may be *"good"* or *"bad"*. Here, the reference signal Z is equal to reinforcement signal R. Using the reinforcement signal R, a training signal is generated to update the weighting such that the actual output will achieve a better evaluation value in the future. Therefore, the reinforcement learning is *learning with a teacher*. Using the reinforcement learning, the neural network acts as a controller to make the system work better according to a pre-defined evaluation function.

• **Unsupervised Learning**

Unlike the previous two learning rules, there is no feedback information from the environment in the unsupervised learning. Neither the desired output or reinforcement signal are available. Instead, the training signal is generated from actual output Y and the internal weighting of the neural network. The training signal here is used to increase the weighings of the nodes that connect to the actual output. That is, the correlation between the chosen input nodes and output data will be enhanced. In the unsupervised learning, the neural network discovers its patterns and the correlation through experiments, which is called *self-organizing*. Therefore, the unsupervised learning are usually applied to deal with the classification or clustering problems.

Like the condition in fuzzy logic controller, there are also diverse implementation ways

to build a neural network as a controller. Two of them which are popular and applied in this dissertation are described in the following subsections.

2.3.2 Multilayer Feedforward Neural Networks

Multilayer feedforward neural networks, as shown in Fig. 2.5, is a typical model for implementing a neural network controller. The neural network controller possesses an ability to perfectly approximate a generic function from input/output data pairs $\{X, Y\}$. Consider a multilayer feedforward neural network $\mathbf{NN}(X, W)$, with input vector X and a set of (link) weight vector W which will be updated by some learning rules; denote a continuous function by $Z = f(X) : D \subseteq R^{n_i} \rightarrow R^{n_o}$, where D is a compact metric space on R, and n_i (n_o) is the input (output) space dimension. The Stone-Weierstrass theorem [5] showed that $\mathbf{NN}(X, W)$ (actual output) can be trained to asymptotically approach any continuous desired output function $f(X)$ as close as possible. That is, an $\mathbf{NN}(X, W)$ with appropriate weight W can be found so that $||\mathbf{NN}(X, W) - f(X)||_X < \varepsilon$ for an arbitrary $\varepsilon > 0$, where $||e||_X = \sum_{X \in D} ||e(X)||^2$ and $|| \cdot ||$ is a vector norm. The neural network is a non-structured network, which cannot incorporate knowledge about system.

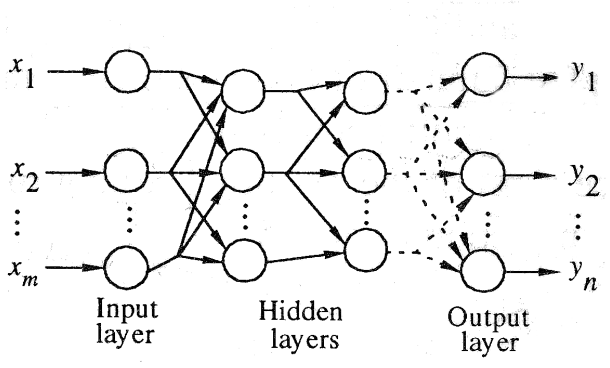


Figure 2.5: The structure of multilayer feedforward neural network

A back-propagation learning algorithm [6], which is a kind of supervised learning, is

usually employed to train the neural network controller. Let $X(i)$ denote the vector randomly sampled from D and used as an input to the neural network controller at time instant t_i, let $\mathbf{NN}(X(i), W) = \hat{z}(i)$ denote the corresponding decision of the neural network controller, and let $f(X(i)) = z(i)$ denote the desired decision. The objective of the back-propagation learning algorithm is to minimize decision error E by recursively adjusting its weight in each layer, where E is defined as

$$\begin{aligned} E &= \frac{1}{2}||NN(X(i), W) - f(X(i))||^2 \\ &= \frac{1}{2}(\hat{z}(i) - z(i))^2. \end{aligned} \tag{2.3}$$

Consider an M-layer feedforward neural network. Each layer has a number of processing elements (neurons) which are fully interconnected with the neurons in neighboring layers via adaptive weights. Neurons in the input layer (layer $k = 1$) do not process the input data; they simply store input data values. Neurons in the hidden layers ($2 \leq$ layer $k \leq M - 1$) and output layer (layer $k = M$) perform two operations. The j^{th} neuron in the k^{th} layer, for example, first calculates a weighted sum, denoted by $S_j^{(k)}$, of all outputs $o_i^{(k-1)}$ of the $(k-1)^{th}$ layer. $S_j^{(k)}$ is given by

$$S_j^{(k)} = \begin{cases} x_j & \text{if } \ k = 1, \\ \sum_{i=1}^{n_{k-1}} w_{ji}^{(k)} o_i^{(k-1)} & \text{if } \ 2 \leq k \leq M, \end{cases} \tag{2.4}$$

where x_j is the input variable of the j^{th} neuron in the input layer, n_{k-1} is the number of neurons in layer $(k-1)$, and $w_{ji}^{(k)}$ is the weight of the link connected from the i^{th} neuron in layer $(k-1)$ to the j^{th} neuron in layer k. After that, the neuron further transforms $S_j^{(k)}$ into output $o_j^{(k)}$ via an activation function $G(\cdot)$. $o_j^{(k)}$ is expressed as

$$o_j^{(k)} = \begin{cases} S_j^{(k)} & \text{if } \ k = 1, \\ G(S_j^{(k)}) & \text{if } \ 2 \leq k \leq M. \end{cases} \tag{2.5}$$

The adjustment of weights is based on a steepest-descent algorithm [6]. It can be ex-

pressed as

$$w_{ji}^{(k),new} = w_{ji}^{(k),old} - \eta \frac{\partial E}{\partial w_{ji}^{(k)}}|_{w_{ji}^{(k)}=w_{ji}^{(k),old}}, \tag{2.6}$$

where η is a gain term that determines the learning rate of the link weight. η is usually set equal to a positive constant less than unity. In order to obtain the partial derivative for the quadratic error E, an error term produced by the j^{th} neuron in layer k, denoted by $\delta_j^{(k)}$, is obtained from

$$\delta_j^{(k)} = -\frac{\partial E}{\partial S_j^{(k)}}, 1 \leq k \leq M, 1 \leq j \leq n_k. \tag{2.7}$$

It was shown in [6] that the error signals $\delta_j^{(k)}$'s can be computed according to a recursive procedure of the generalized delta learning rule [6] described as follows,

$$\delta_j^{(k)} = \begin{cases} G'(S_j^{(k)}) \sum_l \delta_l^{(k+1)} w_{lj}^{(k+1)} & \text{for } \ 2 \leq k \leq M-1, \\ (z - \hat{z}) G'(S_j^{(k)}) & \text{for } \ k = M. \end{cases} \tag{2.8}$$

Once these error signal terms have been determined, the partial derivative for the quadratic error can be computed directly by

$$\frac{\partial E}{\partial w_{ji}^{(k)}} = \frac{\partial E}{\partial S_j^{(k)}} \frac{\partial S_j^{(k)}}{\partial w_{ji}^{(k)}} = -\delta_j^{(k)} o_i^{(k-1)}. \tag{2.9}$$

And the update rule for the back-propagation algorithm is then given by

$$w_{ji}^{(k),new} = w_{ji}^{(k),old} - \eta \frac{\partial E}{\partial w_{ji}^{(k)}} = w_{ji}^{(k),old} + \eta \delta_j^{(k)} o_i^{(k-1)}. \tag{2.10}$$

2.3.3 Radial Basis Function Neural Networks

The radial basis function neural networks (RBFN), which was suggested by Moody and Darken in [7], is another implementation of the neural network to serve as a controller. It has the architecture of the instar-outstar neural network model and uses the hybrid unsupervised and supervised learning scheme. It offers a viable alternative to the two-layer neural network in many applications of signal processing, pattern recognition, control, and function

approximation. The structure of RBFN is showed in Fig. 2.6. Unlike the instar-outstar neural network model in which the hidden nodes are linear winner-take-all nodes, the hidden nodes in the RBFN have normalized Gaussian activation function

$$z_q = g_q(\mathbf{x}) = \frac{R_q(\mathbf{x})}{\sum_k R_k(\mathbf{x})} = \frac{\exp[-\frac{|\mathbf{x}-\mathbf{m}_q|^2}{2\sigma_q^2}]}{\sum_k \exp[-\frac{|\mathbf{x}-\mathbf{m}_k|^2}{2\sigma_k^2}]}, \tag{2.11}$$

where $\mathbf{x}$ is the input vector. Thus, hidden node q gives a maximum response to input vectors close to $\mathbf{m}_q$. Each hidden node q is said to have its own receptive filed $R_q(\mathbf{x})$ in the input space, which is a region centered on $\mathbf{m}_q$ with size proportional to σ_q, where $\mathbf{m}_q$ and σ_q are the mean (an m-dimensional vector) and variance of the qth Gaussian function. The Gaussian function is a particular example of radial basis functions. The output of the RBFN, is denoted by y, is simply the weighted sum of the hidden node output, which is given by

$$y = a(\sum_{q=1}^{l} w_q z_q + \theta), \tag{2.12}$$

where $a(\cdot)$ is the output activation function and θ is the threshold value. Generally, $a(\cdot)$ is an identity function (i.e., the output node is a linear unit) and $\theta = 0$.

The purpose of the RBFN is to pave the input space with overlapping receptive fields. For an input vector $\mathbf{x}$ lying somewhere in the input space, the receptive fields with centers close to it will be appreciably activated. The output of the RBFN is then the weighted sum of the activation of these receptive fields.

The training rule of RBFN is hybrid. It includes unsupervised learning in the input layer and supervised learning in the output layer. The unsupervised part of the learning involves the determination of the receptive field centers $\mathbf{m}_q$ and widths σ_q, $q = 1, 2, \cdots, l$. The proper centers $\mathbf{m}_q$ can be found by unsupervised learning rules such as the vector quantization approach, competitive learning rules, or simply the Kohonen learning rule; that is

$$\Delta\mathbf{m}_{\text{closest}} = \eta(\mathbf{x} - \mathbf{m}_{\text{closest}}), \tag{2.13}$$

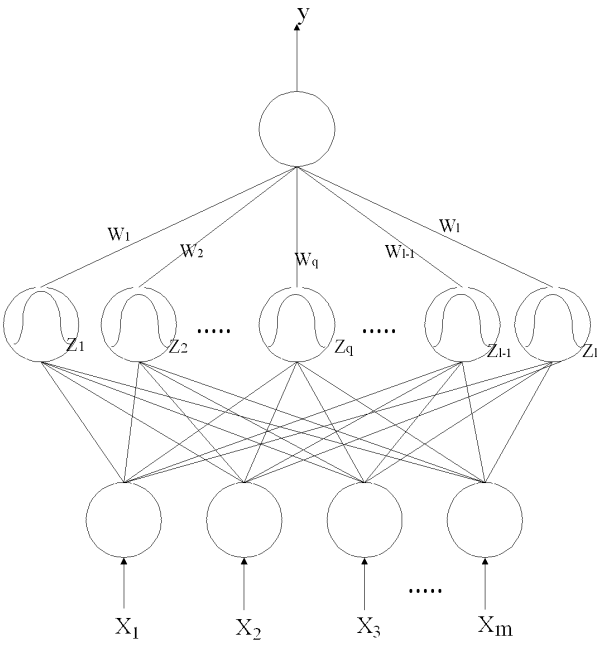


Figure 2.6: The structure of RBFN controller

where $\mathbf{m}_{\text{closest}}$ is the center of the receptive field closest to the input vector x and the other centers are kept unchanged. In simple case, the widths σ_q are determined by

$$\sigma_q = \frac{|\mathbf{m}_q - \mathbf{m}_{\text{closest}}|}{\gamma}, \tag{2.14}$$

where $\mathbf{m}_{\text{closest}}$ is the closest vector to $\mathbf{m}_q$ and γ is an overlap parameter.

According to the delta learning rule, the weights in the output layer can be updated by

$$\Delta w_q = \eta(d - y)z_q. \tag{2.15}$$

When averaged over the p training pairs, the objective is to minimize the following mean squared error cost function:

$$E(w_q) = \frac{1}{2}\sum_k [d^k - y^k]^2 \tag{2.16}$$

$$= \frac{1}{2}\sum_k [d^k - \sum_{q=1}^{l} w_q z_q^k]^2 \tag{2.17}$$

$$= \frac{1}{2}\sum_{k}[d^k - \sum_{q=1}^{l} w_q g_q(\mathbf{x}^k)]^2. \tag{2.18}$$

Although RBFN generally cannot quite achieve the same accuracy as the multilayer feedforward neural network, it can be trained several orders of magnitude faster than the the multilayer feedforward neural network with back-propagation learning. This is due to the advantage of hybrid-learning networks which have only one layer of connections trained by supervised learning. It is suitable for the application where the neural network controller is necessary to be on-line trained to adaptively capture the dynamic features of a system.

2.4 Neural Fuzzy Controller

2.4.1 Integrated Neural Fuzzy Systems

In the field of intelligent techniques, the fuzzy system and neural network are complementary techniques. Fuzzy systems provide a high-interpretable reasoning for the collected data, but the design of the fuzzy rules and the membership functions are not easy tasks which require much domain knowledge. Neural networks, on the other hand, are effective and efficient computing architectures or algorithms with self-learning capability, but the connectivity of hidden nodes of the neural network are somewhat like grey boxes. Thus, it is a promising approach to merge and integrate them into a single system. The integration of the two techniques can be classified into two categories: *neuro-fuzzy system* and fuzzy neural system.

The basic concept of a neuro-fuzzy system is to *use the neural network as tool in a fuzzy model*. The neuro-fuzzy systems can provide the self-learning (automatic tuning) capability for the fuzzy systems. In this approach, the system is firstly designed as a fuzzy inference system based on designers' domain knowledge. Then, via numerous experiments, the fuzzy rules and membership functions are tuned by the neural network. The whole design process would be simplified and the development time would be reduced consequently.

The basic concept of a fuzzy neural system is to *fuzzify the conventional neural network models*. In the fuzzy neural system, the basic properties and node connectivity of neural network are retained, but the operations and activation functions of the nodes are fuzzified. In this approach, a network's domain knowledge becomes formalized in terms of fuzzy sets, later being applied to enhance the learning of the network in such a way that it learns the mapping between input-output fuzzy sets. Generally speaking, the benefits of the fuzzy neural systems are three-folded: firstly, the input nodes are continuous-valued by fuzzification; secondly, the domain knowledge is applied; and thirdly, some degree of uncertainty of the collected data is allowed.

The learning rules introduced in neural networks should also be applied to the neural fuzzy systems. In general, the learning capability of a neural fuzzy systems would make itself achieve the expected performance objectives by adjusting the membership function. Therefore, the membership functions utilized for the terms in the neural fuzzy systems should be defined to be a proper formula which is suitable for the learning operation. That is, the shape and position of the membership function can be easily and well characterized by the parameters of moderate number. Meanwhile, the formula of the membership function is also usually required to be differentiable because of the usual differentiation computation in the learning process, especially the supervised learning. A general example of the membership function for the neural fuzzy controller could be the bell-shaped function defined as

$$f(x) = exp\left[-\frac{(x-m)^2}{\sigma^2}\right] \tag{2.19}$$

where m and σ are, respectively, the center (or mean) and the width (or variance) of the bell-shaped function.

2.4.2 A Typical Five-layer Neural Fuzzy Controller

Fig. 2.7 demonstrates a typical five-layer connectionist architecture to implement the neural fuzzy system as a neural fuzzy controller. It is a structured neural network that can incorporate domain knowledge about the system of the application field. Therefore, the five-layer neural fuzzy controller is a kind of fuzzy neural system. The nodes in layer one and layer five are *input* and *output linguistic nodes*, respectively. There are two kinds of output linguistic nodes: one is for feeding training data (desired output) into the net and the other is for pumping decision signals (actual output) out of the net. The nodes in layer two and layer four are *term nodes* which act as membership functions of the respective linguistic variables. The nodes in layer three are *rule nodes*; each node represents one fuzzy rule and all nodes form a fuzzy rule base. The links in layer three and layer four, accompanied by the nodes in both layers, can function as an inference engine — layer-three links define preconditions of the rule nodes and layer-four links define consequences of the rule nodes. The links in layer two and layer five are fully connected between the linguistic nodes and their corresponding term nodes. They can, accompanied by the nodes in both layers, achieve the fuzzification and defuzzification functions, respectively.

Generally, node i in layer k for the neural fuzzy controller has input function $f_i^{(k)}(u_{ij}^{(k)})$ and activation output function $a_i^{(k)}(f_i^{(k)})$, where $u_{ij}^{(k)}$ denotes the input to node i in layer k from node j in layer $(k-1)$ and $u_{ij}^{(k)}$ is expressed as $u_i^{(k)}$ if $k = 1$. For the five-layer neural fuzzy controller with M-dim input and single output, the detail operations of each layer are described as follows.

Layer 1: In this layer, there are M input nodes with respective to M input linguistic variables x_i, for $i = 1, \ldots, M$. Define

$$f_i^{(1)}(u_i^{(1)}) = u_i^{(1)} \text{ and } a_i^{(1)} = f_i^{(1)}, \tag{2.20}$$

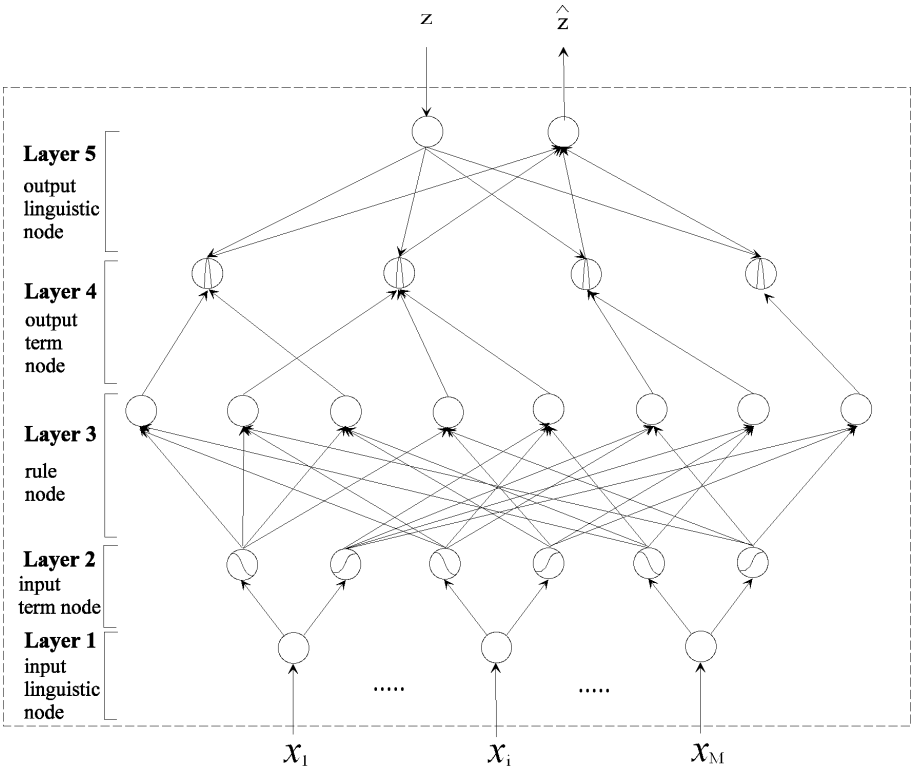


Figure 2.7: The architecture of the five-layer neural fuzzy controller

where $u_i^{(1)} = x_i$ and $1 \leq i \leq M$.

Layer 2: The nodes in this layer are respectively corresponding to a linguistic term of the input linguistic variables. Hence, these nodes are usually named as "term node" and perform the fuzzification function to map the crisp input into a fuzzy membership value according to its associated membership function. Assume K nodes exists in this layer. Each node performs a bell-shaped function defined as

$$f_i^{(2)}(u_{ij}^{(2)}) = exp\left[-\frac{(u_{ij}^{(2)} - m_{jn}^{(I)})^2}{{\sigma_{jn}^{(I)}}^2}\right] \quad \text{and} \quad a_i^{(2)} = e^{f_i^{(2)}}, \tag{2.21}$$

where $u_{ij}^{(2)} = a_j^{(1)}$, $1 \leq i \leq K$, $1 \leq j \leq M$, and $m_{jn}^{(I)}$ and $\sigma_{jn}^{(I)}$ are the mean and the standard deviation of the n-th term of the input linguistic variable from node

j in input layer, respectively.

Layer 3: The nodes in layer three are so-called "rule nodes" and each of them represents one fuzzy rule, respectively. All nodes in layer three form a fuzzy rule base. The links in this layer perform precondition matching of fuzzy control rules. Assume there are R rule nodes in this layer. Each rule node performs the fuzzy AND operation defined as

$$f_i^{(3)}(u_{ij}^{(3)}) = min(u_{ij}^{(3)}; \forall j \in P_i) \quad \text{and} \quad a_i^{(3)} = f_i^{(3)}, \tag{2.22}$$

where $u_{ij}^{(3)} = a_j^{(2)}$ and $P_i = \{j|$ all j that are precondition nodes of the i-th rule$\}$, $1 \leq i \leq R$.

Layer 4: The nodes in this layer have two operating modes: *down-up* and *up-down*. In the down-up operating mode, the links perform consequence matching of fuzzy control rules. Assume there are N term nodes in this layer. Each node performs a fuzzy OR operation to integrate the fired strength of rules that have the same consequence. Thus, we define

$$f_i^{(4)}(u_{ij}^{(4)}) = max(u_{ij}^{(4)}; \forall j \in C_i) \quad \text{and} \quad a_i^{(4)} = f_i^{(4)}, \tag{2.23}$$

where $u_{ij}^{(4)} = a_j^{(3)}$ and $C_i = \{j|$ all j that have the same consequence of the i-th term in the term set of $\hat{z}\}$, $1 \leq i \leq N$. The up-down operating mode is used during the training period. The nodes in this layer and the links in layer five have functions similar to those in layer two. Each node is named as "term node" and performs a bell-shaped function defined as

$$f_i^{(4)}(u_{ij}^{(4)}) = exp\left[-\frac{(u_{ij}^{(4)} - m_j^{(O)})^2}{{\sigma_j^{(O)}}^2}\right] \tag{2.24}$$

$$\text{and} \quad a_i^{(4)} = e^{f_i^{(4)}},$$

where $u_{ij}^{(4)}$ is set to be $a_j^{(5)}$ obtained from the up-down operating nodes in layer five, and $m_j^{(O)}$ and $\sigma_j^{(O)}$ are the mean and the standard deviation of the j-th term of $\hat{z}$, respectively, $1 \leq i \leq N$, $j = 1$.

Layer 5: There are two nodes in this layer. One node performs the down-up operation for the actual decision signal $\hat{z}$. The node and its links act as the defuzzifier. The function used to simulate a center-of-area defuzzification method is approximated by

$$f_i^{(5)}(u_{ij}^{(5)}) = \sum_{j=1}^{4} m_j^{(O)} \sigma_j^{(O)} u_{ij}^{(5)} \tag{2.25}$$

$$\text{and} \quad a_i^{(5)} = U\left(\frac{f_i^{(5)}}{\sum_{j=1}^{4} \sigma_j^{(O)} u_{ij}^{(5)}} - z_a\right)$$

where $u_{ij}^{(5)} = a_j^{(4)}$, $i = 1$, z_a is the decision threshold, and

$$U(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.26}$$

Clearly, $\hat{z} = a_1^{(5)}$ and a new connection will be accepted only if $\hat{z} = 1$. The other node performs the up-down operation during the training period. It feeds the desired decision signal z into the controller to adjust the link weights optimally. For this kind of node,

$$f_i^{(5)}(u_{ij}^{(5)}) = u_{ii}^{(5)}, \text{ and } \quad a_i^{(5)} = f_i^{(5)} \tag{2.27}$$

where $i = j = 1$ and $u_{11}^{(5)} = z$.

2.5 Applications of Intelligent Techniques In This Dissertation

In Chapter 3, neural fuzzy connection admission control (NFCAC) scheme is proposed and a kind of neural fuzzy controller, called NFCAC controller, is applied to deal with the traffic

control problem in multimedia high-speed ATM networks. With properly choosing input variables and designing the rule structure, the proposed NFCAC scheme not only provides a robust framework to mimic experts' knowledge embodied in existing traffic control techniques but also constructs intelligent computational algorithm for traffic control.

In Chapter 4, a power-spectrum-based neural-net connection admission control (PNCAC) scheme is proposed and a neural network controller, called PNCAC controller, is employed to perform the connection admission control for multimedia networks. The PNCAC controller has the learning/adapting capabilities so that the boundary of the decision hyperplane for the connection admission control can be adjusted optimally and dynamically. Simulation results show that the proposed PNCAC scheme enhances significantly the system utilization while fulfilling QoS requirements.

In Chapter 5, two intelligent techniques which are the fuzzy logic systems and neural fuzzy networks, are introduced to implement two intelligent increment controllers, fuzzy increment controller (FIC) and neural fuzzy increment controller (NFIC), respectively, for the usage parameter control (UPC) of multimedia transmission in ATM networks. Both of these two intelligent increment controller are to be incorporated with the conventional leaky bucket algorithm, which is a reference model for UPC function defined in ITU-T Recommendation I.371 [31]. Both the fuzzy and the neural fuzzy increment controllers properly choose the long-term mean cell rate and the short-term mean cell rate as input variables to optimally determine the increment value. Simulation results show that both intelligent leaky bucket algorithms have significantly outperformances over conventional leaky bucket algorithm, in aspects of all considered performance measures such as selectivity, responsiveness, and queueing delay.

2.6 Concluding Remarks

This chapter provides a fundamental overview of the neural/fuzzy techniques, including fuzzy systems, neural networks, and the neural fuzzy systems. Both the fuzzy systems and neural networks are mimicked the behaviors of human brains, where the neural network processes the low-level data clustering, classification, and mapping and the fuzzy system processes the high-level reasoning of the input data. The fuzzy systems and neural networks are complementary techniques. It would be beneficial to integrate the two techniques, which contributes to the rising of the integrated neural fuzzy systems.

In the multimedia high-speed networks, the network operations and performance statistics are chronically collected and stored. These data provide insight and meaningful information for the traffic control schemes. However, with the emerging of future content-rich services, the operation of the multimedia high-speed networks tend to be more dynamic and some service scenario may be far beyond imagination. It is almost impossible to design a comprehensive traffic control schemes in advance. Therefore, the intelligent techniques are promising approaches for the traffic control schemes because they possess the capability to extract the basic operation rules from the past records or experiences, and adaptively modify the operation rules of traffic control schemes according to the network dynamics. In the following three chapters, the intelligent techniques including fuzzy logic systems, neural networks and neural fuzzy controllers, are applied to design precise and sophisticated traffic control schemes for multimedia high-speed networks.

Chapter 3

Intelligent Connection Admission Control Scheme for Multimedia High-speed Networks Using Time-domain Traffic Parameters

This chapter proposes a neural fuzzy approach for connection admission control (CAC) with QoS guarantee in multimedia high-speed ATM networks according to the time-domain traffic parameters. The proposed neural fuzzy connection admission control (NFCAC) scheme is an integrated method that combines the linguistic control capabilities of a fuzzy logic controller and the learning abilities of a neural network. With properly choosing input variables which involve the measured statistics of network performances, and well designing the rule structure for the NFCAC scheme, it can not only provide a robust framework to mimic experts' knowledge embodied in existing connection admission control techniques but can also construct precise and efficient computational algorithms for connection admission control. Simulation results show that as compared with a conventional effective-bandwidth-based CAC, a fuzzy-logic-based CAC, and a neural-net-based CAC, the proposed NFCAC can achieve superior system utilization, high learning speed, and simple design procedure, while keeping the QoS contract.

3.1 Introduction

High-speed network supporting multimedia services has to be capable of handling bursty traffic and satisfying various quality-of-service (QoS) and bandwidth requirements. Therefore, a multimedia high-speed network must have an appropriate connection admission control (CAC) scheme not only to guarantee QoS for existing calls but also to achieve high system utilization. As known, ATM (asynchronous transfer mode) is one of the technologies that can integrate multimedia services for high-speed networks.

Conventional CAC schemes [8], [13], [10], [11], [12], [15], [16] that utilize either capacity estimation or buffer thresholds suffer from some fundamental limitations. One of the limitations is the difficulty of obtaining complete statistics on input traffic to a network. As a result, it is not easy to accurately determine the equivalent capacity or effective thresholds for multimedia high-speed networks in various bursty traffic flow conditions. Besides, these conventional schemes provide optimal solutions only under a steady state. A control scheme that dynamically regulates traffic flows according to changing network conditions, however, requires understanding of network dynamics. The rationale and principles underlying the nature and choice of thresholds or equivalent capacity under dynamic conditions are unclear [14]. Networks are forced to make decisions based on incomplete information [14] so that the decision process is full of uncertainty. Thus, because of unpredictable statistical fluctuations of the system, these control schemes will always be subject to decision error, which degrades performance.

Fuzzy logic systems have been widely employed to deal with CAC-related problems in ATM networks [17], [18]. Fuzzy set theory appears to provide a robust mathematical framework for dealing with real-world imprecision, and the fuzzy approach exhibits a soft behavior which means to have a greater ability to adapt itself to dynamic, imprecise, and bursty environments [17], [18]. Bonde and Ghosh [17] used fuzzy mathematics to provide a flexible,

high-performance solution to queue management in ATM networks. In [18], a fuzzy traffic controller which simultaneously incorporates CAC and congestion control was proposed. It is a fuzzy implementation of the two-threshold congestion control method and the equivalent capacity admission control method extensively studied in the literature. Comparative studies have shown that the proposed fuzzy approaches significantly improve system performance compared with conventional approaches. However, no clear, general technique has been presented to map existing knowledge on traffic control onto the design parameters of the fuzzy logic controller. Self-learning capability should be incorporated into the fuzzy logic controller to simplify the design procedure and obtain better control results.

The self-learning capability of neural networks has been applied to characterize the relationship between input traffic and system performance [19], [20], [22]. In [20], Hiramatsu used a neural network as a connection admission controller. In [22], Youssef, Habib, and Saadawi proposed a call admission controller for ATM networks. A neural network is trained to compute the effective bandwidth required to support MPEG-1 VBR video calls with different QoS requirements. They showed that the adaptability of the neural network controller to new traffic situations had been achieved by adopting a hierarchical approach to the design. However, in most of the proposed neural-net approaches for CAC, the numbers of users for each kind of service were selected as input parameters. The dimension of neural network and the learning time would increase as the number of traffic types grows. The system complexity would increase for system upgrade. Therefore, the application of neural network to CAC is limited to a simplistic traffic environment, such as limited traffic type, simplified traffic source, etc.

Conventional, fuzzy-logic-based, and neural-net-based CAC schemes all have various benefits in handling CAC. Conventional CAC, based on mathematical analysis, provides robust solutions for different kinds of traffic environments but suffers from estimation error (due to

modelling) and approximation error (due to the need to complete calculations in real time), so is not suitable for dynamic environments. Fuzzy-logic-based CAC is excellent in dealing with real-world imprecision and has a greater ability to adapt itself to dynamic, imprecise, and bursty environments, but lacks the learning capability needed to automatically construct its rule structure and membership functions so as to achieve optimal performance. Neural-net-based CAC provides learning and adaptation capabilities which reduce the estimation error of conventional CAC and achieve performance similar to that of a fuzzy logic controller. However, the knowledge embodied in conventional methods is difficult to incorporate into the design of a neural network.

This chapter proposes a *neural fuzzy connection admission control* (NFCAC) scheme, which absorbs benefits of the three approaches while minimizing their drawbacks, for multimedia high-speed networks. The NFCAC scheme utilizes the learning capability of the neural network to reduce decision errors of conventional CAC policies resulted from modeling, approximation, and unpredictable traffic fluctuations of the system. It also employs the rule structure of the fuzzy logic controller to prevent operating errors, due to incorrect learning, and to decrease training time. Furthermore, the neural fuzzy network is a simple structured network. We here properly choose input variables and design the rule structure for the NFCAC scheme so that it not only provides a robust framework to mimic experts' knowledge embodied in existing connection admission control techniques but also constructs intelligent computational algorithm for connection admission control to achieve high system utilization while supporting QoS-guarantee. Simulation results reveal that the NFCAC scheme achieves superior system utilization and high learning speed while keeping the QoS contract, compared with the effective-bandwidth-based CAC (EBCAC) [10], the fuzzy-logic-based CAC (FLCAC) [18], the neural-net-based CAC (NNCAC) [23], and the radial-basis-function-based CAC (RBFCAC) schemes.

The rest of this chapter is organized as follows. In Section 3.2, the basic concepts behind a neural fuzzy controller are introduced, and an NFCAC scheme is proposed to cope with CAC-related problems in multimedia high-speed networks. Section 3.3 presents simulation results comparing the proposed NFCAC scheme with the existing effective bandwidth approach, the fuzzy logic approach, and neural-net approach. Finally, some concluding remarks are given in Section 3.4.

3.2 Neural Fuzzy Call Admission Control

3.2.1 System Model

Fig. 3.1 shows an NFCAC controller with its peripheral processors to handle the call

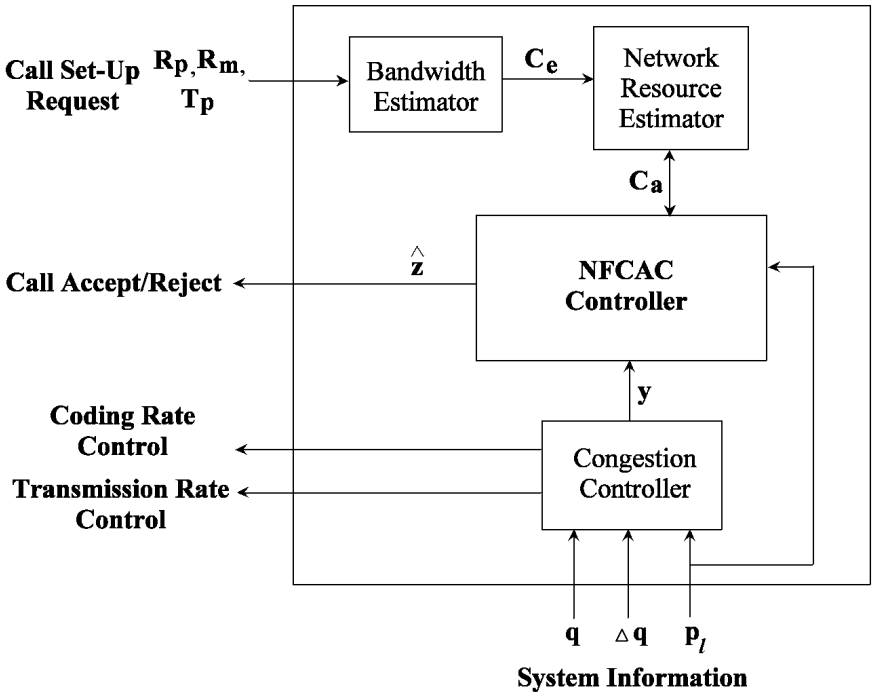


Figure 3.1: An NFCAC controller with its peripheral processors

admission control and traffic rate control simultaneously for multimedia high-speed networks. The *congestion controller* generates a congestion indicator y according to the measured

system statistics, such as the queue length q, the change rate of the queue length Δq, and the cell loss ratio p_l. Different congestion control algorithms could be employed to implement the congestion controller. For example, rate-based feedback congestion control approaches, which commonly take the queue length and the cell loss ratio into account, could be used. One of the most frequently used congestion control methods is the buffer threshold method, where a congestion alarm occurs whenever the queue length exceeds some predefined thresholds. Here, we adopt a fuzzy congestion controller [18] which is a fuzzy implementation of the two-threshold congestion control scheme proposed in [28]. Network congestion is then averted by regulating the traffic flow of the incoming sources according to the traffic load adjustment parameter generated by the fuzzy congestion controller. The *bandwidth estimator* estimates the required capacity C_e for a new connection from its traffic description parameters such as the peak cell rate, sustainable cell rate, and peak cell rate duration, denoted by R_p, R_m, and T_p, respectively. It employs the equivalent-capacity-based algorithm proposed in the literature. The equivalent capacity method [8, Eq. (2)] transforms the traffic characteristics (usually described by three traffic parameters: peak cell rate, sustainable cell rate, and peak cell rate duration) of a new call into a unified metric, called the equivalent bandwidth, to reduce the dependence of the proposed control mechanism on the traffic type. Such a transformation can greatly reduce the number of dimensions of the NFCAC scheme and save a large percentage of learning time. Here, we adopt a fuzzy bandwidth estimator [18], which is a fuzzy implementation of the equivalent capacity method in [8]. The *network resource estimator* does the accounting for system-resource usage. When a new connection with bandwidth C_e is accepted, the value of C_a is updated by subtracting C_e from the original value of C_a. Conversely, when an existing connection with bandwidth C_e is disconnected, the value of C_a is updated by adding C_e to the original value of C_a. C_a is initially set to 1. The *NFCAC controller* takes the available

capacity C_a, the congestion indicator y, and the system performance feedback of cell loss ratio p_l as input linguistic variables to handle the CAC procedure and sends a decision signal $\hat{z}$ back to the new connection to indicate acceptance or rejection of the new call request.

3.2.2 NFCAC Controller

The NFCAC controller is a neural fuzzy controller which is a control system that integrates a fuzzy logic system with a neural network. The integration brings the low-level learning and computational power of the neural network into the fuzzy logic system, and provides the high-level, human-like thinking and reasoning of fuzzy logic system for the neural network. The neural fuzzy controller is generally implemented by taking the form of a multi-layer neural network to incorporate the fuzzy logic system [1].

Fig. 3.2 demonstrates the architecture of the proposed NFCAC controller which is implemented by the five-layer neural fuzzy controller introduced in section 2.4. The NFCAC controller adopts three linguistic inputs of an available capacity C_a, a congestion indicator y, and a cell loss ratio p_l and outputs a decision signal $\hat{z}$ to indicate acceptance or rejection of the new call request. As mentioned above, the nodes in the second layer are the term nodes and they would co-operate with the links of the same layer to act as the fuzzifier. According to the CAC methods in [8][18], the term used to describe the remaining capacity available for a new connection is either "Enough" or "Not Enough." Thus the term set for the available capacity is defined as $T(C_a) = \{$*Not Enough (NE)*, *Enough (E)*$\}$. The system is either in a congested state ("y is Negative") or a congestion-free state ("y is Positive"), so the term set for the congestion status can be defined as $T(y) = \{$*Negative (N)*, *Positive (P)*$\}$. The term used to describe the cell loss ratio, which is one of the dominant QoS requirements, is either "Satisfied" or "Not Satisfied." Thus the term set for the cell loss ratio is defined as $T(p_l) = \{$*Satisfied (S)*, *Not Satisfied (NS)*$\}$. In summary, we have six nodes in layer two with respect to the six terms defined by the three input linguistic variables C_a,

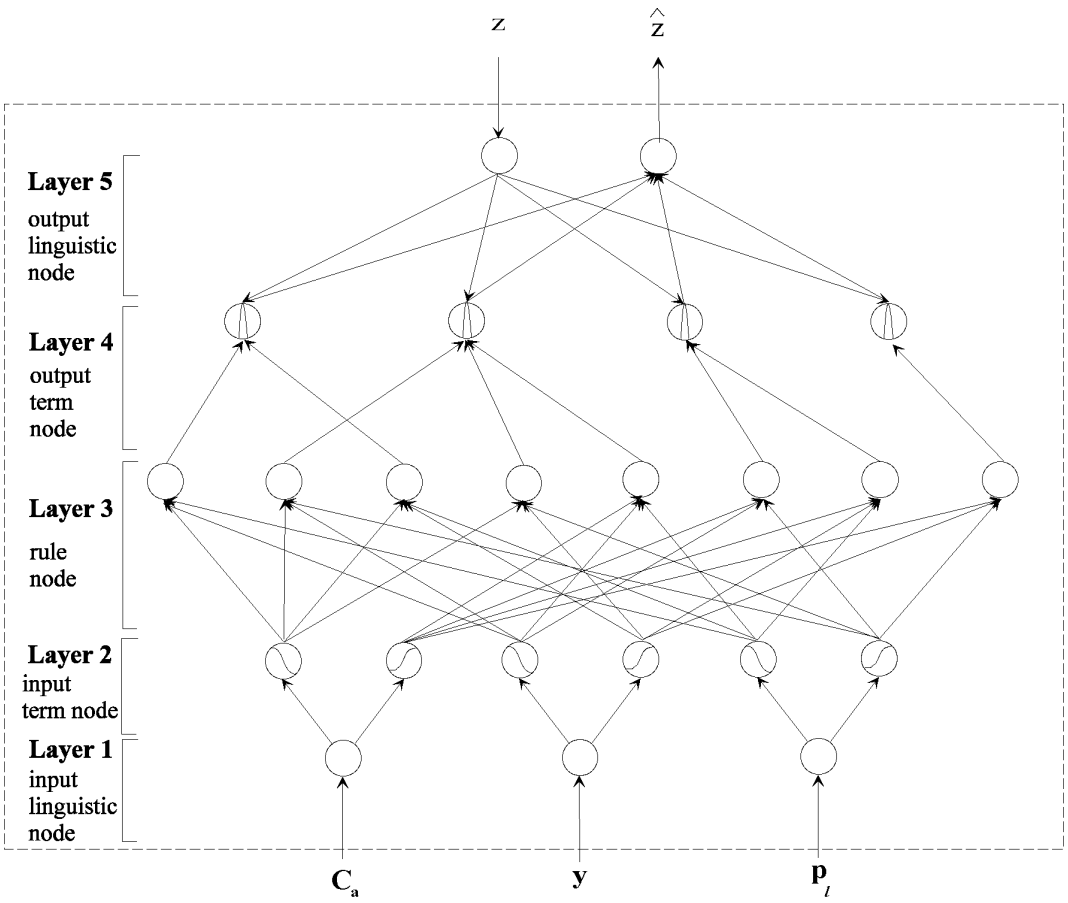


Figure 3.2: The architecture of the NFCAC controller

y and p_l. In order to provide a soft admission decision with multiple decision result levels, not only "Accept" and "Reject" but also "Weak Accept" and "Weak Reject" are employed to describe the accept/reject decision. Therefore, the NFCAC controller may have an alternative choice for calls which fall into the area around the call acceptance/rejection decision boundary. Thus, the term set of the output linguistic variable $\hat{z}$ is defined as $T(\hat{z}) = \{$*Reject (R)*, *Weak Reject (WR)*, *Weak Accept (WA)*, *Accept (A)*$\}$, and consequently there are four corresponding nodes in the fourth layer.

3.2.3 Hybrid Learning Algorithm

A hybrid learning algorithm is applied in the design of the NFCAC controller. The algorithm is a two-phase learning method. In phase one, a self-organized learning scheme is used to construct the rules and to locate the initial membership functions. In phase two, a supervised learning scheme is adopted to optimally adjust the membership functions for desired outputs. Training data must be provided for the learning process, in addition to the size of the term set for each input/output linguistic variable and the fuzzy control rules. The procedure for constructing the set of training data is described below:

[*Construction of Training Data:*]
For a new connection request with traffic parameters of
R_m, R_p, and T_p

Estimate the required capacity C_e by using fuzzy bandwidth estimator

Count the available capacity C_a by using network resource estimator

Generate a congestion indicator y by using fuzzy congestion controller

Get the cell loss ratio p_l measured from system information statistics

If $p_l >$ QoS

Then

Reject the request and set the desired output

$z = 0$

Else

Accept the request and set the desired output

$z = 1$

[*Verification of the Acceptance Decision:*]

Continue the simulation for a predefined time interval, without accepting any new connection requests

Obtain the statistics of cell loss ratio p_l'

If $p_l' >$ QoS (acceptance decision is failed), then

Set $z = 0$

EndIf

EndIf

Store training data of C_a, y, p_l, and z ∎

Using the input training data C_a, y, p_l, the desired output z, the fuzzy partitions $|C_a|$, $|y|$, $|p_l|$, $|\hat{z}|$, and the desired shape of the membership functions, the self-organized training would locate the membership functions and find the fuzzy control rules. If an initial knowledge base is employed to help constructing an initial structure of the fuzzy control rules, a number of possible rule structures can be formed by slight modification of rules. Among all of the possible structures, the one that yields the minimum square error E for the training data is selected. E is defined as

$$E = \frac{1}{2}\sum_{j=1}^{N}[z(t_j) - \hat{z}(t_j)]^2, \tag{3.1}$$

where N is the number of training data, $z(t_j)$ and $\hat{z}(t_j)$ are the desired output and the actual output obtained at time t_j, respectively.

If an initial knowledge base is not provided, the initial locations of membership functions are estimated by using Kohonen's self organizing feature-maps algorithm and the *N-nearest-neighbors* scheme [26], and the initial rule structure is constructed via genetic algorithms

(GAs) [27].

The procedure to locate the means m_i of the i-th membership function for linguistic variable x, $1 \leq i \leq M$, given a set of training data x_j for x, $1 \leq j \leq N$, is described below. It employs the statistical clustering technique of Kohonen's feature-maps algorithm [25].

[*Obtain m_i by using Kohonen's Feature-Maps Algorithm:*]

Step 1: Set initial values of m_i for all membership functions, $1 \leq i \leq M$, such that

$$\min_{1 \leq j \leq N} x_j \leq m_i \leq \max_{1 \leq j \leq N} x_j.$$

Set an initial learning rate α $(0 < \alpha < 1)$.

Step 2: Set $j = 1$.

Step 3: Present training data x_j and compute the distance $d_i = |x_j - m_i|$, $1 \leq i \leq M$.

Step 4: Determine the kth membership function which has the minimum distance d_k $(d_k = \min_{1 \leq i \leq M} d_i)$.

Update m_k by

$$m_k = m_k + \alpha(x_j - m_k).$$

Step 5: If $j < N$, $j = j + 1$, Goto **Step 3**

Else

Decrease α and Goto **Step 2**.

EndIf ∎

The above procedure will stop until $\alpha \leq 0$. The determination of which d_i is minimum at **Step 4** can be quickly accomplished in constant time via a winner-take-all circuit [25]. The adaptive algorithm can be independently performed to obtain m_i for each input and output linguistic variables.

As for the corresponding standard deviation σ_i of the i-th membership function of x, since m_i and σ_i will be finely tuned in the supervised learning phase, we just use a first-nearest-neighbor heuristic to estimate σ_i, which is given by

$$\sigma_i = \frac{|m_i - m^*|}{\gamma}, \tag{3.2}$$

where

$$m^* = \begin{cases} m_{i-1} & \text{for } |m_i - m_{i-1}| < |m_i - m_{i+1}| \\ m_{i+1} & \text{otherwise,} \end{cases} \tag{3.3}$$

and γ is called an overlap parameter used to describe the degree of overlapping for the two membership functions.

GAs are search algorithms based on the mechanics of natural selection and natural genetics [29, pp. 1–22]. They combine the survival of the fittest and some of the innovative flair of human search. According to the fittest values among those randomly selected string structures, a structured but randomized information exchange is defined to form a search algorithm. Although the randomized generating procedure is used, GAs are not simple random walks. They efficiently make use of the historical information to speculate on new search points with expected improved performance [29]. The input/output rule structure is encoded into a gene string $G(t)$ defined as

$$G(t) = [g_1(t), g_2(t), \ldots, g_n(t)], \tag{3.4}$$

where n is the total number of rule nodes, and $g_i(t)$ $(1 \le i \le n)$ denotes the i-th gene in $G(t)$. For example, if the i-th rule node in Layer 3 is connected to the j-th node $(1 \le j \le |T(\hat{z})|)$ in Layer 4 at time t, then $g_i(t)$ is set to j. Initially, the rules $g_i(0)$ are integers and are randomly assigned within the range of $[1, |T(\hat{z})|]$. $G(t)$ is then updated by genetic operators of *crossover* and *mutation* according to the value of fitness function, which is defined as the inverse of the error E defined in Eq. (3.1). The structure that provides the minimum value of E will be chosen as the optimal structure.

After the self-organized training phase, the NFCAC controller then enters the supervised learning phase. The aim of the supervised learning is to further minimize E for the training data using a back-propagation learning algorithm. Starting at the output node, a backward pass is used to compute $\frac{\partial E}{\partial w}$ for all the hidden nodes in Layer 4 and Layer 2. Assuming that w is an adjustable parameter in a node (i.e., the mean or the standard deviation of the membership function), the general learning rule is

$$w^{new} = w^{old} + \eta \frac{\partial E}{\partial w}, \tag{3.5}$$

where η is the learning rate and

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial f}\frac{\partial f}{\partial w} = \frac{\partial E}{\partial a}\frac{\partial a}{\partial f}\frac{\partial f}{\partial w}. \tag{3.6}$$

f and a were defined in the previous subsection. Here, different values of η could be used in Layer 2 and Layer 4 to provide different learning rates for input and output variables. Different values of η represent different adoption rates for these variables. If the membership function of a specific linguistic variable is not intended to be modified, then $\eta = 0$ is used.

3.3 Simulation Results and Discussions

Simulations were performed to test the effectiveness of the proposed NFCAC scheme. Before discussing the results of the simulations, we will first describe the simulation environment.

3.3.1 Simulation Environment

Assume that an ATM network is chosen to be the high-speed network supporting multimedia services. The input traffic is categorized into two types: real-time (type-1) and non-real-time (type-2) traffic. Video and voice services are examples of type-1 traffic, while data services are examples of type-2 traffic. The network system provides two separate finite

buffers with size K_i, in order to support different QoS requirements for type-i traffic, i=1 and 2. When the buffer is full, incoming cells are blocked and lost. The system reserves C_r portion of its capacity for type-1 traffic and the remaining $(1-C_r)$ portion for type-2 traffic. When there is unused type-1 or type-2 capacity, it is used for the other type of traffic. In the simulations described here, $K_1 = K_2 = 100$ cells and $C_r = 0.8$. Also, the QoS requirement for type-1 traffic $QoS_1 = 10^{-5}$ and that for type-2 traffic $QoS_2 = 10^{-6}$.

The cell generation process for a video coder is assumed to have two motion states: one is the low motion state for the rate of interframe coding and the other is the high motion state for the rate of intraframe coding [30]. The rate of intraframe coding is further divided into two parts: the first part has the same rate as the interframe coding and the second part, called difference coding, is the difference between the rates of intraframe coding and interframe coding. The interframe coding and the difference coding are all modeled as discrete-state Markov-modulated Bernoulli processes (MMBP) with basic rates A_r and A_a. The state-transition diagram is shown in Figs. 3.3(a) and 3.3(b). Let $\lambda_a(t)$, $\lambda_r(t)$, and $\lambda'_a(t)$ denote the cell generation rates for intraframe coding, interframe coding, and difference coding at time t, respectively, from the video coder. Clearly, $\lambda_a(t) = \lambda_r(t) + \lambda'_a(t)$. The process of $\lambda_r(t)$ is an (M_r+1)-state birth-death Markov process. The state-transition diagram for $\lambda_r(t)$ uses the label $m_r A_r$ to indicate the cell generation rate of interframe coding of a state and uses the labels $(M_r - m_r)\gamma$ and $m_r\omega$ to denote the transition probabilities from state $m_r A_r$ to state $(m_r+1)A_r$ and from state $m_r A_r$ to state $(m_r-1)A_r$, respectively. Similarly, the process for $\lambda'_a(t)$ is an (M_a+1)-state birth-death Markov process. The state-transition diagram for $\lambda'_a(t)$ uses the label $m_a A_a$ to indicate the additional cell generation rate of a state due to intraframe coding and uses the labels $(M_a - m_a)\phi$ and $m_a\psi$ to denote the transition probability from state $m_a A_a$ to state $(m_a+1)A_a$ and from state $m_a A_a$ to state $(m_a-1)A_a$, respectively. One should note that the long-term correlation behavior of a video source is resulted from the

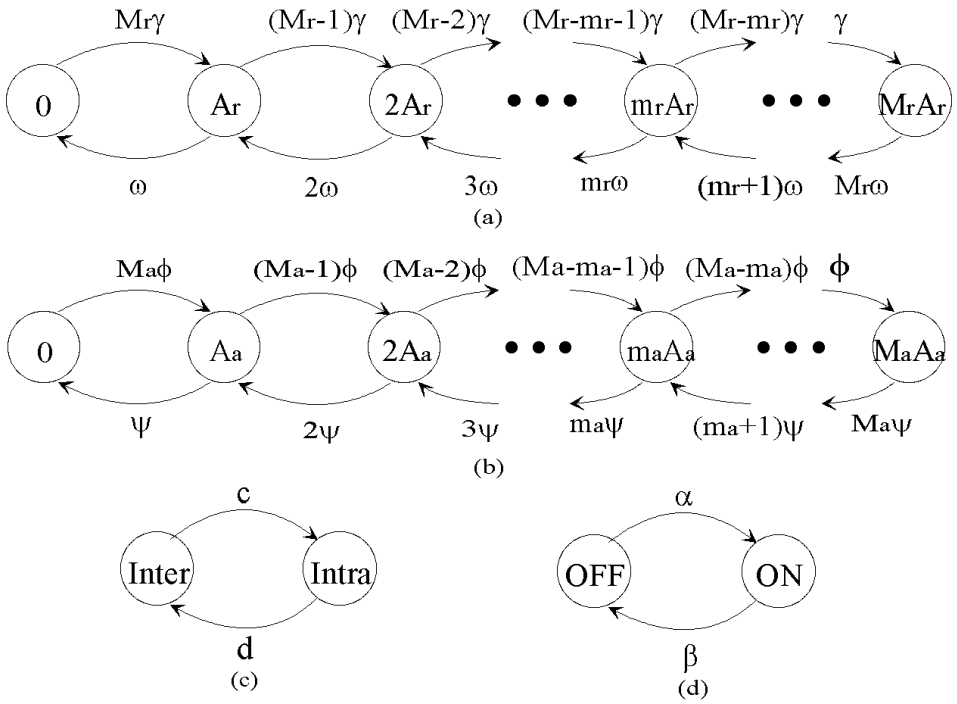


Figure 3.3: Level transition diagram for (a) interframe coding $\lambda_r(t)$ (b) difference state $\lambda'_a(t)$ (c)interframe and intraframe alternate model (d)voice source

process $\lambda_a(t)$. The video source will alternate between interframe and intraframe, depending on the video source activity factor. As shown in Fig. 3.3(c), there is a transition rate c in the interframe state and a transition rate d in the intraframe state. The values of γ, ω, M_r, A_r, ϕ, ψ, M_a, A_a, c, and d can be obtained from the traffic variables R_p, R_m, and T_p.

The cell generation process for a voice call is modeled by an interrupted Bernoulli process (IBP) [28]. As shown in Fig. 3.3(d), during the ON (talkspurt) state, voice cells are generated with rate A_v; during the OFF (silence) state, no cells are generated. A voice source has a transition rate α in the OFF state and a transition rate β in the ON state.

As for the data source, there are high-bit-rate and low-bit-rate data services. The generation of high-bit-rate and low-bit-rate data cells is characterized by Bernoulli processes with rates θ_1 and θ_2, respectively. Also, the distributions of the holding times for video, voice,

high-bit-rate data, and low-bit-rate data are assumed to be exponentially distributed.

In the simulations, for the arrival process of a video source, it is assumed that R_p=3.31 $\times$ 10^{-2}, R_m=1.10 $\times$ 10^{-2}, and T_p=0.5 seconds, which would give M_r=M_a=20, A_r=1.34 $\times$ 10^{-3}, A_a=3.15 $\times$ 10^{-4}, γ= 3.77 $\times$ 10^{-6}, ω=5.65 $\times$ 10^{-6}, ϕ=ψ=2.83 $\times$ 10^{-5}, c=5.65 $\times$ 10^{-6}, and d=5.09 $\times$ 10^{-5}; for the arrival process of a voice source, it is assumed that R_p=4.71 $\times$ 10^{-4}, R_m=2.12 $\times$ 10^{-4}, and T_p=1.35 seconds, which would give A_v=4.71 $\times$ 10^{-4}, α=1.71 $\times$ 10^{-6}, and β=2.09 $\times$ 10^{-6}; for high-bit-rate data sources, it is assumed that R_p=7.36 $\times$ 10^{-2}, R_m=7.36 $\times$ 10^{-3}, and T_p=3.14 $\times$ 10^{-2} seconds, which would give θ_1=0.1, and for low-bit-rate data sources, it is assumed that R_p=3.68 $\times$ 10^{-2}, R_m=7.36 $\times$ 10^{-4}, and T_p= 2.88 $\times$ 10^{-2} second, which give θ_2=0.02. The mean holding time is 60 minutes for a video service, 3 minutes for a voice service, and 18 seconds for both high- and low-bit-rate data services. Notice that the values of R_p and R_m have been normalized by the network capacity.

Two kinds of cell loss ratios for type-i traffic are considered: the source loss ratio due to selective discarding at the customer side $p_{s,i}$ and the node loss ratio due to blocking at the network side $p_{n,i}$. The overall cell loss ratio for type-i traffic $p_{l,i}$ is defined as

$$p_{l,i} = \kappa p_{s,i} + p_{n,i}, \quad i = 1, 2, \tag{3.7}$$

where κ is used to indicate the significance of the node loss ratio over the source loss ratio. $\kappa = 0.8$ is assumed here because selectively discarding cells at the source should have less effect on information retrieval than blocking cells at the node. In the simulations, the cell loss ratio is estimated as the total loss cells divided by the arriving cells during the whole simulation interval.

3.3.2 Simulation Results and Discussions

On the basis of prior knowledge concerning CAC, the rule structure and parameters of the NFCAC controller can be initially set and then properly adjusted via the learning

algorithm. The membership functions of the linguistic variables for type-1 and type-2 traffic were initially specified in the left-hand side of Fig. 3.4(a) and Fig. 3.4(b), respectively. As we know, the available capacity C_a, deduced from the equivalent capacity C_e of the existing calls, may possess estimation errors. In order to utilize the network as much as possible, we may employ an idea of "budget deficit" to over-assign the capacity. Thus, the mean value $m_{11}^{(I)}$ of the membership function of NE was set to be a negative value and the mean value $m_{12}^{(I)}$ of the membership function of E was set to be a value close to zero.

The behavior of the congestion indicator y could be monitored from the congestion and congestion-free states during a long-term simulation of the network operation. Thus, the membership functions of y could be initially optimized based on the obtained information. The mean value $m_{22}^{(I)}$ of the membership function of P would be set to be the mean value of the queue-length change rate during congestion-free periods, the mean value $m_{21}^{(I)}$ of the membership function of N would be set to be the mean value of the queue-length change rate during congestion periods, and let $\sigma_{21}^{(I)} = \sigma_{22}^{(I)} = m_{22}^{(I)} - m_{21}^{(I)}$. These parameters could be further off-line optimized via GA by simulation.

The initial membership functions of the cell loss ratio p_l were set according to the QoS requirement. The mean value $m_{32}^{(I)}$ of the membership function of NS would be set to be the QoS requirement, the mean value $m_{31}^{(I)}$ of the membership function of S would be set to be a fraction of the QoS requirement, and the standard deviations would be set to be $\sigma_{31}^{(I)}$=$\sigma_{32}^{(I)}$= $m_{32}^{(I)} - m_{31}^{(I)}$. As a result, there exists a safety margin between the membership functions of terms S and NS provided to tolerate the dynamic behavior of the network operation and insure the QoS requirement.

Here, little information about the setting of initial values for the mean $m_j^{(O)}$ of the term set T($\hat{z}$) could be employed; therefore, the values of $m_j^{(O)}$ are set to be equally spaced in the range of [0,1]. Based on the initial membership functions, an optimal rule structure shown

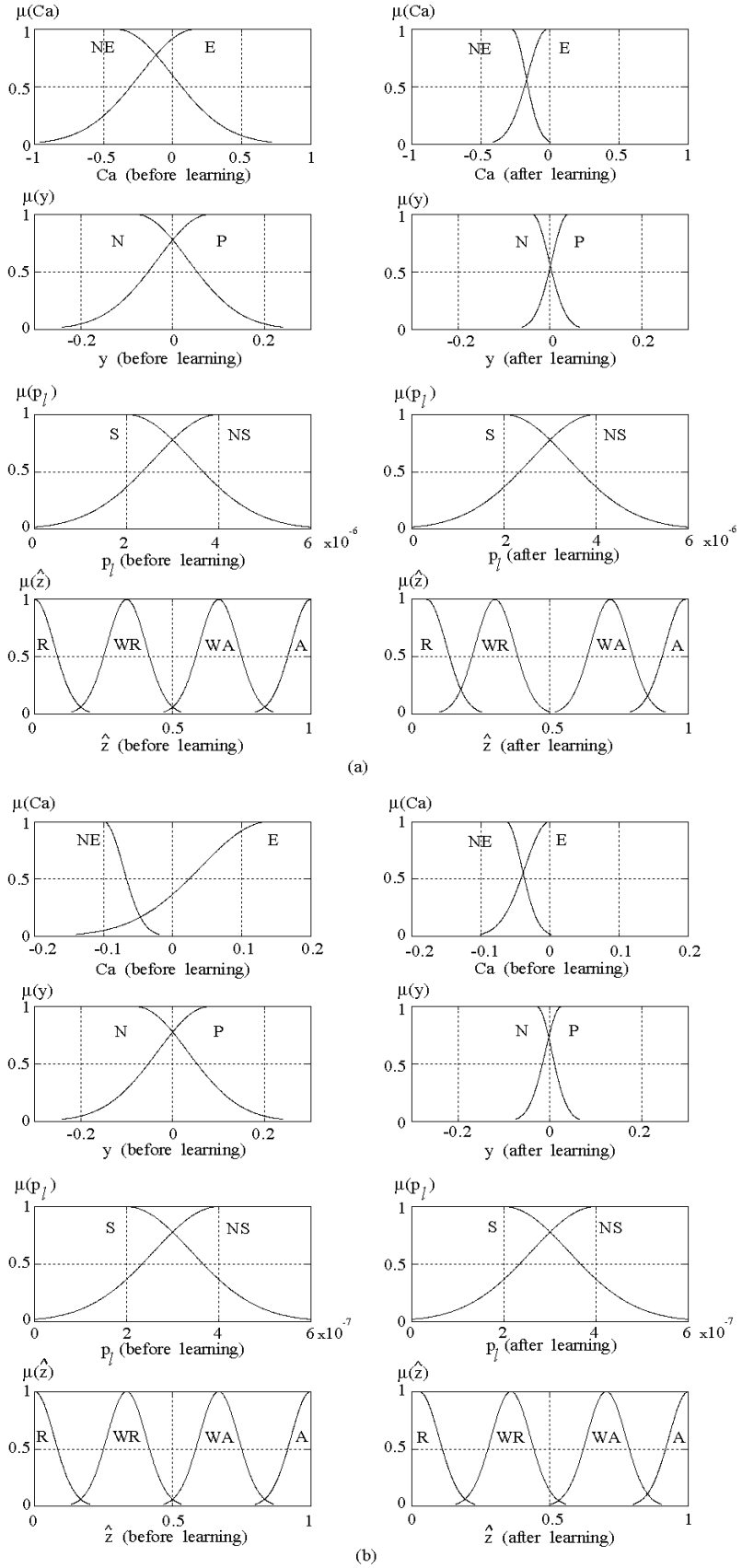


Figure 3.4: Membership functions of C_a, y, p_l and $\hat{z}$ for (a) type-1 traffic, (b) type-2 traffic

Rule	C_a	y	p_l	$\hat{z}$	Rule	C_a	y	p_l	$\hat{z}$
1	NE	N	NS	R	5	E	N	NS	WR
2	NE	N	S	WR	6	E	N	S	WA
3	NE	P	NS	R	7	E	P	NS	WA
4	NE	P	S	WR	8	E	P	S	A

Table 3.1: The rule structure for the NFCAC

in Table 3.1 was obtained by using GA in the self-organized learning phase. When the fuzzy logic rules were found, the NFCAC controller entered the supervised learning phase, in which the membership functions were adjusted optimally.

Three different values of η were used for the variables C_a, y, p_l, and $\hat{z}$. η was set to zero for p_l because the membership functions were specified by the QoS constraint and should not be modified. $\eta = 0.001$ was used for y because the membership functions of y were initially optimized. As for C_a and $\hat{z}$, their initial membership functions were heuristically set and required further optimization in the supervised learning phase. Thus, $\eta = 0.01$ was used. The use of different η may drastically reduce the training time required in the supervised learning phase. The learned membership functions of the linguistic variables for type-1 and type-2 traffic were shown in the right-hand side of Fig. 3.4(a) and Fig. 3.4(b), respectively.

For type-1 traffic in Fig. 3.4(a), it can be found that the differences of the membership functions before and after learning are: For the membership functions of C_a, the mean value $m_{11}^{(I)}$ of the membership function of NE was properly modified from -0.4 to -0.27. Similarly, the mean value $m_{12}^{(I)}$ of the membership function of E was properly modified from 0.16 to -0.02. There is a drastically change for membership functions of C_a, and the phenomenon can also be found in the membership functions of y. It is because we heuristically set their initial values and we used only two terms to describe C_a or y. The change of the position of one term of C_a and y will squeeze the other term but receive less counteraction from the other one term (compared to $\hat{z}$ described later). Membership functions of p_l are not changed since

η for p_l was chosen to be zero. For the membership function of $\hat{z}$, however, the mean $m_1^{(O)}$ of the membership function of R is slightly increased from 0 to 0.05, representing that the effect of "Reject" is decreased. Also, the mean $m_3^{(O)}$ of the membership function of WA is slightly increased from 0.67 to 0.72, representing that the effect of "Weak Accept" is increased. The small change is because we used four terms to describe $\hat{z}$. The change of the position of one term of $\hat{z}$ will squeeze the other three terms but receive more counteraction from the three terms. Therefore, the change of position would be confined in a smaller range. The changes of membership functions of $\hat{z}$ imply that the NFCAC controller prefers to accept new calls. This phenomenon demonstrates that the NFCAC controller intends to recover some system bandwidth which the equivalent capacity method wastes due to over-estimation, while keeping the QoS contract. It may be the reason for the utilization improvement of the proposed NFCAC controller, which will be shown below. Similar results could be found for type-2 traffic in Fig. 3.4(b).

We compare the NFCAC scheme with the effective-band-width-based CAC (EBCAC) scheme proposed in [10], the fuzzy-logic-based CAC (FLCAC) scheme proposed in [18], the neural-net-based CAC (NNCAC) scheme proposed in [23], and the radial-basis-function-based CAC (RBFCAC) scheme from the aspects of the cell loss ratio (CLR), the system utilization, and/or the training time under the constraint of QoS guarantee. The EBCAC scheme is a hybrid technique combining the conventional techniques of the Gaussian approximation and the bufferless analysis; it is an improved version of the equivalent capacity method [8]. Simulation of the EBCAC scheme is simply to calculate the required bandwidth of a new connection. The new connection request is accepted if the total bandwidth required by the new connection and the existing connection is less than the system capacity. Otherwise, it is rejected. The FLCAC scheme is a fuzzy implementation of the equivalent capacity admission control method; details for the FLCAC scheme can be referred to

[18]. The NNCAC and RBFCAC schemes are neural-net implementation of the equivalent capacity admission control method, where the NNCAC adopts the multi-layer perceptron (MLP) structure with 30 hidden nodes, while the RBFCAC uses radial basis function network (RBFN) with 30 hidden nodes. Details for the NNCAC scheme can be referred to [23]. In the simulations, the FLCAC, NNCAC, or RBFCAC controller is equipped with the same three peripheral processors as those used in the NFCAC controller shown in Fig. 3.1. The sizes of training set and test set are all equal to 200, the number of repeated experiments is 20, and the standard deviation is less than 5%.

Fig. 3.5 shows the CLRs of an ATM traffic controller employing the NFCAC scheme, and the EBCAC, FLCAC, NNCAC, RBFCAC schemes. It is found that the QoSs for both types of traffic are indeed guaranteed for all of these control schemes. Fig. 3.6 shows that the system utilization of the NFCAC scheme and the four schemes. We can find that the utilization of the NFCAC scheme is slightly greater than that of the NNCAC and the RBFCAC schemes; the system utilizations of NFCAC, NNCAC, and RBFCAC are 91%, 90.5%, and 89%, respectively; and the NFCAC scheme offers about 32% and 11% greater system utilization than the EBCAC scheme and the FLCAC scheme. It is because NFCAC can incorporate the domain knowledge obtained from both the analytical-based method (the equivalent capacity scheme [8] is employed in the bandwidth estimator) and the measurement-based method (the system statistics of the queue length, the change rate of the queue length, and the CLR are considered in the congestion controller). Also, the reason for the performance improvement is that NFCAC possesses the learning capability of the neural network.

Fig. 3.7 shows the training time required for the NFCAC scheme and the NNCAC, RBFCAC schemes. Here, a widely used back-propagation learning algorithm was employed to adjust the membership functions (i.e. represented in terms of weights) of the multi-layer

neural fuzzy network and neural network for the NFCAC and NNCAC schemes, while the RBFCAC scheme is basically trained by the hybrid learning rule: unsupervised learning in the input layer and supervised learning in the output layer. It is found that NFCAC has training time of 7 (4) epochs, while RBFCAC and NNCAC have training time of 103 (40) and 5×10^4 (6×10^2), respectively, for type-1 (type-2) traffic. The NFCAC has higher learning speed than the RBFCAC and NNCAC. One reason is that the neural fuzzy network is a structured network, thus the NFCAC controller can easily adopt the domain knowledge of conventional control methods to construct the initial rule structure and the parameters of the membership functions, providing an excellent initial guess in adjusting its weights; on the contrast, the neural network is a non-structured network, which cannot incorporate domain knowledge about system. The other reason is that the neural fuzzy network has simpler structure than the neural network; the number of tunning parameters used in the neural fuzzy network is quite small, as compared to the neural network such as MLP and RBFN considered here. In this chapter, there are only 16 weighting parameters used in NFCAC, while there are 150 and 480 weighting parameters required for the RBFCAC and NNCAC, respectively. It is also noted that the RBFCAC scheme has less learning time than the NNCAC scheme. This is because the RBFCAC scheme can have the proper initial setting of means and variances for the Gaussian activation functions during unsupervised learning according to the prior knowledge, and it has only one layer of connection needed to be trained by supervised learning.

As usually noted, RBFCAC can have faster training speed than NNCAC but cannot achieve the same accuracy as the back-propagation NNCAC. In the simulations, we first adopted the same set of data used to train NFCAC and NNCAC for RBFCAC. However, it was found that RBFCAC finally violated the QoS contracts due to its error decision of accepting more users than it should be. In order to provide QoS guarantee for RBFCAC, we

have to prepare much more training data, especially those around the acceptance/rejection boundary. This will increase the training time of RBFCAC in each epoch than those required by NFCAC and NNCAC. Moreover, the overall processing time of RBFCAC is greater than that needed by either NFCAC or NNCAC because RBFCAC uses more nodes (compared with NFCAC) and a more complicated activation function (compared with NNCAC). All these would degrade the performance of RBFCAC in real application.

3.4 Concluding Remarks

This chapter has proposed a neural fuzzy approach for connection admission control in high-speed multimedia networks. The neural fuzzy connection admission control (NFCAC) scheme combines the linguistic control capability of a fuzzy logic controller and the learning ability of a neural network. This type of integrated neural fuzzy system can automatically construct a rule structure by learning from training examples and can self-calibrate parameters of membership functions. It not only provides a robust framework to mimic experts' knowledge embodied in existing traffic control techniques but also constructs intelligent computational algorithms for traffic control. It can be easily trained and enhances system utilization. Simulation results show that the proposed NFCAC scheme provides system utilization about 32% and 11% higher than the EBCAC and FLCAC schemes proposed in [10] and [18], respectively, and the NFCAC scheme requires only a fraction of the 10^3 order and the 10^1 order of training cycles, consumed by the NNCAC scheme proposed in [23] and RBFCAC scheme, respectively. An NFCAC scheme such as the one introduced here may be the answer to the problem of designing a coherent call admission controller for ATM systems.

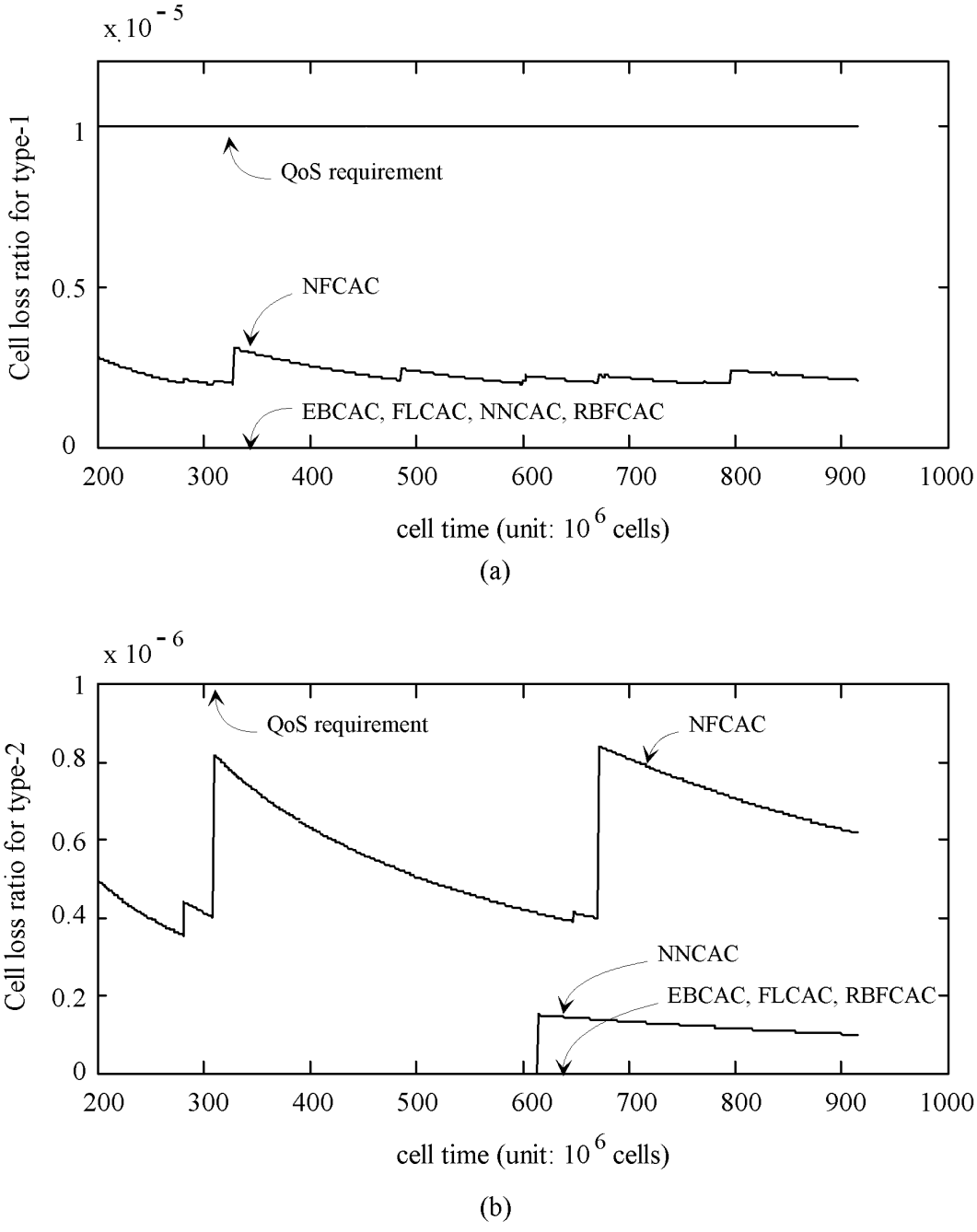


Figure 3.5: Cell loss ratio for (a) type-1 traffic, (b) type-2 traffic

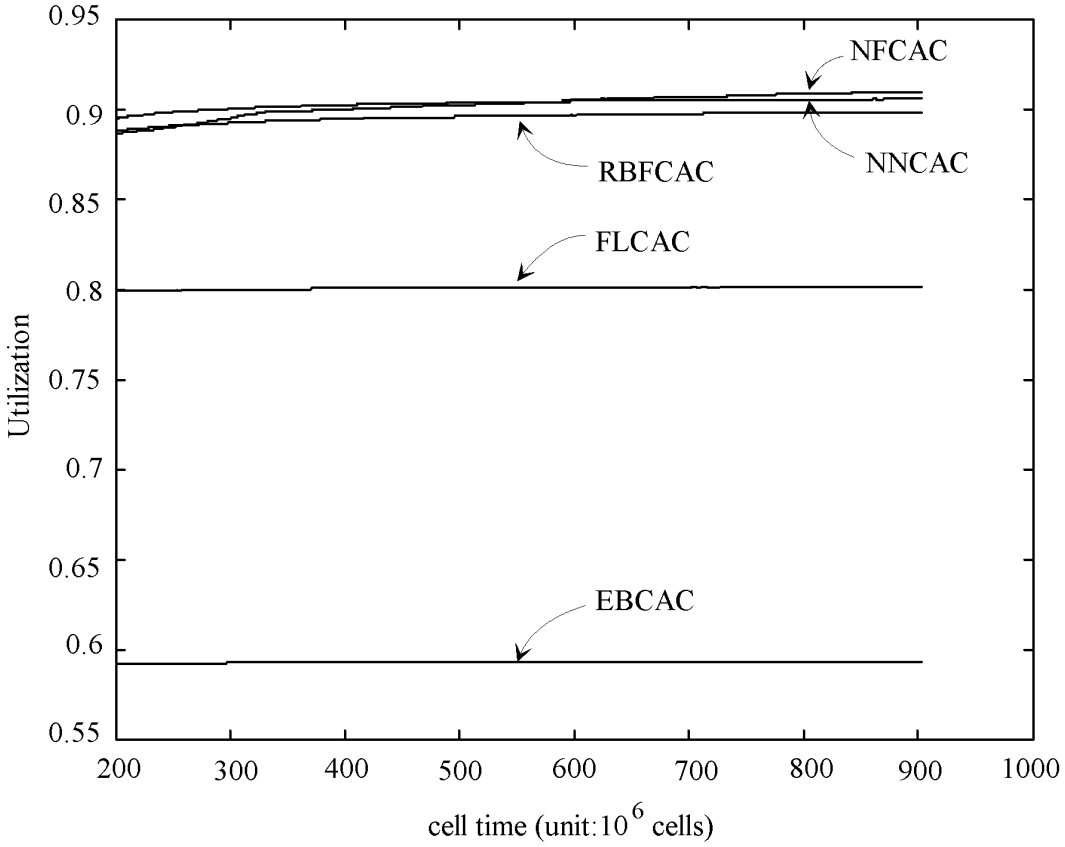
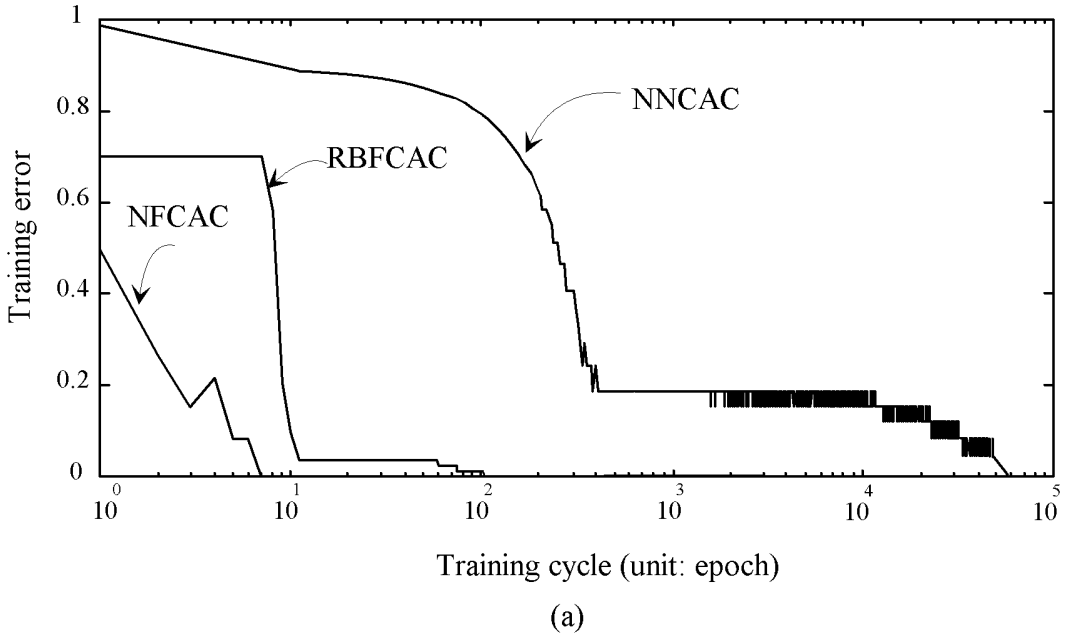
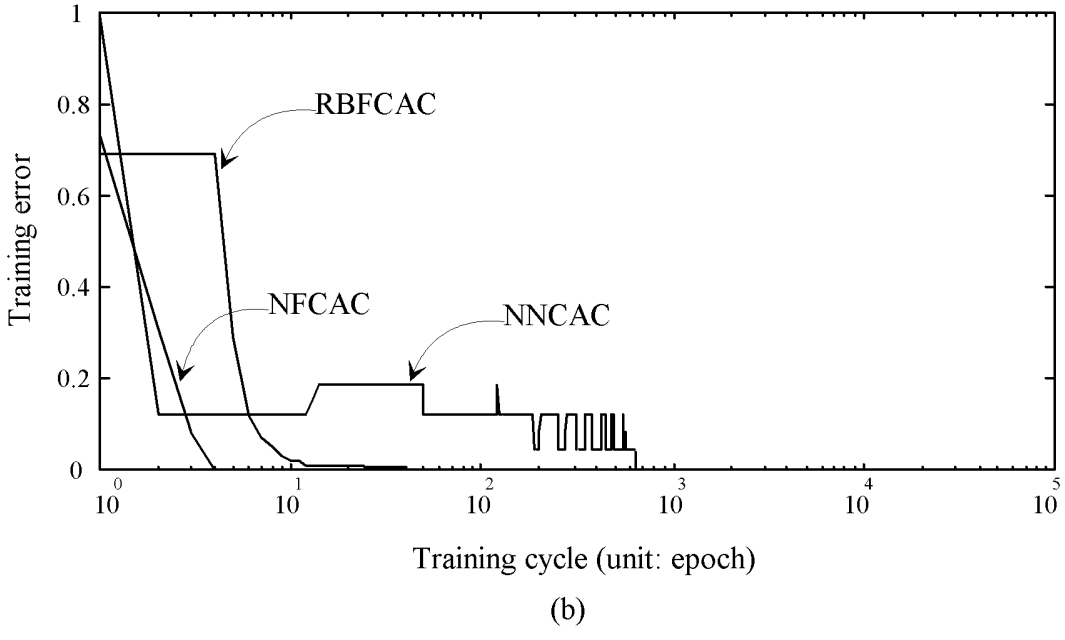


Figure 3.6: System utilization



(a)



(b)

Figure 3.7: Training cycles needed for (a) type-1 traffic, (b) type-2 traffic

Chapter 4

Intelligent Connection Admission Control Scheme for Multimedia High-speed Networks Using Frequency-domain Traffic Parameters

As contrast to the CAC scheme proposed in Chapter 3, which is based on the time-domain traffic parameters to make CAC decisions, this chapter proposes a power-spectrum-based neural-net connection admission control (PN-CAC) scheme for multimedia high-speed ATM networks. It employs a neural network controller to handle the CAC function according to the frequency-domain power spectral density (PSD) parameters of the traffic sources. Since the PSD function of an input traffic contains the correlation and burstiness properties of the traffic, and it has been proven capable to characterize the queueing performances of the input traffic, the PSD parameters describing the PSD function can well correspond to the queueing performances also. With a composition algorithm to easily obtain the three PSD parameters of an aggregate traffic, it is suitable to adopting PSD parameters for CAC accordingly. Simulation results show that, after well training the neural network, an optimal CAC decision hyperplane based on the input variables is constructed to provide an efficient and robust admission control under dynamic network environments, while the QoS requirements are strictly assured.

4.1 Introduction

Multimedia high-speed networks should be equipped with a set of traffic control functions to ensure QoS of each service connection and to enhance system utilization. One of the traffic control functions is connection admission control. Connection admission control (CAC) is defined as "a set of actions taken by the network in order to determine whether a connection can be accepted" [31]. A new connection is accepted only if sufficient network resources are available and the required performance can be maintained.

Several conventional CAC control techniques for high-speed networks have been proposed. In the peak rate allocation, QoS is always guaranteed if the aggregate bit rate never exceeds the system capacity. However, it leads to low utilization of network resources. An equivalent capacity (effective bandwidth) method was proposed to estimate the required bandwidth for individual or aggregate connections with desired QoS [8], [11]. A call admission scheme by inferring the upper bound of cell loss probability from the traffic parameters specified by users was studied in [13]. And a simple bandwidth assignment policy by classifying all traffic sources was presented. All the studies were conducted mainly on the basis of traffic parameters in *time domain*.

On the other hand, Li and Hwang [33] and Sheng and Li [34] have studied the queueing performance of a high-speed network from the point of view in the *frequency-domain traffic parameters*. The process of input traffic inherently contains a *power spectral density* (PSD) function, which is the Fourier transform of the input traffic process's autocorrelation function. From their studies, two characteristics of PSD are concluded: (i) The PSD can be represented by three main parameters such as the DC component, the average power, and the half-power bandwidth. (ii) The low-frequency band of the input PSD has a dominant impact on queueing performance, while the high-frequency band can be neglected to a large extent. It is because the low frequency component of PSD contains the correlation and burstiness of

the input process. The more the low-frequency components are, the burstier the input traffic will be [35]. Therefore, according to the above two PSD characteristics from Li's studies, it can be conducted that these three PSD parameters can well characterize the input traffic and correspond to its queueing performances, and thus this reveals a chance to employ the PSD parameters for CAC.

A composition algorithm is proposed in [36] to obtain the three PSD parameters of an aggregate traffic source from the given PSD parameters of these individual traffic sources which build the aggregate one. The computation process of the composition algorithm is just through some simple arithmetic operations. It can then be concluded that PSD parameters possess *additive* property; this makes the PSD parameters more suitable for admission control, no matter how many types of traffic sources there are, because the PSD parameters of the virtually aggregated total traffic enrolling the new call could be easily derived as the new call request arrives and maintain the same (three) reference variables, which can correspond to the queueing performances, for the admission control decision making. The design of the CAC algorithm based on PSD parameters can be made accordingly and this indeed greatly reduce the complexity for admission control.

A power-spectrum-based table-lookup CAC method for multimedia communications in ATM networks was studied, where the table content was the cell loss probability indexed by the PSD parameters of voice/video calls and arrival rates of data calls [36]. The table can be constructed through several explorative simulations. However, since the table is constructed based on the original three parameters for the power-spectrum: DC component, half-power bandwidth, and average power [36], there is a drawback of large-dimensional CAC table. An "equivalent source" concept is consequently introduced to transform the PSD parameters of an offered traffic source into the so called "equivalent" PSD parameters which are corresponding to another (equivalent) traffic source [37]. The word "equivalent" exactly stands for

almost the same queueing performances in some evaluation aspects. That is, the corresponding traffic source with the "equivalent" PSD parameters generated by the transformation is expected to have equivalent queueing performances with the offered traffic source characterized by original PSD parameters, so that the equivalent PSD parameters could substitute the original ones. The transformation was done by using a low-pass filter and an integrator, while a heuristic method to obtain a typical value for the cut-off frequency is adopted. A modified power-spectrum-based table-lookup CAC method was then proposed in [37] where the CAC lookup table is significantly reduced by one dimension than that proposed in [36], since the PSD parameters of each table entry in [36] can be transformed to the equivalent ones with the pre-defined half-power bandwidth value which is identical among all transformed entries, and thus only the DC component and the transformed equivalent average power have to be specified to characterize and distinguish each (voice/video) traffic source. The offered three PSD parameters of a new call request would also be transformed to the equivalent ones at first to adapt to the operations based on the dimension-reduced CAC table. Although the transformation may introduce some degradations on performances, simulation results show that the modified power-spectrum-based table-lookup CAC scheme is still efficient enough by about 9% higher system utilization than that of the conventional equivalent capacity CAC method [8], while improves the feasibility for practical implementations.

In recent years, neural networks have been widely employed to deal with the traffic control problems in high-speed multimedia networks [20], [23]. A major feature of the neural network is the self-learning capability which can be utilized to characterize the relationship between input traffic and system performance. In [20], Hiramatsu proposed a connection admission controller using neural network. The neural network in the controller learns the relations between the offered traffic and the service quality. Because the declared traffic parameters were used only to divide calls into several bit-rate classes, the neural network actually learns

the relationship between the numbers of existing connections in each bit-rate class and their corresponding QoSs according to the statistical characteristics of each bit-rate class. Results showed that the neural network learned a complicated boundary for call acceptance decision. We have also proposed a neural network connection admission control (NNCAC) scheme [23] and a neural fuzzy connection admission control (NFCAC) scheme as mentioned in Chapter 3 for ATM networks. Simulation results reveal that call admission control with either neural networks or neural fuzzy networks can improve significantly system utilization, under QoS constraint.

In this chapter, we propose a power-spectrum-based neural-net connection admission control (PNCAC) method for multimedia high-speed networks. It introduces the three PSD parameters (DC component, half-power bandwidth, and average power) of the virtually aggregated total traffic as inputs and chooses a neural network controller to accommodate all the inputs and generate the admission control decisions. We first transform the time-domain parameters of source traffic of connections into the power-spectrum parameters in frequency domain, then a decision hyperplane of the connection admission control is constructed under the constraint of QoS after the neural network has been trained. The decision hyperplane splits the sample space into two — one is for "accept" and the other is for "reject." We further adopt the learning/adapting capabilities of the neural network to adjust the optimal location of the boundary between these two decision spaces (i.e. use the back-propagation training algorithm to adjust the link weights of PNCAC to the optimum value). Simulation results show that PNCAC achieves higher system utilization, superior by 23.8% to the conventional equivalent capacity CAC proposed in [8], and comparable to that of Hiramatsu's neural network CAC (NNCAC) [20]. However, PNCAC is more robust than Hiramatsu's NNCAC in dynamic high-speed multimedia networks whose characteristics may alter with time. As characteristics of traffic sources change, the connection number of each traffic type utilized

as the input variables in Hiramatsu's NNCAC can no longer characterize the traffic. At this time, Hiramatsu's NNCAC should perform on-line training and even the node-growing or -pruning learning process in order to adapt to the variation in traffic sources; otherwise, the performance would deeply degrade with the QoS no longer guaranteed. However, the proposed PNCAC can still perform well without any other modifications or re-training process.

The rest of the chapter is organized as follows. Section 4.2 describes the transformation of time-domain traffic parameters of an input process into power-spectrum parameters of its power spectral density. In Section 4.3, we describe the system configuration and address the design of the proposed PNCAC. In Section 4.4, some simulation examples are illustrated to justify the feasibility of PNCAC, by comparing it with the conventional equivalent capacity CAC (ECCAC) and the Hiramatu's NNCAC. Finally, concluding remarks are presented in Section 4.5.

4.2 Power Spectrum of Input Process

If an input rate process $a(t)$ is modeled as an $(M+1)$-state Markov-modulated Poisson process (MMPP), the MMPP can be represented by (**Q**,**r**), where **Q** is the state transition-rate matrix and $\mathbf{r} = [\gamma_0, \gamma_1, ..., \gamma_M]$ is the vector representing the arrival rate at each MMPP state. The stationary probability vector of state, denoted by $\mathbf{\Pi} = [\pi_0, \pi_1, ..., \pi_M]$, can be obtained by solving equations of $\mathbf{\Pi Q} = 0$ and $\mathbf{\Pi e} = 1$, where **e** is an unit column vector. The average input rate $\bar{\gamma}$ is then given by

$$\bar{\gamma} = \sum_{i=0}^{M} \gamma_i \pi_i. \tag{4.1}$$

Q is diagonalizable and can be represented by spectral decomposition as

$$\mathbf{Q} = \sum_{l=0}^{M} \lambda_l \mathbf{g}_l \mathbf{h}_l, \tag{4.2}$$

where λ_l is the l-th eigenvalue of $\mathbf{Q}$, and $\mathbf{g}_l$ and $\mathbf{h}_l$ are the associated right column and left row column eigenvectors of $\mathbf{Q}$ with respect to λ_l, respectively [34].

Then the autocorrelation function of the MMPP, defined as $R(\tau) \equiv \overline{a(t)a(t+\tau)}$, can be derived. Its corresponding PSD, denoted by $P(\omega)$, can also be given, via Fourier transformation of $R(\tau)$, by [34]

$$P(\omega) = \bar{\gamma} + 2\pi\Psi_0\delta(\omega) + \sum_{l=1}^{M} b_l(\omega), \tag{4.3}$$

where

$$\delta(\omega) = \begin{cases} \infty & \text{for } \omega = 0 \\ 0 & \text{elsewhere;} \end{cases} \tag{4.4}$$

Ψ_0 is the DC component, given by

$$\Psi_0 = \bar{\gamma}^2; \tag{4.5}$$

and $b_l(\omega)$ is the bell-shaped function with respect to non-zero λ_l, given by

$$b_l(\omega) = \frac{\Psi_l B_l}{(B_l/2)^2 + (\omega - \omega_l)^2}. \tag{4.6}$$

Ψ_l in (4.6) is the average power contributed by λ_l, given by

$$\Psi_l = \sum_i \sum_j \pi_i \gamma_i \gamma_j g_{li} h_{lj} \quad \text{for } 1 \le l \le M, \tag{4.7}$$

where g_{li} and h_{lj} are the i-th and j-th entities of the vector $\mathbf{g}_l$ and $\mathbf{h}_l$, respectively. B_l in (4.6) is the half-power bandwidth, $B_l = -2Re\{\lambda_l\}$, and the ω_l in (4.6) is the central frequency of the bell-shaped function $b_l(\omega)$, $\omega_l = Im\{\lambda_l\}$, where $Re\{\cdot\}$ and $Im\{\cdot\}$ denote the real part and the imaginary part of the argument, respectively.

From (4.3), it can be found that the PSD of an MMPP process is constituted by white noise $\bar{\gamma}$, DC component $2\pi\Psi_0$, and a set of bell-shaped functions $b_l(\omega)$ described by the average power Ψ_l, the half-power bandwidth B_l, and the central frequency w_l, with respect to the l-th eigenvalue of $\mathbf{Q}$. The white noise is contributed by the Poisson local dynamics. From the result showed in [33], the influence of the white noise on a queueing system can be neglected.

If the traffic source is further assumed to be an $(M+1)$-state birth-death MMPP, which is a superposition of M independent and identically distributed (iid) two-state MMPPs with parameters (α, β, r_o) as shown in Fig. 4.1, it would have all eigenvalues real and all bell-shaped functions zero-centered. Take a two-state MMPP for example, its time-domain traffic parameters described by (α, β, r_o) is shown in Fig. 4.2(a), and the PSD parameters characterized by $(\bar{\gamma}, B, \Psi)$ are shown in Fig. 4.2(b), where $\bar{\gamma} = \frac{\beta r_o}{\alpha+\beta}$, $B = 2(\alpha+\beta)$, and $\Psi = \frac{\alpha\beta r_o^2}{(\alpha+\beta)^2}$. The PSD of the $(M+1)$-state birth-death MMPP can be obtained by composing PSDs of the M iid two-state MMPPs into a composite power spectrum which is further approximated by an impulse DC component and a single bell-shaped function with parameters:

$$\bar{\gamma} = \frac{M\beta r_o}{\alpha+\beta}, \tag{4.8}$$

$$B = 2(\alpha+\beta), \tag{4.9}$$

$$\Psi = \frac{M\alpha\beta r_o^2}{(\alpha+\beta)^2}. \tag{4.10}$$

The composition algorithm for the two different power spectrums is stated in the appendix content of section 4.6.

Therefore, it can be concluded that: a birth-death MMPP traffic source can be described by its PSD with power-spectrum parameters $(\bar{\gamma}, B, \Psi)$ including the DC component $(\bar{\gamma})$, the half-power bandwidth (B) and the average power (Ψ) of the bell-shaped function. These parameters can be obtained from the $(M+1)$-state MMPP parameters (α, β, r_o). The larger the mean input rate is, the higher the $\bar{\gamma}$ will be; the more correlated the input process is, the smaller the B will be; and the larger the input rate variance is, the higher the Ψ will be. Moreover, PSD parameters of input process possess *additive* property, which does *not* exist in the time-domain traffic parameters.

When a new call request provides its time-domain traffic parameters such as peak bit rate (R_P), mean bit rate (R_M), and average peak bit rate duration (T_D) during the call

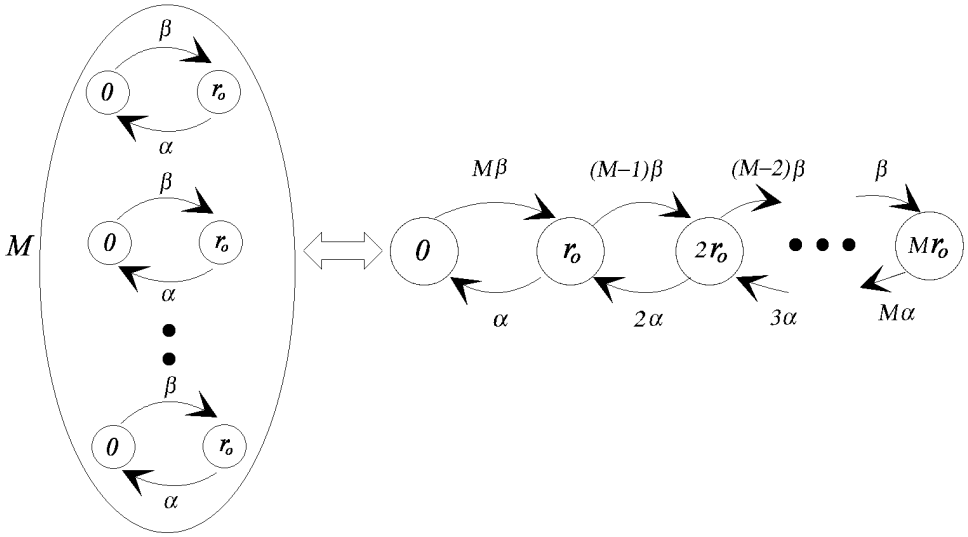


Figure 4.1: The (M+1)-state birth-death MMPP model

establishment phase, the modeled $(M+1)$-state birth-death MMPP process with parameters (α, β, r_o) can be obtained from these traffic parameters (R_P, R_M, T_D) by

$$\alpha = \frac{1}{MT_D}, \tag{4.11}$$

$$\beta = \frac{R_M}{MT_D(R_P - R_M)}, \tag{4.12}$$

$$r_o = \frac{R_P}{M}. \tag{4.13}$$

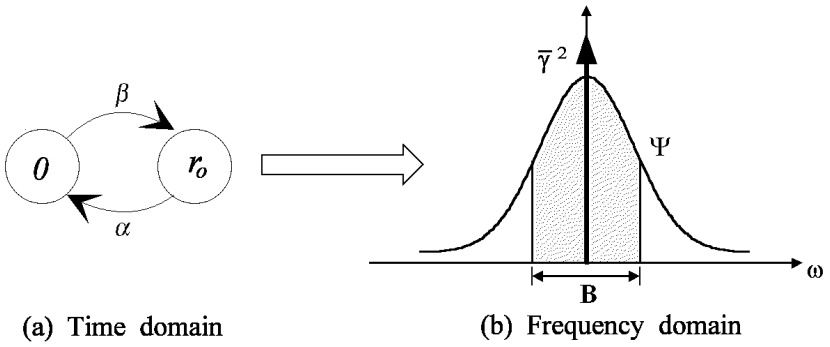


Figure 4.2: The time/frequency-domain parameters of the two-state MMPP

And the power-spectrum parameters $(\bar{\gamma}, B, \Psi)$ of the input traffic of the new call can then be converted from the $(M+1)$-state birth-death MMPP parameters (α, β, r_o) by Eqs. (4.8)-(4.10).

4.3 PSD-based Neural-net Connection Admission Controller

Fig. 4.3 shows the functional block diagram of the PSD-based neural-net connection admission controller. It mainly contains a *PNCAC controller*, a *time/frequency parameters converter*, a *data rate register*, a *PSD parameter register*, a *data rate composer*, and a *power spectrum composer*. Input traffic is assumed to be classified into two types. Type-1 traffic is the real-time traffic such as voice and video, and type-2 traffic is the non-real-time traffic such as data.

As the new call request for type-1 traffic claims its traffic parameters: R_P, R_M, and T_D in the call establishment phase, the *time/frequency parameter converter* transforms the (R_M, R_P, T_D) in time domain into $(\bar{\gamma}, B, \Psi)$ in frequency domain. The *PSD parameter register* keeps the record of the power-spectrum parameters $(\bar{\gamma}_E, B_E, \Psi_E)$ of the total existing type-1 connections, where $\bar{\gamma}_E$ is the total average input rate, B_E is the total half-power bandwidth, and Ψ_E is the total average power. The two sets of parameters $(\bar{\gamma}, B, \Psi)$ and $(\bar{\gamma}_E, B_E, \Psi_E)$ are added to form a new set of parameters $(\bar{\gamma}_T, B_T, \Psi_T)$ through the *power spectrum composer* which performs the power spectrum composition and approximation functions mentioned in section 4.6. If the new call request belongs to type-2 data traffic, it claims the data rate Γ as the traffic parameters in the call establishment phase. The *data rate register* records the overall data rate Γ_E of the existing type-2 connections, and the *data rate composer* adds these two data rate, Γ and Γ_E, to form a new parameter Γ_T. The set of PSD parameters $(\bar{\gamma}_T, B_T, \Psi_T)$ accompanied with the data rate Γ_T is then fed into the *PNCAC*

controller as the input variables. As shown in Fig. 4.4, the PNCAC controller is a multi-layer feedforward neural network [1], [23], which possesses capabilities of approximation to a perfect connection acceptance decision function. And a back-propagation learning algorithm [6] is used here to train the neural network. The PNCAC controller will then decide whether to accept or reject this connection request using the neural-network and feed the decision output (Y) back to the source.

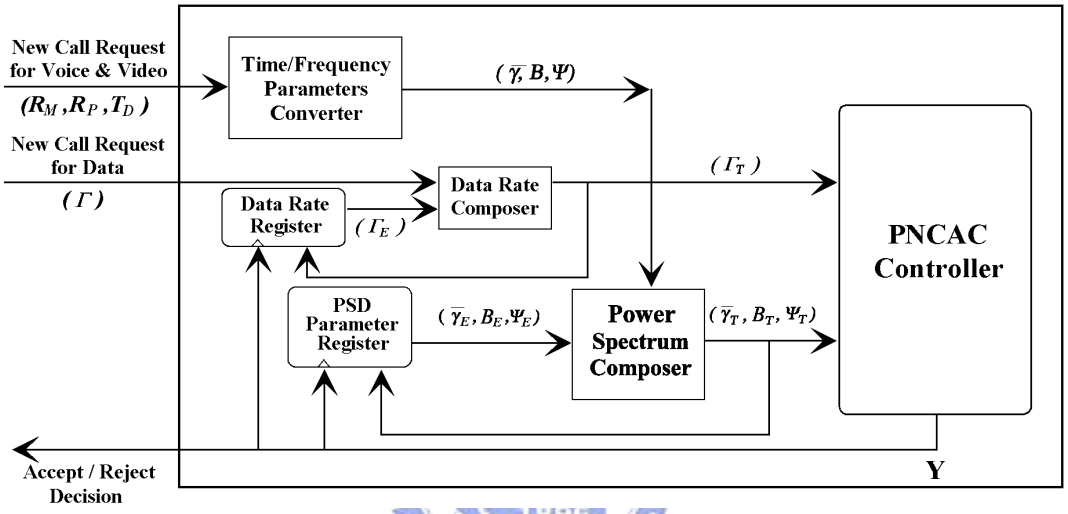


Figure 4.3: The functional block diagram of the PSD-based neural-net connection admission controller

If the decision is to accept the new type-1 call, the *PSD parameter register* will be triggered to update the stored power-spectrum parameters $(\bar{\gamma}_E, B_E, \Psi_E)$ to be $(\bar{\gamma}_T, B_T, \Psi_T)$. So does the type-2 *data rate register* if a type-2 call is accepted. If the decision is to reject the new call, no updating procedure is needed. Notice that $(\bar{\gamma}, B, \Psi)$ or Γ should be subtracted from $(\bar{\gamma}_E, B_E, \Psi_E)$ or Γ_E when a type-1 or type-2 call is disconnected, respectively, which is not shown here.

The implementation of the proposed PNCAC takes about 200 lines of C codes, in which 370 multiplication and 310 addition operations are included. The computation time to make an admission decision would be no more than $500\mu sec$ under general purpose CPU such as

INTEL Pentium-II or above. Therefore, the PNCAC would be feasible in real implementation for high-speed multimedia networks. If special purpose CPU or DSP processors with pipeline architecture or optimized computation capabilities are adopted, less time should be taken to response to a call request. Also, the compiled machine (execution) codes for INTEL CPU is about 20KBytes. The proposed PNCAC scheme can even be downloaded to the embedded systems (platforms).

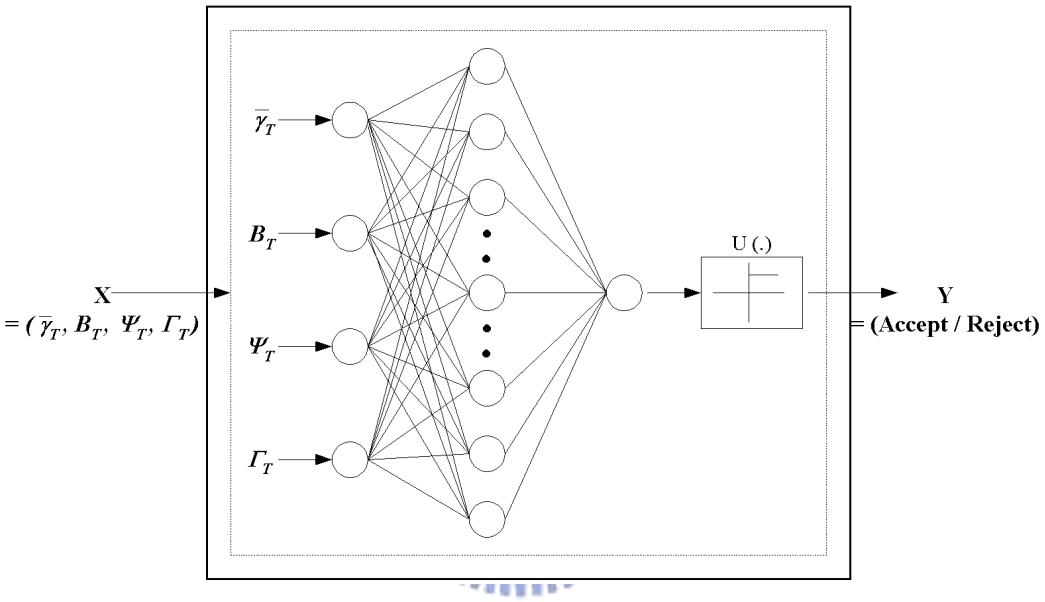


Figure 4.4: The basic structure of the PNCAC controller

4.4 Simulation Results and Discussions

Here we assume that the call admission controller is designed and implemented in an ATM switch/router in multimedia high-speed networks, and input messages are segmented into fixed-length ATM cells. Two separate buffers with buffer size K1 and K2 are for type-1 and type-2 traffic, respectively. One buffer space can accommodate one ATM cell. When the buffer is full, new coming cells are blocked and lost. The service discipline for type-1 and type-2 traffic is that the system initially allocates equal capacity for both types, and the

remaining capacity of one type traffic can be used by the other type traffic.

In the simulations, the buffer sizes K1 and K2 are all set to be 100 cells; the system capacity is assumed to be 150 Mbps. Different QoS requirements for these two types of traffic are defined: the required cell loss probability is set to be 10^{-5} for type-1 traffic and 10^{-6} for type-2 traffic. The voice sources are modeled by a two-state on-off Markov chain (MMPP); the video sources are modeled by a modified Markov process addressed in Chapter 3, where the numbers of video interframe and intraframe are assumed to have five states; and the data sources are modeled by a Poisson process. The traffic parameters for voice and video sources are shown in Table 4.1, and the mean rate for data sources is 1 Mbps. The call arrival rate for voice is 15.4 calls/sec with mean holding time of one minute, the call arrival rate for video is 0.082 calls/sec with mean holding time of five minutes, and the call arrival rate for data is 3.2 calls/sec with mean holding time of 20 seconds. For all traffic types, the *call* arrival processes are assumed to be independently Poisson distributed, and the mean holding time is assumed to be exponentially distributed. Note that in the transformation of the three input time-domain parameters (R_M, R_P, T_D) into PSD parameters $(\bar{\gamma}, B, \Psi)$, both voice and video sources are assumed to be two-state birth-death MMPP.

The neural networks adopted by the PNCAC is a three-layered full-connected feedforward neural network with 50 hidden nodes, as the one used by Hiramatsu's NNCAC which has 30 hidden nodes. It uses 683 and 280 training data and takes about 221,467 and 199,501 iterations to well train the PNCAC and Hiramatsu's NNCAC, respectively.

Table 4.1: Traffic source parameters

Traffic Parameters	Peak Rate	Mean Rate	Peak Rate Duration
Voice	64.0 Kbps	27.6 Kbps	1.366 sec
Video	5.7 Mbps	1.9 Mbps	0.033 sec
Data	-	1.0 Mbps	-

Fig. 4.5 shows the cell loss ratio of type-1 traffic in (a), the cell loss ratio of type-2 traffic in (b), and the system utilization in (c) for the three approaches, where the characteristics of traffic source in simulation are exactly the same as the ones used in training data generation phase. We can find that, after the neural networks have been well trained, both Hiramatsu's NNCAC and PNCAC have larger cell loss ratio than ECCAC, but still guarantee QoS requirements; while Hiramatsu's NNCAC and PNCAC can improve significantly the system utilization over the conventional ECCAC by about 24.4% and 23.8%, respectively. Note that this utilization is obtained by averaging those values between 10^9 and $2*10^9$ slot times. It is due to the learning and adaptive capability of neural networks. Also, Hiramatsu's NNCAC has slightly better system utilization than PNCAC by about 0.6%. This is because Hiramatsu's NNCAC adopts the connection number of each traffic characteristic as the input to decide whether a call request is accepted or not, and the traffic characteristics in simulations are exactly the same as the ones used in training data generation phase for Hiramatsu's NNCAC.

We further consider two simulation examples when the neural networks were well trained according to the traffic characteristics illustrated in Table 4.1, but the system receives heavier and lighter traffic sources with parameters in Table 4.2 and Table 4.3, respectively.

Table 4.2: Heavier traffic source parameters

Traffic Parameters	Peak Rate	Mean Rate	Peak Rate Duration
Voice	64.0 Kbps	40.958 Kbps	1.742 sec
Video	11.4 Mbps	3.8 Mbps	0.033 sec
Data	-	1.5 Mbps	-

Fig. 4.6 shows the cell loss ratio of type-1 and type-2, and the system utilization, for the heavier traffic source, in (a), (b), and (c), respectively. It can be seen that the cell loss ratio of Hiramatsu's NNCAC, denoted by the dashed line, seriously violates the QoS requirements,

Table 4.3: Lighter traffic source parameters

Traffic Parameters	Peak Rate	Mean Rate	Peak Rate Duration
Voice	64.0 Kbps	23.042 Kbps	0.98 sec
Video	2.85 Mbps	0.95 Mbps	0.033 sec
Data	-	0.5 Mbps	-

in both type-1 and type-2 traffic, although the utilization of Hiramatsu's NNCAC is the highest one and approaches 100%. On the other hand, the proposed PNCAC and ECCAC can still fulfill the QoS requirements, and the system utilizations of PNCAC and ECCAC are 85.8% and 77.7%, respectively.

Fig. 4.7 shows the cell loss ratios of type-1 and type-2, and the system utilization, for the lighter traffic source, in (a), (b), and (c), respectively. It can be seen that all the three CAC schemes have zero cell loss ratios and guarantee the required QoS but obtain low system utilizations, compared to those of the normal case shown in Fig. 4.5. The Hiramatsu's NNCAC suffers more degradation and turns out to have the worst system utilization.

From these two simulation examples, it can be concluded that Hiramatsu's NNCAC has worse adaptivity and flexibility than PNCAC and ECCAC. This is because the connection number of each traffic type adopted by Hiramatsu's NNCAC could apply only when the traffic characteristics of traffic sources fed into the operational system are the same as the ones in the training phase. However, this is usually impossible in real practice. As traffic characteristics of sources change, the neural network should learn to adapt to the variation in sources by on-line training, and moreover, the structure of the neural network should be modified to have proper inputs by node-growing or -pruning learning process, if necessary. This would make Hiramatsu's NNCAC infeasible. Because both ECCAC and PNCAC depend on traffic characteristic parameters which can react to the variation in traffic characteristics, these two schemes can adapt to traffic properly without any other modifications or re-training and

still perform the CAC decision well. It is also because the transformed equivalent capacity for ECCAC and the PSD parameters for PNCAC are both unified metrics corresponding to traffic characteristics of all different sources and possess additive property, while the connection number adopted by Hiramatsu's NNCAC as the input variables for neural networks could not be summed for different traffic types.

In addition, the proposed PNCAC has better performance than ECCAC. Both PNCAC and ECCAC depend on traffic characteristic parameters; however, PNCAC transforms the three time-domain traffic characteristic parameters into the corresponding three PSD-domain parameters, while ECCAC converts the same time-domain parameters to a single equivalent capacity. Although the equivalent capacity is also additive, the proposed PNCAC adopts the three PSD parameters as the inputs of neural networks to perform the CAC decision, which could capture more traffic characteristics and less composition approximation error than the single equivalent capacity. The self-learning capability of neural network also makes the PNCAC more adaptive to the traffic.

4.5 Concluding Remarks

In this chapter, we propose a power-spectrum-based neural-net connection admission control (PNCAC) scheme for ATM networks. The PNCAC method adopts the converted power-spectrum parameters of traffic source to represent its traffic characteristics and uses neural network to implement the connection admission control. The frequency-domain power-spectrum parameters of traffic source possess additive property and can capture the correlation and burstiness behavior more than the time-domain parameters such as peak rate, mean rate, and peak rate duration. The neural network has the learning/adapting capabilities so that the boundary of the decision hyperplane for the connection admission control can be adjusted optimally and dynamically. We demonstrate results whenever the

input voice and video traffic sources are modeled by MMPP and modified MMPP, respectively, and the data traffic sources are modeled by a Poisson process. Simulation results show that the proposed PNCAC enhances significantly the system utilization while fulfilling QoS requirements. Not only it is superior to the conventional equivalent capacity CAC scheme (ECCAC), it also obtains more flexibility and robustness than Hiramatsu's NNCAC.

4.6 Appendix: Composition Algorithm for Power Spectrums

Assume that $b_1(\omega)$ and $b_2(\omega)$ are two bell-shaped functions corresponding to zero-centered PSDs with parameters $(\bar{\gamma}_1, B_1, \Psi_1)$ and $(\bar{\gamma}_2, B_2, \Psi_2)$, as shown in Fig. 4.8, and $b(\omega)$ is the approximated bell-shaped function corresponding to the composite power spectrum with parameters $(\bar{\gamma}, B, \Psi)$. To compose the two zero-centered PSDs, we add the two DC components and the two bell-shaped functions directly. We then approximate $b_1(\omega) + b_2(\omega)$ to be $b(\omega)$. In the approximation, we set $\Psi = \Psi_1 + \Psi_2$, and $b(\omega) = b_1(\omega) + b_2(\omega)$ at $\omega = 0$. Therefore, $(\bar{\gamma}, B, \Psi)$ of the approximated power spectrum are given by

$$\bar{\gamma} = \bar{\gamma}_1 + \bar{\gamma}_2, \tag{4.14}$$

$$B = \frac{(\Psi_1 + \Psi_2) B_1 B_2}{\Psi_1 B_2 + \Psi_2 B_1}, \tag{4.15}$$

$$\Psi = \Psi_1 + \Psi_2. \tag{4.16}$$

Note that the approximated bell-shaped function contains more low-frequency components than $b_1(\omega) + b_2(\omega)$.

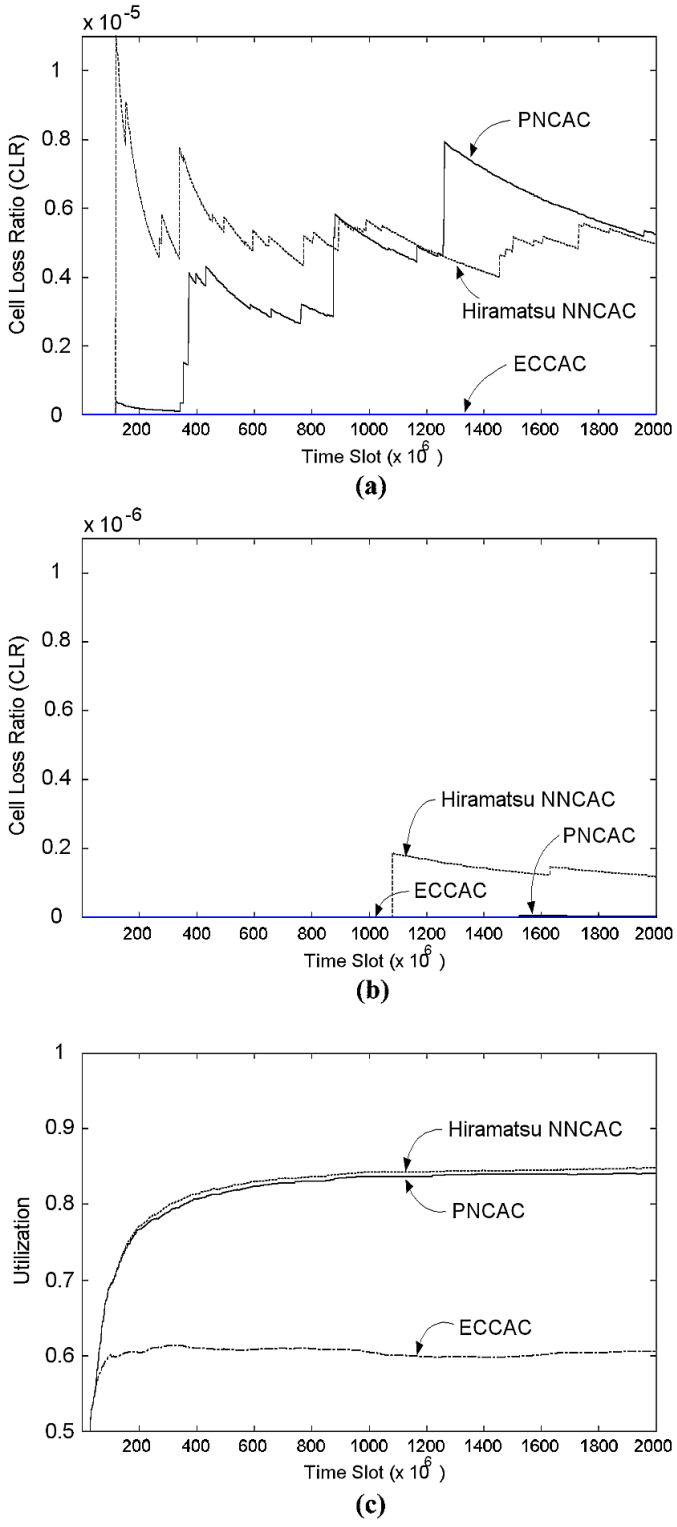


Figure 4.5: (a) The type-1 cell loss ratio (CLR), (b) the type-2 cell loss ratio (CLR), and (c) the system utilization of the ECCAC, Hiramatsu's NNCAC, and PNCAC

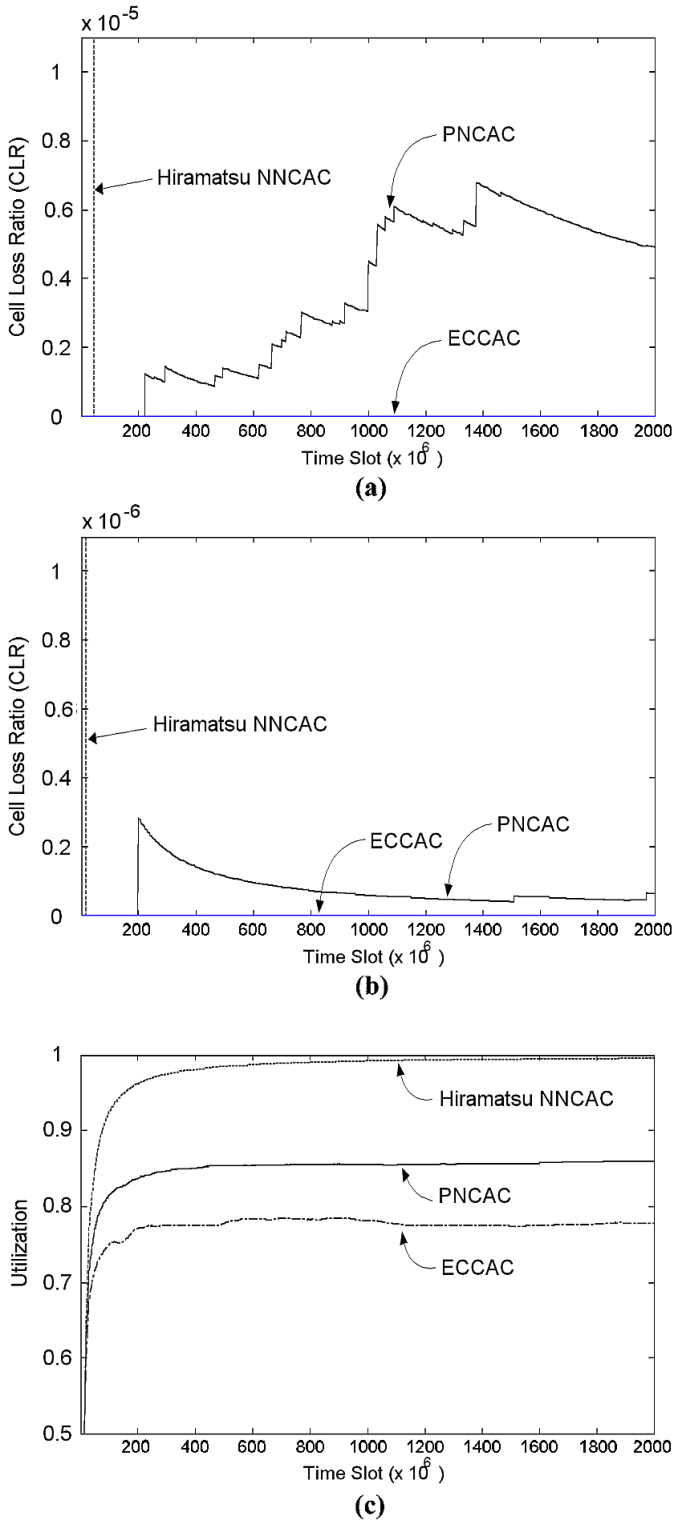


Figure 4.6: (a) The type-1 cell loss ratio (CLR), (b) the type-2 cell loss ratio (CLR), and (c) the system utilization of the ECCAC, Hiramatsu's NNCAC, and PNCAC with heavier traffic sources

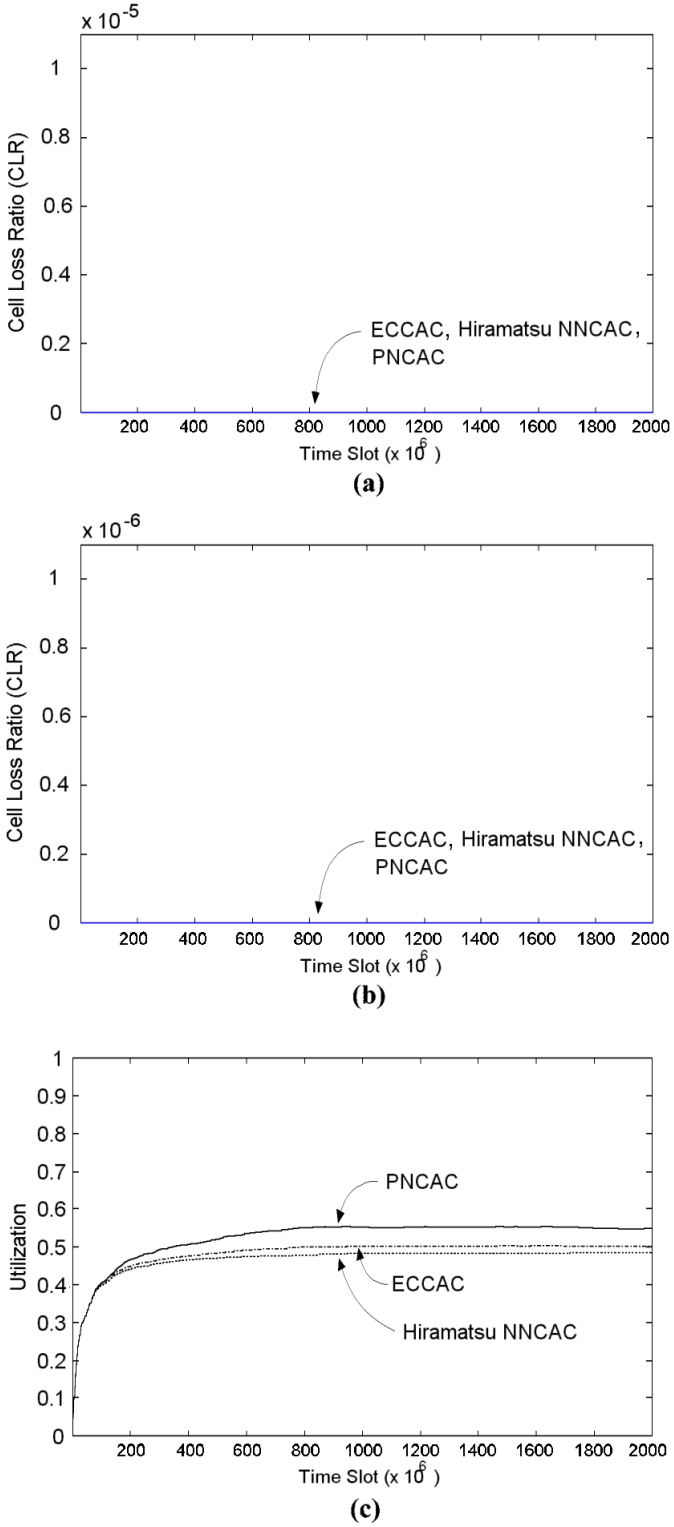


Figure 4.7: (a) The type-1 cell loss ratio (CLR), (b) the type-2 cell loss ratio (CLR), and (c) the system utilization of the ECCAC, Hiramatsu's NNCAC, and PNCAC with lighter traffic sources

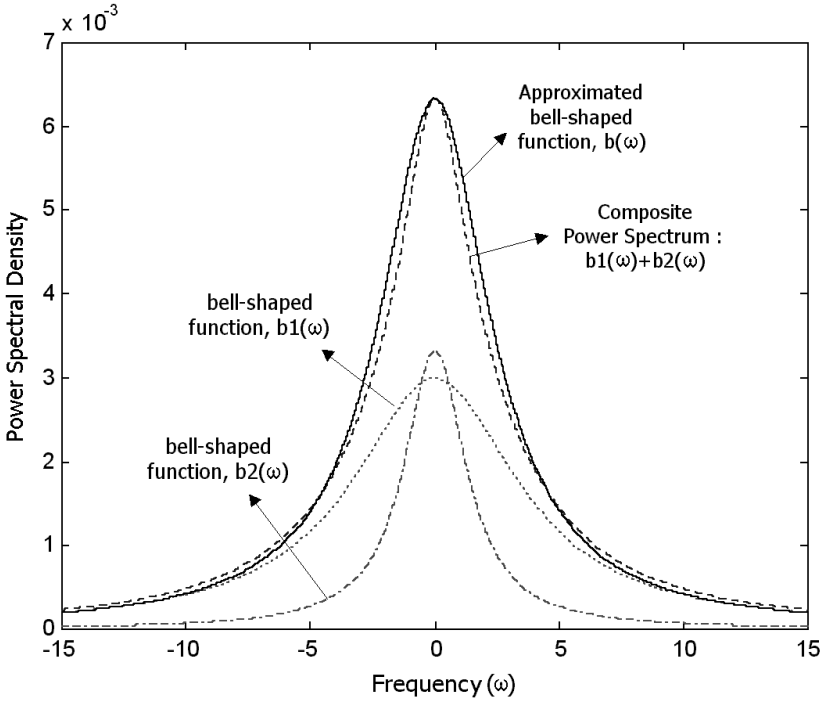


Figure 4.8: The approximated bell-shaped function of the composed bell-shaped PSD

Chapter 5

An Intelligent Usage Parameter Controller for Multimedia High-speed ATM Networks

For CAC to perform correctly, a traffic policing mechanism is necessary to ensure that all established connections conform to their respective traffic contracts. Therefore, in this chapter, two intelligent usage parameter controllers are proposed to implement the traffic policing function for multimedia transmissions in ATM networks. One is the fuzzy usage parameter controller realized by the fuzzy leaky bucket algorithm, in which a fuzzy increment controller (FIC) is incorporated with the conventional leaky bucket algorithm; the other is the neural fuzzy usage parameter controller base on the neural fuzzy leaky bucket algorithm, where a neural fuzzy increment controller (NFIC) is added to the conventional leaky bucket algorithm. Both of FIC and NFIC properly choose the measured long-term and short-term mean cell rates, as input variables to adaptively determine the optimal increment value. Simulation results show that both intelligent leaky bucket algorithms have significantly outperformed the conventional one by responding about 160% *faster when taking control actions against a non-conforming connection, while reducing as much as* 50% *of the queueing delay experienced by a conforming connection. In addition, the neural fuzzy leaky bucket algorithm outperforms the fuzzy one especially in the aspect of responsiveness.*

5.1 Introduction

The emergence of multimedia services has diversified the quality-of-service (QoS) and bandwidth requirements for communication services. Asynchronous transfer mode (ATM) is considered as a suitable technique to meet the diverse requirements. Several traffic control mechanisms are recommended for ATM networks [31]. Among them, connection admission control (CAC) and usage parameter control (UPC) are most important.

CAC is performed at the call setup phase to decide whether the connection can be accepted or not. It accepts the connection if the required bandwidth and QoS of the connection can be afforded while QoS of existing connections can still be maintained. A traffic contract, which specifies traffic descriptors such as the *peak cell rate* (PCR), the *sustainable cell rate* (SCR) and the *maximum burst size* (MBS), would then be built between the accepted connection and the network. For CAC to perform correctly, all the established connections must not violate their respective traffic contracts which are of vital importance to the decision making of CAC. To make sure that the established connections conform to their traffic contracts, the UPC mechanism which is the traffic policing function defined in ATM networks should be employed and co-operate with the CAC.

UPC is performed at the user-network interface (UNI) during the data transfer phase. It is defined as the set of actions taken by the network to police the offered traffic of a connection so that the negotiated traffic contract is respected. That is, some portion of the traffic of a connection would be dropped or shaped (by introducing queueing effect) to enforce the resultant traffic compliant with the traffic profile negotiated in the traffic contract during the call setup phase. Sometimes, the non-conforming portion of a connection would be tagged rather than directly dropped, so that the residual traffic satisfy the contract and some future processing would be performed upon the tagged non-conforming traffic to attain some operation objectives. The main purpose of UPC is to protect network resources from

malicious as well as unintentional misbehavior which can affect the QoS of other already established connections.

Monitoring and controlling PCR of a connection is not difficult because we only have to determine if the peak emission interval is smaller than the reciprocal of the negotiated PCR, Λ_{PCR}. However, policing the SCR of a connection is much more complicated because the connection is eligible to transfer cells with a short-term mean rate higher than the negotiated SCR, Λ_{SCR}, while the long-term mean rate of the connection still conforms to Λ_{SCR}. The difficulty of the UPC to control the SCR of a connection lies in finding a simple, universal, and effective scheme which is able to police any type of traffic to meet the long-term SCR specified in its traffic contract by making short-term and packet-level processing decision upon each incoming packet. Besides, the wide variety of multimedia services with different traffic characteristics and QoS requirements would further complicate the UPC especially for the SCR policing. Since the SCR policing is not a easy job, we here concentrate on the UPC for the SCR of the offered connection.

In this chapter, we assume that a traffic shaper (TS) is equipped within the customer premise equipment (CPE) to regulate the cell stream of the traffic source so as to conform the negotiated SCR. The regulation is to alter the traffic characteristics of the cell stream to achieve a desired traffic shape. However, the consequence of the regulation would cause an increase in the mean cell transfer delay. The conjunction of TS and UPC, named as TS-UPC, should employ identical schemes with same parameters settings in both TS and UPC so that any possible illegal cell that might have been detected as non-conforming by UPC will be detected ahead of time and saved in the queue by TS. In this way, the TS-UPC can guarantee zero cell loss ratio at UPC for a non-violating connection. Nevertheless, if a user intentionally or unintentionally changes the parameters settings in TS and illegally enjoys a higher bit-rate service there, UPC will detect the violation and take actions against it. The

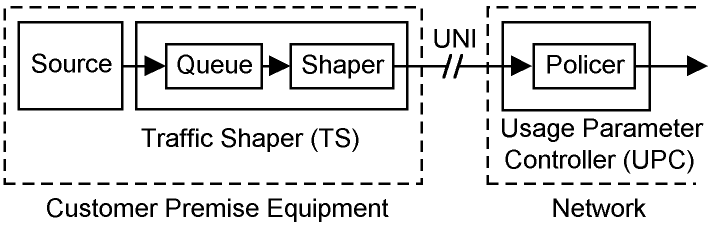


Figure 5.1: The connection model

primitive connection model with TS-UPC is shown in Fig. 5.1. The component attached to the traffic source is the TS which contains a shaper and a queue. It bypasses the conforming cells but stores the non-conforming cells in the queue for further legal transmission. The component at the entrance of the network is UPC, where a policer is incorporated. It bypasses the conforming cells but drops or tags the non-conforming cells.

Three performance objectives have to be fulfilled by TS-UPC and they can also be adopted as the criteria to evaluate the efficiency of the TS-UPC in ATM networks: (i) *High selectivity (detection accuracy)*: UPC should detect and tag (drop) the non-conforming cells of a violating connection as many as possible, while being transparent when the connection conforms to its traffic contract. (ii) *High responsiveness*: the time for UPC to detect a violating connection should be rather short. (iii) *Low queueing delay*: cells of a non-violating connection should not experience too much queueing delay at TS. However, the queueing delay introduced by TS on a violating connection is beyond our consideration.

Several UPC schemes such as the jumping window, triggered jumping window, moving window, exponentially weighted moving average, and leaky bucket algorithm were studied and compared [40], [41], [42], [43]. The most popular and well-known policing scheme is the leaky bucket algorithm because of its simplicity and effectiveness. The conventional TS-UPC using the leaky bucket algorithm recommended in ITU-T I.371 has a crisp structure with two fixed parameters of *threshold* and *increment*. It uses parametric model to analyze, thus

resulting in the lack of real multimedia traffic information which are dynamic, imprecise, non-linear, and even non-stationary. Generally speaking, it is difficult for networks to acquire complete statistics of input traffic. Therefore, it is not easy to accurately determine the threshold or the increment in the multimedia traffic flows. The rationale and principles underlying the nature and choice of the threshold or the increment under dynamic and bursty conditions are unclear. As a result, the decision process of the network is based on incomplete information and full of uncertainty.

The *fuzzy logic system* and *neural network* are both numerical model-free estimators and dynamical systems [1]. The fuzzy set theory appears to provide a robust mathematical framework for dealing with real-world imprecision. Its approach exhibits a soft behavior which has the capability to adapt to dynamic, imprecise, and bursty environments. The fuzzy logic system can represent information in a way that resembles natural human communication, and can handle the information in a way similar to human reasoning [3], [17], [18]. It would be an intelligent implementation that not only refers to the mathematical formulation of classical control but also mimics expert knowledge in traffic control. Neural networks are trainable systems that demonstrate the ability to learn, recall, and generalize from training patterns or data. Through learning, neural networks can predict non-linear complex functions, thus making themselves effective tools to be employed in ATM networks for traffic modeling and prediction [1].

In recent years, neural network research has pursued either by a pre-structuring of the neural network to improve its performance, or by a possible interpretation of the synaptic matrix following the learning stage; and fuzzy logic research has pursued the development of methods for automatic tuning of the parameters which characterize the fuzzy control system. Notice that the fuzzy set theory possesses no clear, general technique to map expert knowledge of traffic control onto the design parameters of the fuzzy logic controller. Hence,

one approach that gets benefits of neural networks and fuzzy logics and solves their respective problems, called *neural fuzzy network*, is developed. The neural fuzzy network integrates the fuzzy logics within a neural network. The integration brings the low-level learning and computational power of the neural network into the fuzzy logic system, and provides the high-level, human-like thinking and reasoning of fuzzy logic system into the neural network. The neural fuzzy network generally takes the form of a multi-layer neural network to realize a fuzzy logic system. It is a *structured* neural network that can incorporate domain knowledge from conventional policies; and it not only provides a robust framework to mimic experts' knowledge embodied in existing traffic control techniques but also constructs intelligent computational algorithm for traffic control [1].

Some literature had also studied to utilize the intelligent techniques for the UPC [44], [45], [46], [47]. In [44], a fuzzy logic implementation of the leaky bucket algorithm that used a channel utilization feedback to manage voice cells in ATM networks was proposed. Simulation results showed that the fuzzy leaky bucket had performance improvement over the conventional leaky bucket algorithm. In [45], a neural network traffic enforcement mechanism using window-based scheme for ATM networks was presented. It is based upon an accurate estimation of the probability density function (pdf) of the traffic via a counting process, and the system performance is evaluated in terms of the pdf violation. It has scalability and convergence problems if the number of previous windows is required to be a large value. In [46], the paper designed a fuzzy traffic policer based on window control scheme, which has the characteristic of simplicity and the capability to combine a fast responsiveness with a high-degree selectivity close to that of an ideal traffic policer. In [47], the proposed policing strategy integrated with a linear prediction filter is used to forecast the cell rate of the policed traffic source.

In this chapter we propose two *intelligent* UPC schemes to perform the sustainable-cell-

rate usage parameter control of multimedia transmissions in ATM networks. One is the *fuzzy* TS-UPC realized by the fuzzy leaky bucket algorithm, in which a fuzzy increment controller (FIC) is incorporated with the conventional leaky bucket algorithm for dynamic increment adjustment. Two system parameters, the long-term mean cell rate and the short-term mean cell rate, of a connection are fed into the FIC to adaptively calculate the appropriate increment. Simulation results show that the fuzzy TS-UPC can have higher selectivity, faster responsiveness, and smaller queueing delay than the conventional TS-UPC, as anticipated. The other proposed intelligent UPC scheme is the *neural fuzzy* TS-UPC base on the neural fuzzy leaky bucket algorithm, where a neural fuzzy increment controller (NFIC) is added to the conventional leaky bucket algorithm to dynamically adjust the increment. Neural fuzzy network is a *structured neural network*; it integrates intelligent learning and computation of neural networks with fuzzy logic systems. Also, the reinforcement learning is here applied for NFIC since we cannot measure the desired increment. Simulation results show that the neural fuzzy TS-UPC performs further better than the fuzzy TS-UPC in the above-mentioned performance measures of selectivity, responsiveness, and queueing delay, especially as the multimedia traffic flows are more bursty, dynamic, and non-stationary.

The chapter is oriented as follows. In Section 5.2, we provide an introduction of the leaky bucket algorithm recommended in ITU-T I.371 for conventional TS-UPC and the problems it encounters. In Section 5.3, we describe the proposed fuzzy leaky bucket algorithm for the fuzzy TS-UPC. In Section 5.4, we describe the proposed neural fuzzy leaky bucket algorithm for the neural fuzzy TS-UPC. The performance measures of selectivity, responsiveness, and queueing delay for the conventional TS-UPC, the fuzzy TS-UPC, and the neural fuzzy TS-UPC are compared in Section 5.5. Finally, some concluding remarks are presented in Section 5.6.

5.2 Leaky Bucket Algorithm

ITU-T Recommendation I.371 [31] recommends the Generic Cell Rate Algorithm (GCRA) as a conformance test for the cell stream of a connection. GCRA has two equivalent versions – the virtual scheduling algorithm and the leaky bucket algorithm. The latter seems to be better comprehended since it can be pictured as a virtual leaky bucket whose content determines the conformance of a cell. As shown in Fig. 5.2, the leaky bucket is viewed as a finite capacity bucket whose real-valued content drains out at one unit rate but is increased by T units for each conforming cell. If a cell arrives at the time when the bucket content X' is above the threshold value τ, then the cell is non-conforming; otherwise, the cell is conforming and the bucket content is added by an increment T.

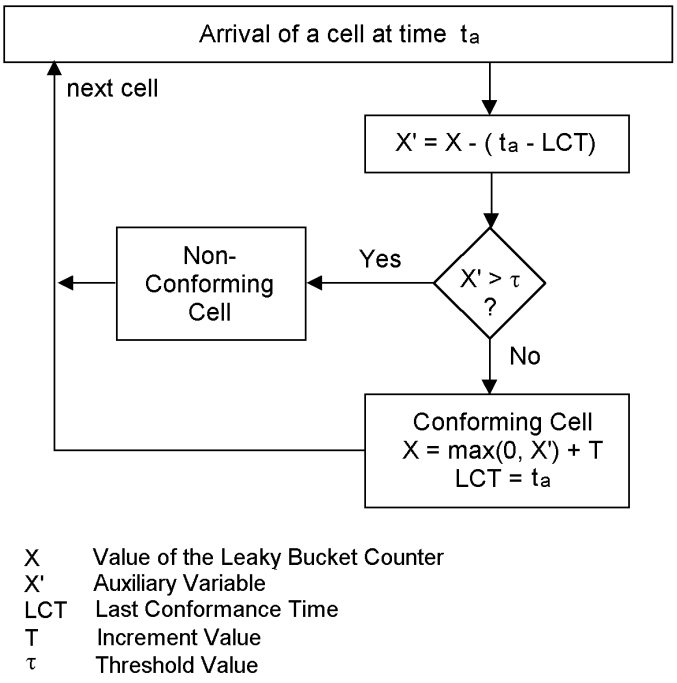


Figure 5.2: The flow chart of the conventional leaky bucket algorithm

The conventional TS-UPC employs the leaky bucket algorithm in the shaper and the policer as their schemes to monitor the sustainable cell rate of a connection. The threshold

value τ is taken to be $\tau_{IBT} + \tau'_{SCR}$ and the increment T is taken to be the reciprocal of the negotiated sustainable cell rate Λ_{SCR} of the connection, where τ_{IBT} is the intrinsic burst tolerance (IBT) used to limit the burst size to the negotiated maximum burst size (MBS) and τ'_{SCR} is an additional tolerance added to account for the cell delay variation (CDV) introduced by multiplexing schemes. Details of the two parameters τ_{IBT} and τ'_{SCR} can be found in the ITU-T Recommendation I.371.

If Λ_{SCR} is set to be the mean cell rate Λ_{mean} for the TS-UPC, then the possible rate fluctuations of the connection around the claimed mean cell rate will cause the leaky bucket within TS to detect some non-conforming cells. These detected non-conforming cells are stored in the queue, resulting in a long queueing delay. The undesirable long queueing delay can be avoided by making the bucket threshold τ in TS and UPC deviate from $\tau_{IBT} + \tau'_{SCR}$ to a large value. Unfortunately, a higher τ would cause the slower response time for UPC. Another solution without changing the bucket threshold is to make Λ_{SCR} be Λ_{mean} multiplied by a magnifying factor C, $C > 1$. By doing this, we can eliminate the retardation provoked by a higher τ. However, it has a risk of letting a connection with small rate fluctuations, e.g. a CBR connection, enjoy bandwidth higher than that negotiated. There are an infinite number of admissible couples of values for Λ_{SCR} and τ. The detailed analysis for the selection of Λ_{SCR} and τ and the consequent system performance can be found in [48].

5.3 Fuzzy Leaky Bucket Algorithm

Fig. 5.3 shows an intelligent leaky bucket algorithm which contains the conventional leaky bucket algorithm (enclosed by the dashed line) incorporated with an intelligent increment controller (IIC). The first intelligent leaky bucket algorithm we proposed is the fuzzy leaky bucket algorithm which employs a fuzzy increment controller (FIC) to implement IIC. FIC is a fuzzy logic controller and is designed to dynamically adjust T, instead of using a fixed $T =$

$1/\Lambda_{SCR}$, so that the selectivity, responsiveness, and queueing delay can be optimally achieved. The reason we use the fuzzy logic system to implement the increment controller is that the fuzzy logic can represent information in a way resembling natural human communication and handle the information in a way similar to human reasoning [3]. The domain knowledge for the adjustment of T is as follows. When the cell stream of a connection appears to be violating the negotiated sustainable cell rate, T should be adjusted to be big so that the leaky bucket can *quickly* detect the non-conforming cells; while in contrast, when the cell stream of a connection appears to be conforming or conservative to the sustainable cell rate, T should be adjusted to be reasonably small so that no cell of the connection will be detected as non-conforming cells by the fuzzy leaky bucket (i.e., the leaky bucket would be transparent to the connection).

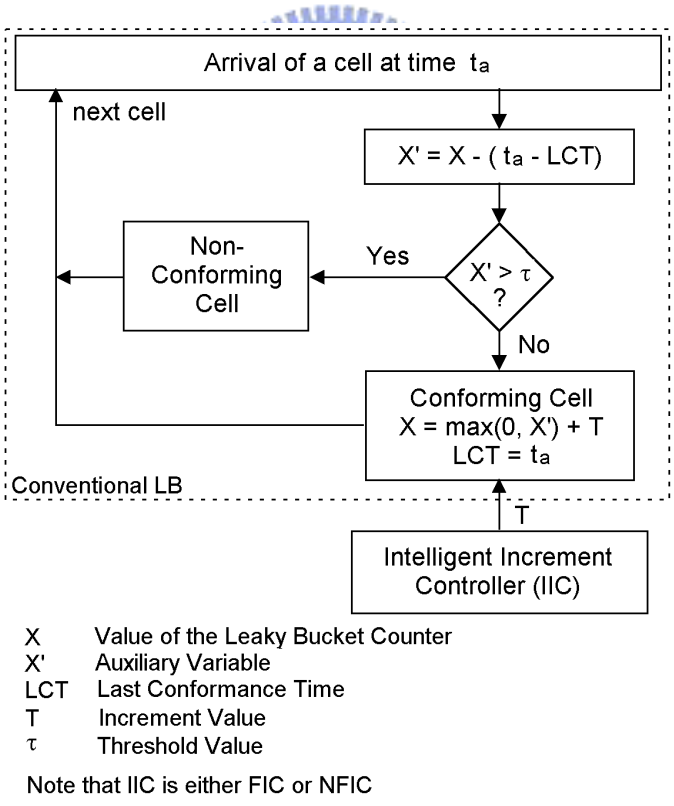


Figure 5.3: The intelligent leaky bucket algorithm

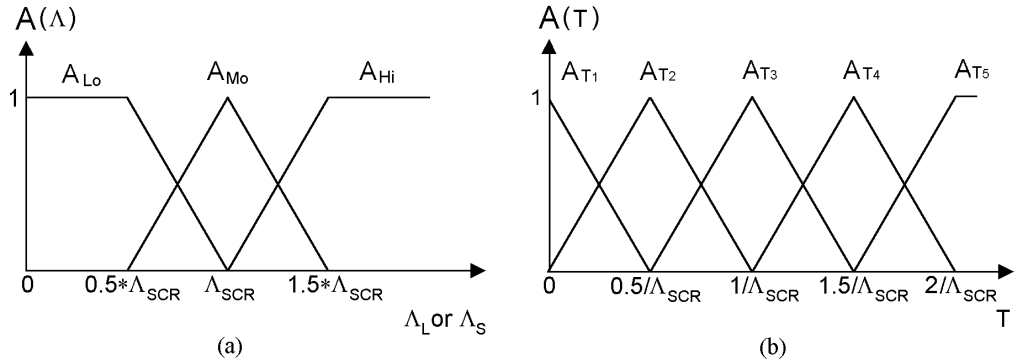


Figure 5.4: (a) The membership functions for the input variables Λ_L and Λ_S (b) The membership functions for the output variable T

We choose two input variables for FIC – the long-term mean cell rate Λ_L and the short-term mean cell rate Λ_S of the connection being policed. The long-term mean cell rate is defined as the average cell rate of a connection since the beginning of the connection, and the short-term mean cell rate is defined as a moving average cell rate in a time window. Λ_L and Λ_S are used to provide an indication of the conformance degree of the connection. At the arrival of a cell, the statistics of Λ_L and Λ_S are fed into FIC to obtain an optimal increment T.

We design Λ_L and Λ_S to have the same term set – $\{Low, Moderate, High\}$. And let T have five terms – $\{VerySmall\ (T_1), Small\ (T_2), Medium\ (T_3), Big\ (T_4), VeryBig\ (T_5)\}$. Fig. 5.4(a) and Fig. 5.4(b) show the membership function of the input and output variables, respectively. The membership functions for $\{Low, Moderate, High\}$ are denoted by $\{A_{Lo}, A_{Mo}, A_{Hi}\}$; the membership function for $\{T_1, T_2, T_3, T_4, T_5\}$ are denoted by $\{A_{T_1}, A_{T_2}, A_{T_3}, A_{T_4}, A_{T_5}\}$. These membership functions in the figures are represented by either triangular or trapezoidal functions, which have the advantage of simple computational complexity.

The rule base is designed according to the domain knowledge on how the fuzzy increment controller should behave. For example, the knowledge and experience tell us: when both Λ_L

Table 5.1: The rule base for FIC

Rule	Λ_L	Λ_S	T
1	Low	Low	Very Small (T_1)
2	Low	Moderate	Very Small (T_1)
3	Low	High	Small (T_2)
4	Moderate	Low	Small (T_2)
5	Moderate	Moderate	Medium (T_3)
6	Moderate	High	Big (T_4)
7	High	Low	Big (T_4)
8	High	Moderate	Very Big (T_5)
9	High	High	Very Big (T_5)

and Λ_S are lower than Λ_{SCR}, FIC should generate a very small T so that the connection can enjoy a higher cell rate later because the connection is likely to be too conservative; when both Λ_L and Λ_S are higher than Λ_{SCR}, the connection is likely to violate the negotiated sustainable cell rate and FIC should generate a very big T so that the violation will be detected quickly. The inference rule base is shown in Table 5.1. Below is an example of how the rules should be read.

Rule 1: If (Λ_L is Low) and (Λ_S is Low), then (T is Very Small).

The proposed FIC is then implemented by the Mamdani fuzzy model introduced in section 2.2. The linguistic values of T_1, T_2, T_3, T_4, and T_5 of the output linguistic variable T are defined over a discrete universe of discourse having 65,536 points. The inference method adopts *min-max* scheme. Take rule 1 and rule 2 which have the same term $VerySmall$ (T_1) for example. In the first step, the *min-max* inference method applies the *min* operator on membership values of associated term of all the input linguistic variables for each rule. We denote the firing strength of rule 1 and rule 2 by w_1 and w_2:

$$w_1 = min(A_{Lo}(\Lambda_L), A_{Lo}(\Lambda_S)) \tag{5.1}$$

$$w_2 = min(A_{Lo}(\Lambda_L), A_{Mo}(\Lambda_S)). \tag{5.2}$$

Then applying the max operator between w_1 and w_2 yields the overall membership value of T_1, denoted by:

$$w_{T_1} = max(w_1, w_2). \tag{5.3}$$

The defuzzification method uses the *centroid of area* mechanism to obtain T:

$$T = \frac{\sum_{i=1}^{n} A_T(z_i) * z_i}{\sum_{i=1}^{n} A_T(z_i)}, \tag{5.4}$$

where n is the number of points of the output, $n = 65536$, z_i is the amount of control output at point i, and $A_T(z_i)$ represents its membership value in the output term set $\{T_1, T_2, T_3, T_4, T_5\}$ [3], which is given by

$$A_T(z_i) = \max_{j \in [1,5]} [min(w_{T_j}, A_{T_j}(z_i))]. \tag{5.5}$$

After FIC is built, the membership functions can be manually fine-tuned by observing the progress of simulation. The tuning can be done with different objectives, such as the response time and queueing delay. Any gain in response time must be traded off by a possible increase in the queueing delay experienced by a cell. However, since the tuning of the membership functions is intuitive, it is easy to achieve an appropriate balance between an acceptable queueing delay and a satisfactory responsiveness. The final control surface of FIC is shown in Fig. 5.5.

5.4 Neural Fuzzy Leaky Bucket Algorithm

The second intelligent leaky bucket algorithm we proposed is the neural fuzzy leaky bucket algorithm which employs a neural fuzzy increment controller (NFIC) to realize IIC. The NFIC is a neural fuzzy controller and is also expected to dynamically adjust T to achieve better performances on selectivity, responsiveness and queueing delay. The NFIC also chooses the long-term mean cell rate Λ_L and the short-term mean cell rate Λ_S as input variables and the increment T as the output variable; it adopts the same term sets for Λ_L, Λ_S, T and the

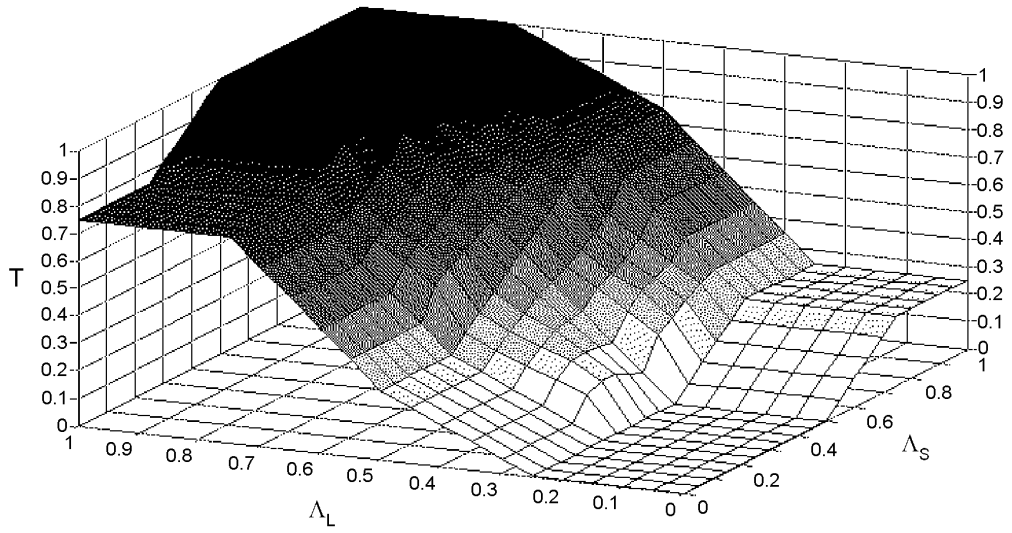


Figure 5.5: The control surface of FIC

same rule base as those employed by FIC, before the learning. However, the membership functions for all terms are chosen to be the bell-shaped function as defined in Eq. (2.19) in section 2.4.

5.4.1 Structure of NFIC

Here, as shown in Fig. 5.6, the five-layer neural fuzzy controller introduced in section 2.4 is also adopted to implement the NFIC. The nodes in layer one and layer five are *input* and *output linguistic nodes*, respectively. Two input linguistic nodes exist in layer one for input linguistic variables Λ_L and Λ_S. There are two kinds of output linguistic nodes: one is for feeding training data (desired output) r into the net and the other is for pumping decision signals (actual output) T out of the net. The nodes in layer two and layer four are *term nodes* which are respectively corresponding to a linguistic term of the input linguistic variables, and perform the fuzzification function to map the crisp input into a fuzzy membership value according to its associated membership function. As mentioned earlier, Λ_L and Λ_S have the same term

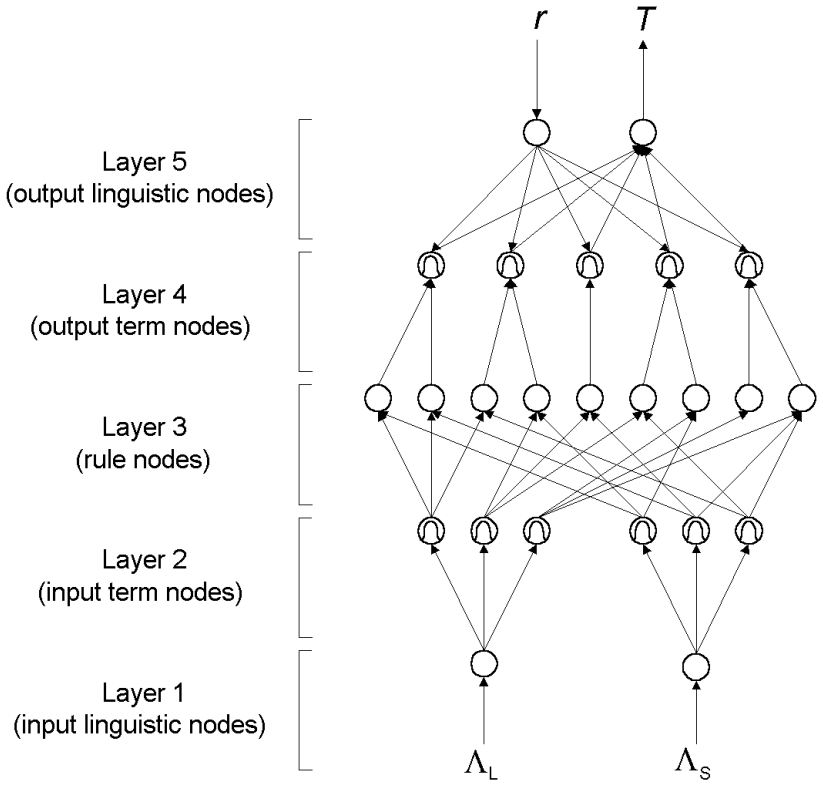


Figure 5.6: The structure of the neural fuzzy increment controller (NFIC)

set – $\{Low, Moderate, High\}$, thus we have six nodes in layer two, and five nodes in layer four for the term set $\{VerySmall\ (T_1), Small\ (T_2), Medium\ (T_3), Big\ (T_4), VeryBig\ (T_5)\}$ of T. The nodes in layer three are *rule nodes*; each node represents one fuzzy rule and all nodes form a fuzzy rule base. There are nine nodes in layer three with respect to the rule base shown in Table 5.1. The links in layer three and layer four, accompanied by the nodes in both layers, can function as an inference engine — layer-three links define preconditions of the rule nodes and layer-four links define consequences of the rule nodes. Thus, the links and nodes in layer three would execute the fuzzy AND operation while the links and nodes in layer four perform the fuzzy OR operation to integrate the fired strength of rules that have the same consequence. Accordingly, the fuzzy reasoning is done by the nodes and links in both layer three and four. The links in layer two and layer five are fully connected between

the linguistic nodes nodes and their respective term nodes. They can, accompanied by the nodes in both layers, achieve the fuzzification and defuzzification functions, respectively.

5.4.2 Reinforcement Learning

The diagram of the reinforcement learning for NFIC is shown in Fig. 5.7, where the ATM system offers statistics of Λ_L and Λ_S input to NFIC, provides a reinforcement signal r as a desired output to NFIC, and receives the updating increment value T from NFIC. The reinforcement signal is defined as

$$r = P_d^* - P_d, \tag{5.6}$$

where P_d^* denotes the desired cell loss ratio and P_d is the actually measured cell loss ratio.

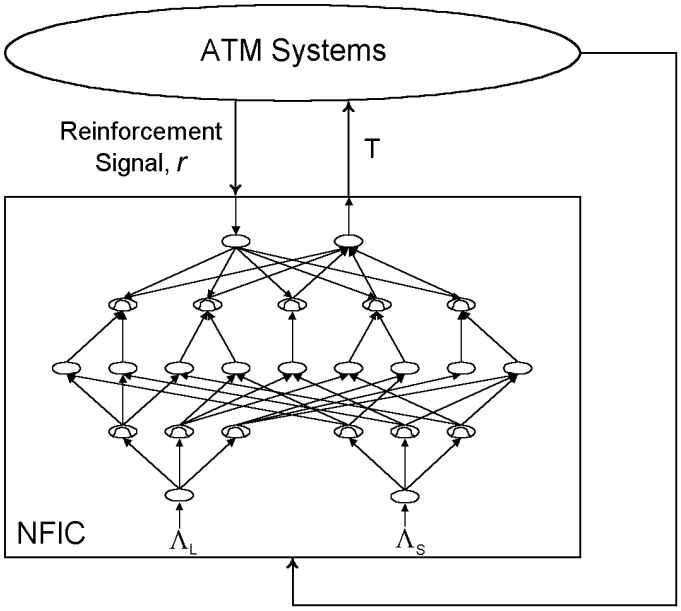


Figure 5.7: The configuration of the reinforcement learning for NFIC

Based on this connectionist structure established in Fig. 5.6, the reinforcement learning is applied to optimally adjust parameters of input and output membership functions, according to the input training data, the reinforcement signal, the fuzzy partition, and the fuzzy logic

rules. It derives updating rules for the mean and the standard deviation of the bell-shaped membership functions so as to minimize the error function, defined as

$$E = \frac{1}{2}r^2 = \frac{1}{2}(P_d^* - P_d)^2. \tag{5.7}$$

For each training data set, starting at the input nodes, the $down-up$ operation can compute to obtain the actual output of increment T. On the opposite direction, starting at the output node, the $up-down$ operation is used to compute $\frac{\partial E}{\partial w}$ for all hidden nodes, where w is the adjustable parameters such as the mean and the standard deviation for the input and output bell-shaped membership functions. We adopt the general learning rule to do the adjustment, which is given by

$$w(n+1) = w(n) + \eta \cdot (-\frac{\partial E}{\partial w}), \tag{5.8}$$

where η is the learning rate. The updating rules for parameters are layer by layer listed below.

Layer 5: The updating rule for $m_j^{(O)}$ in this layer can be obtained by

$$m_j^{(O)}(n+1) = m_j^{(O)}(n) + \eta \cdot r \cdot \frac{\sigma_j^{(O)} u_{ij}^{(5)}}{\sum_j \sigma_j^{(O)} u_{ij}^{(5)}},\ 1 \le j \le 5, \tag{5.9}$$

and the updating rule for $\sigma_j^{(O)}$ is given by

$$\sigma_j^{(O)}(n+1) = \sigma_j^{(O)}(n) + \eta \cdot r \cdot \frac{m_j^{(O)} u_{ij}^{(5)}(\sum_j \sigma_j^{(O)} u_{ij}^{(5)}) - u_{ij}^{(5)}(\sum_j m_j^{(O)} \sigma_j^{(O)} u_{ij}^{(5)})}{(\sum_j \sigma_j^{(O)} u_{ij}^{(5)})^2}. \tag{5.10}$$

The error signal, $\delta^{(5)}$, to be propagated to the proceeding layer, is given by

$$\delta^{(5)} = r. \tag{5.11}$$

Layer 4: In this layer, only the error signal, $\delta_i^{(4)}$, needs to be computed and propagated. $\delta_i^{(4)}$ is derived as

$$\delta_i^{(4)} = r \cdot \frac{m_i^{(O)} \sigma_i^{(O)}(\sum_i \sigma_i^{(O)} u_i^{(5)}) - \sigma_i^{(O)}(\sum_i m_i^{(O)} \sigma_i^{(O)} u_i^{(5)})}{(\sum_i \sigma_i^{(O)} u_i^{(5)})^2}. \tag{5.12}$$

Layer 3: As in layer 4, only the error signal, $\delta_i^{(3)}$, needs to be computed as

$$\delta_i^{(3)} = \delta_i^{(4)}. \tag{5.13}$$

Layer 2: The adaptive rule of $m_{ij}^{(I)}$ is derived as

$$m_{ij}^{(I)}(n+1) = m_{ij}^{(I)}(n) + \eta \cdot \delta_i^{(2)} \cdot e^{f_i^{(2)}} \cdot \frac{2(u_i^{(2)} - m_{ij}^{(I)})}{\sigma_{ij}^{(I)2}}, \tag{5.14}$$

and the adaptive rule of $\sigma_{ij}^{(I)}$ becomes

$$\sigma_{ij}^{(I)}(n+1) = \sigma_{ij}^{(I)}(n) + \eta \cdot \delta_i^{(2)} \cdot e^{f_i^{(2)}} \cdot \frac{2(u_i^{(2)} - m_{ij}^{(I)2})}{\sigma_{ij}^{(I)3}}. \tag{5.15}$$

where $\delta_i^{(2)} = -\sum_k q_k$; and $q_k = -\delta_k^{(3)}$ if $a_i^{(2)}$ is minimum in kth rule node's inputs, $q_k = 0$ otherwise.

5.5 Simulation Results and Discussions

We verify the effectiveness of the intelligent leaky bucket algorithms for TS-UPC by comparing to the conventional leaky bucket algorithm. In the simulations, the primitive connection model as shown in Fig. 5.1 is adopted as the system model while a 2-state Markov modulated deterministic process (MMDP), a 2-state Markov modulated Bernoulli process (MMBP), and a VBR MPEG video "Star Wars" are employed as three different source models for the verification. We set the 2-state MMDP and the 2-state MMBP sources to have the mean active duration of 350 msec, the mean silence duration of 650 msec, and the mean cell rate $\Lambda_{mean} = 21.875$ cells/sec. The holding time of each state of the two source models follows a geometric distribution. During the active state, the 2-state MMDP source is a deterministic process which transmits cells at a fixed packetization interval of $T_{PCR} = 16$ msec, whereas the 2-state MMBP source is a Bernoulli process which, for every fixed time interval $T_{PCR} = 1.6$ msec, is likely to transmit a cell with probability of 0.1. The peak cell

rate of the VBR MPEG video is 4000 cells/sec, and the mean cell rate $\Lambda_{mean} = 975$ cells/sec. The window size for calculating the short-term mean rate is set to be ten times the sum of the mean active duration and mean silent duration, i.e., window size $= 10 * (350 + 650)$ msec $= 10$ sec.

The primitive connection model with TS-UPC is shown in Fig. 5.1.

In the simulations, C is set to be 1.1, thus for MMDP and MMBP sources, $\Lambda_{SCR} = C * \Lambda_{mean} = 24.0625$ cells/sec. The increment T for the conventional leaky bucket algorithm, which is taken to be the inverse of the sustainable cell rate, equals 0.041558, and the threshold τ_{SCR} of the leaky bucket equals $\tau_{IBT} + \tau'_{SCR}$, where $\tau_{IBT} = \lceil (MBS - 1)(T_{SCR} - T_{PCR}) \rceil$ and $\tau'_{SCR} = T_{SCR}$ for MMDP and MMBP. In order to compare the performance under the MMDP and MMBP sources, τ_{SCR} for the MMBP source is set to be the same as the MMDP source. To calculate τ_{IBT} for the MMDP source, we need the maximum burst size of the source. We set the allowed MBS for the MMDP source to be ten times the mean number of cell arrivals during the active state, i.e., $MBS = 10 * (350/16) = 218.75$ cells. Then τ_{SCR} can be calculated as 5.607. For the VBR MPEG video source, τ_{SCR} is set to be 3.79, and $MBS = 4870$. For simplicity of simulation and not to distract our attention, the queue in TS is assumed to be of infinite capacity.

In the chapter, we define Source σ as the ratio of the actual mean cell rate to the sustainable cell rate of the traffic source. There are three regions for Source σ: *non-violation region*, *intermediate region*, and *violation region*. The non-violation region ranges from Source $\sigma = 0$ to Source $\sigma = 1/C$, where C is the magnifying factor. The user within this region is a legal user and is guaranteed a zero cell dropping (or tagging) probability imposed by UPC and a negligible queueing delay introduced by TS. The intermediate region is the region between Source $\sigma = 1/C$ and Source $\sigma = 1$. Any user within this region is also a legal user and can still have zero cell dropping probability, but it does not have a

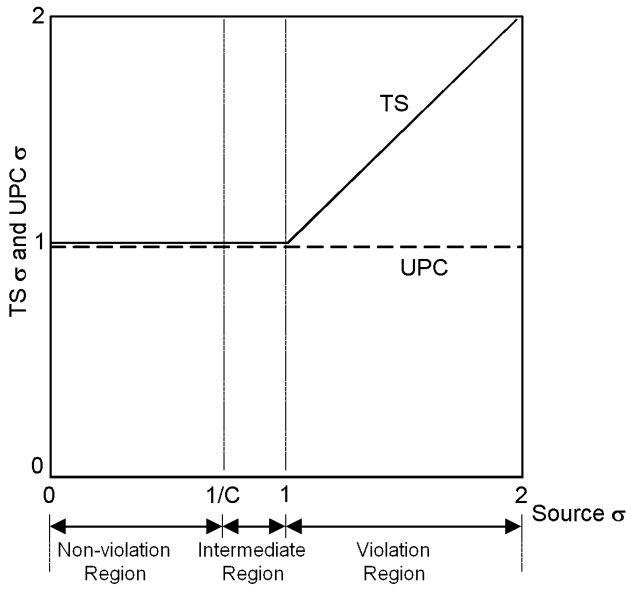


Figure 5.8: The correspondence between TS σ, UPC σ, and Source σ

satisfactory queueing delay. Finally, the violation region is from those beyond Source $\sigma = 1$. The user whose Source σ is located in this region is an illegal user, and both the cell dropping probability and queueing delay are not guaranteed by TS-UPC.

A connection with Source σ may have corresponding TS σ and UPC σ for TS and UPC, respectively, where TS σ (UPC σ) is defined as the ratio of the allowed mean cell rate to the sustainable cell rate at TS (UPC). As can be seen from Fig. 5.8, UPC σ is always held 1 for all Source σ's such that UPC can pass legal cells but drop (tag) illegal cells. TS σ is fixed at 1 for Source $\sigma \leq 1$, denoting that legal cells can pass TS transparently; TS σ = Source σ for Source $\sigma > 1$, denoting that the illegal calls can still pass TS in the sense that the badly-behaved user of the connection enlarges TS σ and intends to illegally enjoy a higher bit-rate service. If the user had not changed TS σ, then the cell stream passed by TS would have been conforming even though Source $\sigma > 1$, but there would be tremendous queueing delay incurred.

Fig. 5.9(a), Fig. 5.9(b), and Fig. 5.9(c) show the selectivity for the conventional leaky

bucket algorithm, the fuzzy leaky bucket algorithm, and the neural fuzzy leaky bucket algorithm, under the 2-state MMDP traffic source, the 2-state MMBP traffic source, and the MPEG video traffic source, respectively. The ideal curve of cell loss ratio is $P_d^* = 1 - 1/\text{Source } \sigma$ for Source $\sigma > 1$ and $P_d^* = 0$ for Source $\sigma \leq 1$. As it can be seen, the three algorithms present a zero cell loss ratio for Source $\sigma \leq 1$. But for Source $\sigma > 1$, the neural fuzzy leaky bucket algorithm has a cell loss ratio closest to the ideal curve, then the fuzzy leaky bucket algorithm and the conventional leaky bucket algorithm. And in the case of MPEG video traffic source, the difference phenomenon is more significant. It is because MPEG video traffic is burstier than MMDP and MMBP traffic sources and the intelligent TS-UPCs can be more adaptive than the conventional TS-UPC in dynamic, non-stationary systems.

Fig. 5.10(a), Fig. 5.10(b), and Fig. 5.10(c) show the responsiveness behavior of the three leaky bucket algorithms under the 2-state MMDP traffic source, the 2-state MMBP traffic source, and the MPEG video traffic source, for Source $\sigma = 1.5$. The responsiveness is illustrated in terms of the cell loss ratio versus time. From the figures, it can be seen that the intelligent leaky bucket algorithms not only have a shorter response time (i.e., the time it takes control action to start dropping the cells of a violating connection) which is about 1.5 sec., as compared to 4 sec. of the conventional leaky bucket algorithm, but also has a higher detection rate (i.e., the rate the cell loss ratio grows) than the conventional leaky bucket algorithm, under the MMDP, the MMBP, and the MPEG video traffic sources. It is because the adopted intelligent techniques have the ability to quickly express the control structure system using a priori knowledge; they are less dependent on the availability of a precise model of the controlled process and are more capable of handing non-linearities. Also, the neural fuzzy leaky bucket algorithm performs better than the fuzzy leaky bucket algorithm. It is because the neural fuzzy network is a neural network structured on the basis of fuzzy logics; it integrates intelligent learning and computation of neural networks

into fuzzy logic systems. Note that the fuzzy and neural fuzzy leaky bucket algorithms have similar detection rate to the fuzzy policer proposed in [46], but the former two have much earlier response time than the latter.

Fig. 5.11 shows the mean queueing delay versus different Source σ's under the 2-state MMDP, the 2-state MMBP, and the MPEG video traffic sources. We only consider the queueing delay of a connection with Source $\sigma \leq 1$ because the mean queueing delay of a violating connection needs not to be concerned. The figure reveals that the intelligent leaky bucket algorithms have the queueing delay more satisfactory than the conventional leaky bucket algorithm, regardless of the traffic source model used. This improvement owes to the fact that the intelligent leaky bucket algorithms further consider two system parameters, namely, the long-term and short-term mean rates. With these two parameters, the intelligent leaky bucket algorithms can know that the connection is conforming, so they set the increment to be very small in order to reduce the probability of cells being stored in the queue and thus decrease the mean queueing delay. Besides, the neural fuzzy leaky bucket algorithm has almost the same mean queueing delay as the fuzzy leaky bucket algorithm. Apparently, since the traffic source is legal, nearly nothing can be learned from the reinforcement learning by the neural fuzzy leaky bucket algorithm to improve its performance.

5.6 Concluding Remarks

In this chapter we employ intelligent techniques, which are the fuzzy logic controller and neural fuzzy networks, to design two intelligent usage parameter controllers for policing the sustainable-cell-rate of multimedia transmissions in ATM networks. The first algorithm we proposed is the fuzzy leaky bucket algorithm, which as the name implies, employs a fuzzy increment controller (FIC) in conjunction with the conventional leaky bucket algorithm. The FIC monitors the long-term mean rate and the short-term mean rate of a connection and

uses the fuzzification, inference rules and defuzzification to process them in order to derive the optimal increment value. The other intelligent leaky bucket algorithm we proposed is the neural fuzzy leaky bucket algorithm, which utilizes a neural fuzzy increment controller (NFIC) to dynamically adjust the increment value. The NFIC is basically an FIC except that it further employs a neural network to optimize its fuzzy logic system through the reinforcement learning.

Simulation results show that, regardless of the traffic sources chosen, both intelligent leaky bucket algorithms achieve better performances in terms of selectivity, responsiveness and mean queueing delay as compared to the conventional leaky bucket algorithm. The performance gain of the intelligent algorithms is a result of employing fuzzy logic and neural fuzzy controllers as well as taking the long-term and short-term mean rates as the feedback information. Based on the feedback information, both intelligent algorithms can adapt to the time-varying and non-stationary traffic, and thus enhance their performances.

Simulation results also show that the neural fuzzy leaky bucket algorithm achieves better performances than the fuzzy leaky bucket algorithm in all aspects especially the responsiveness. Despite the fuzzy logic is excellent in dealing with real-world impression and is capable of adapting itself to dynamic and bursty environments, it lacks the capability of automatically constructing its rule structure and membership functions to achieve the optimal performance. On the other hand, the neural fuzzy leaky bucket algorithm has perfected the impairment of the fuzzy leaky bucket algorithm by utilizing the learning capability of the neural network to continuously update the membership functions of the fuzzy logic system. However, the implementation cost of the neural fuzzy leaky bucket algorithm could be higher than that of the fuzzy leaky bucket algorithm.

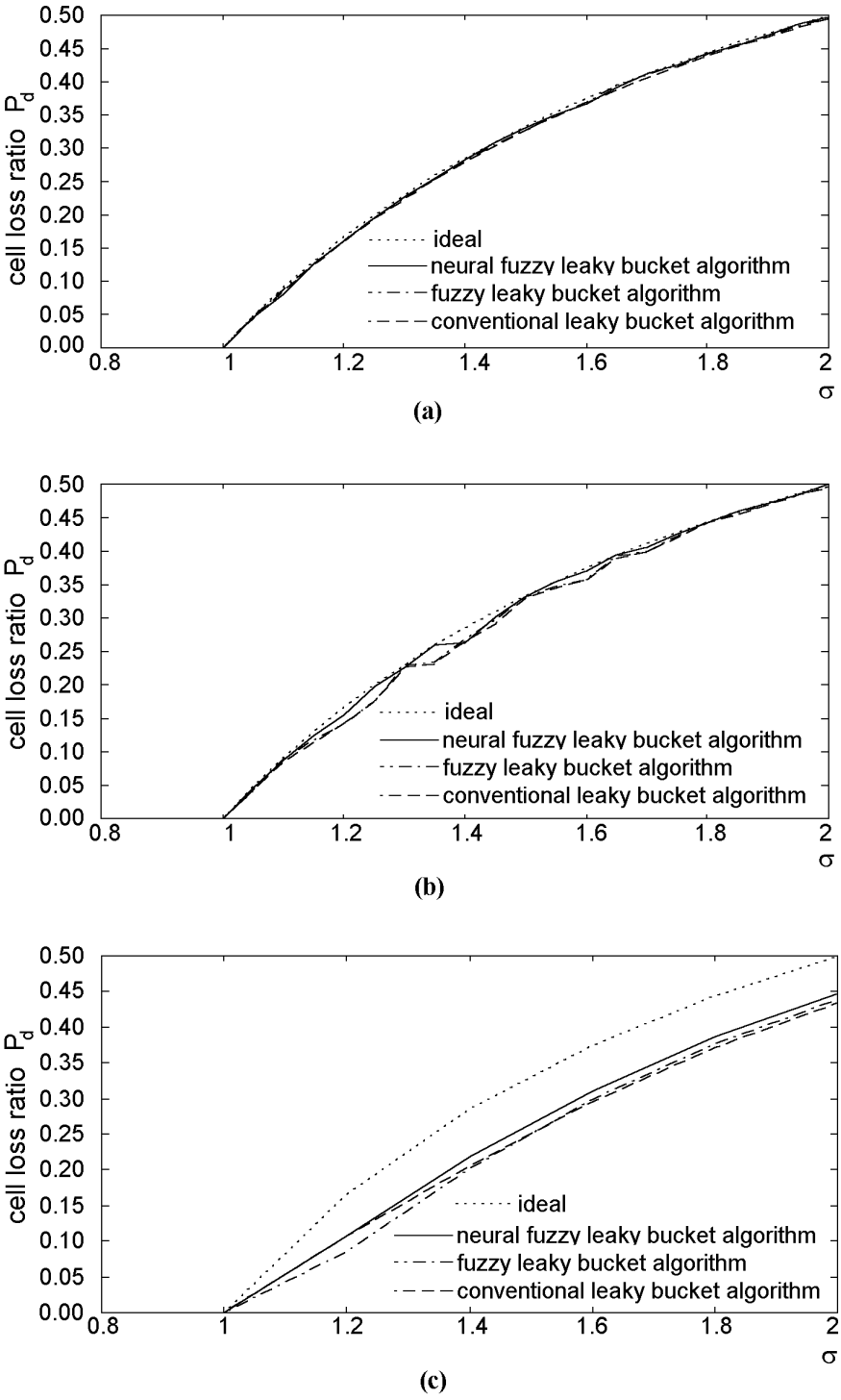


Figure 5.9: The selectivity of the conventional leaky bucket algorithm, the fuzzy leaky bucket algorithm, and the neural fuzzy leaky bucket algorithm under (a) MMDP traffic source (b) MMBP traffic source (c) MPEG video traffic source

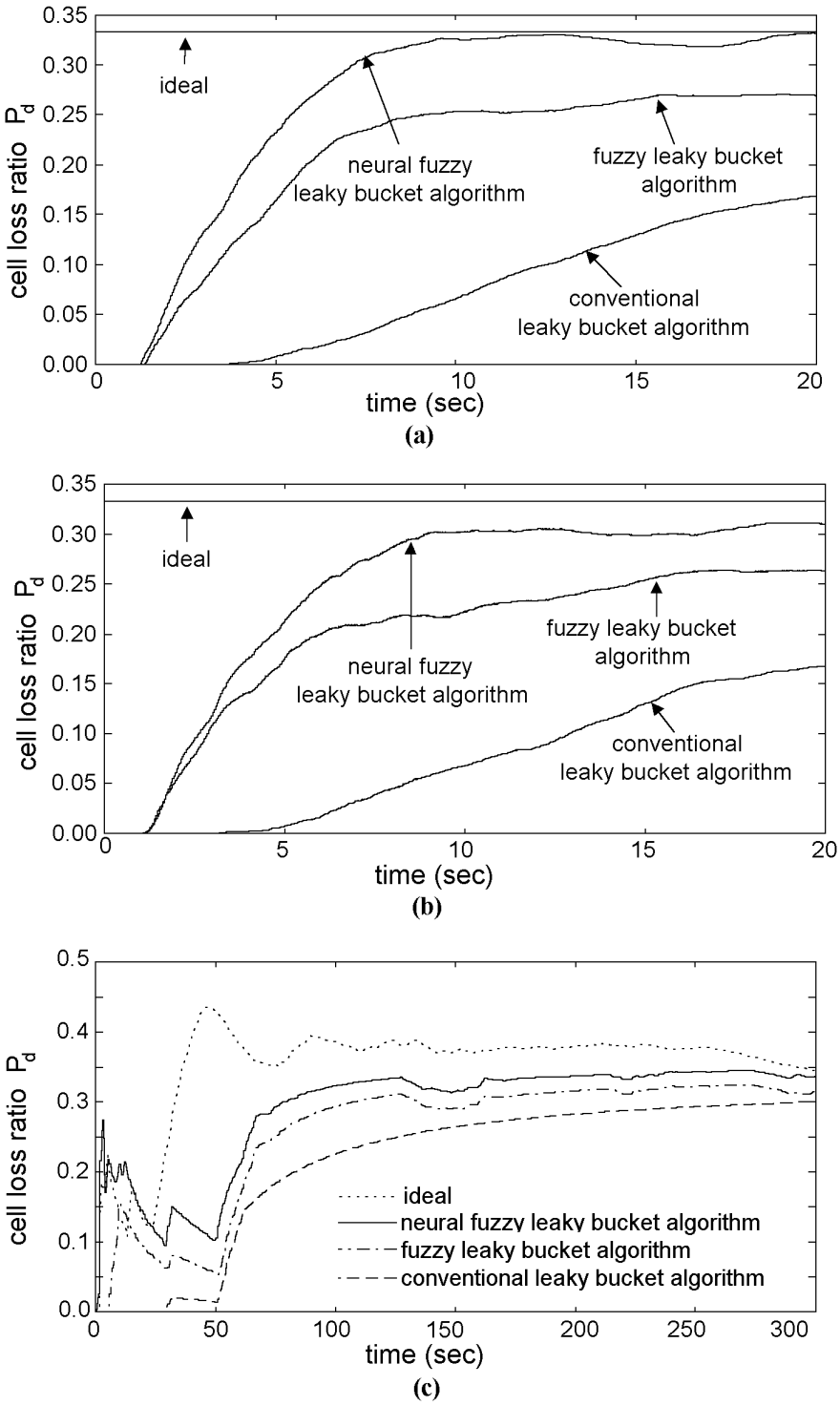


Figure 5.10: The responsiveness of the conventional leaky bucket algorithm, the fuzzy leaky bucket algorithm, and the neural fuzzy leaky bucket algorithm under (a) MMDP traffic source (b) MMBP traffic source (c) MPEG video traffic source, for Source σ=1.5

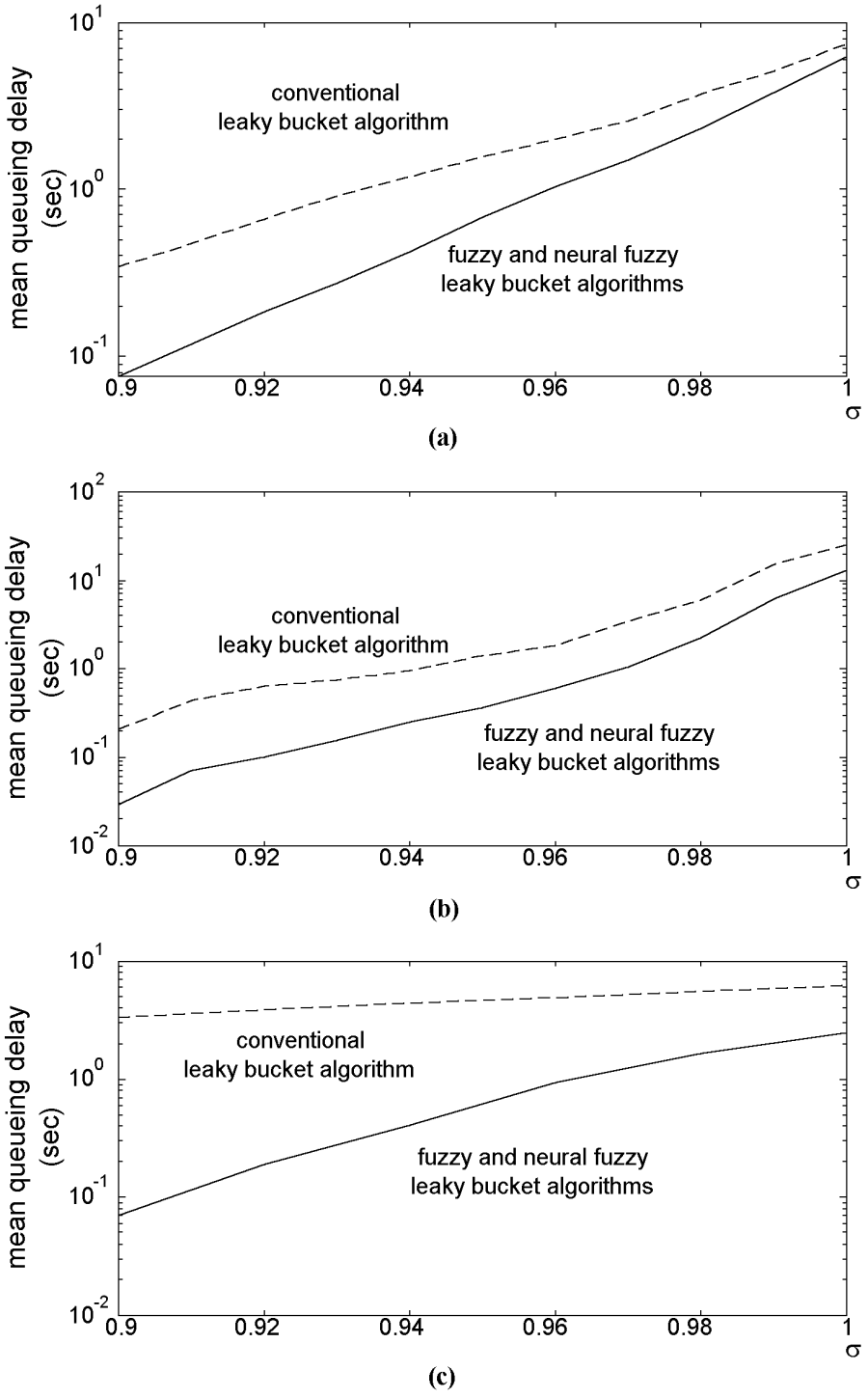


Figure 5.11: The mean queueing delay of the conventional leaky bucket algorithm, the fuzzy leaky bucket algorithm, and the neural fuzzy leaky bucket algorithm under (a) MMDP traffic source (b) MMBP traffic source (c) MPEG video traffic source

Chapter 6

An Enhanced Traffic Conditioner for Multimedia High-speed DiffServ IP Networks

As the UPC is the traffic policing function defined in ATM networks, the policing function for the IP networks is the traffic conditioning function handled by the traffic conditioner proposed in the differentiated services (DiffServ) model. In this chapter, an enhanced traffic marker (ETM) based on the Two-Rate-Three-Color-Marker (TRTCM) scheme is proposed for the traffic conditioner to perform traffic policing by properly determining the conforming level of the incoming packet and making a corresponding color notation on the packet. For the conventional traffic conditioner, the marking fairness among all connections within the aggregate traffic is not guaranteed because the processing is in the sense of an aggregate connection only. In addition, the end-to-end QoS of the applications would be degraded since the only native demotion processing would make the traffic rate of the high conforming level decline along the communication route when the traffics traverse across several network hops or domains [55]. The proposed ETM scheme introduces the features of aggressive promotion and fair share marking, and incorporates them into the existing traffic policing function. Simulation results show that the ETM scheme can fairly allocates the color notations among connections within an aggregate one and achieves a higher throughput for the traffic of each conforming level than the (conventional) TRTCM scheme does.

6.1 Introduction

There are increasing demands for the supporting of quality-of-service (QoS) over Internet. However, the IP network is basically originated on the "best-effort" service model and can hardly provides QoS guarantees for any connection because the bandwidth resources are allocated in a competition sense among all connections. Accordingly, Internet Engineering Task Force (IETF) has proposed two QoS-provisioning service models, named the Integrated Services (IntServ) [49] and the Differentiated Services (DiffServ) [50], respectively, to support Internet QoS. The IntServ model reserves the network resource before using it. It ensures the end-to-end QoS for each application (i.e. micro-flow) but has the scalability problem [51]. The DiffServ model focuses on the QoS of the aggregate connections and support only a set of finite number of predefined QoS classes in order to reduce the complexity and provide a promising solution to scalability. The connections that require a similar QoS level would be assigned to the same class, and thus (virtually) form an aggregate connection with a unique QoS processing including traffic conditioning. In a DiffServ network, an edge router is responsible for classifying the traffic into several aggregate connections associated with different QoS classes, conditioning the classified aggregate connections with respect to their traffic contract, and also processing the packet according to the QoS requirement defined for each class. The conditioning and processing functions are handled by a model named a traffic conditioner.

The traffic conditioner, consisting of a meter, a marker and a shaper (or a dropper), would continually determine the conforming level of the incoming traffic of an aggregate connection according to the measured traffic flow and its traffic contract [50], [51]. After that, a notation would be made on the traffic packets by the *marker* to indicate the conforming level, and a corresponding processing action such as dropping, shaping and bypassing is then taken upon the packets by the traffic shaper or dropper. The packet notation assigned by the

traffic marker in DiffServ networks is defined as three colors, denoted as green, yellow, and red, which are corresponding to three different pre-defined conforming levels for the packet with respect to the traffic contract. The packets assigned a green notation can be called as green packets for simplicity, and so do the packets marked a yellow or a red notation. The green packets stand for that these packets belong to the best conforming level and have the lowest dropping precedence (or the shortest shaping delay); the red packets, on the contrary, represents that these packets are judged to be with the worst conforming level (e.g. the violation level) and have the highest dropping precedence (or the longest shaping delay).

Several traffic conditioning schemes such as Single-Rate-Three-Color-Marker (SRTCM) [52], Two-Rate-Three-Color-Marker (TRTCM) [53] and Time-Sliding-Window-Three-Color-Marker (TSWTCM) [54] were proposed in RFC to implement the traffic conditioner. The TRTCM, which is popular because of its simplicity and effectiveness, adopts a couple of token buckets to police two rate properties of a traffic source simultaneously. The output traffic rate of green packets as well as the aggregate output rate of green and yellow packets are both ensured individually to conform to the traffic profile, where the green traffic rate is usually corresponding to the policed mean (or sustainable) rate of the incoming traffic and the aggregated green and yellow traffic rate represents the policed peak rate of the incoming traffic.

In addition to the *color-blind* operation mode, where the color marking decisions are based on only the metering results against the traffic contract, the alternative *color-aware* operation mode of the TRTCM performs the color marking according to not only the metering results against the traffic contract, but also the existing color notation of the packets, simultaneously. The purpose and operation principle of the color-aware mode is to maintain the existing color notation of the policed packets as best as it can while still conforming to the traffic contract. This is because, as noted above, the color notation of the packets can

represent the conforming level and correspond to the pre-defined QoS-provisioning packet processing behaviors. A packet may originally have its first color notation assigned by the output shaping function at the source node according to not only the metered results but also the *importance of the packet's content*. By properly allocating color notations representing higher conforming level and better QoS-provisioning packet processing behaviors to the packets with application critical contents, the QoS of each application is then expected to be quite improved while the traffic contract remains assured, since the packets with important application data are supported and served with better QoS. For example, the I-frame in the MPEG video is more vital than the other two coding frames, the B-frame and P-frame, because it serves as the base frame to reconstruct a series of video frames. The packets containing I-frame data can be assigned with the color notation representing higher conforming level and better QoS-provisioning packet processing behaviors so that the quality of the replayed video at the destination can be improved. Accordingly, the TRTCM operating in the color-aware mode can support better QoS for the applications than the TRTCM running in the color-blind mode.

As the TRTCM is a scheme to implement the traffic policing function in DiffServ IP networks, the packet *demotion* capability that re-marks a packet with a color notation corresponding to a lower conforming level than its existing one is inevitable and natural. However, a packet that is demoted due to occasionally short-term congestions or a locally stricter traffic profile may not have the chance to restore its existing or even the original conforming level. It has also been observed that the output rate of green packets might be impaired by the excessive incoming yellow packets: many packets with existing green notation are thus demoted to be with red color directly because the token resources are excessively consumed by the incoming yellow packets with the rate exceeding the traffic profile. These facts would result in the end-to-end QoS degradations for the applications since more packets carrying

critical application data and originally denoted with a high conforming level maybe treated by worse packet processing behaviors due to the demotions. Also, the marking fairness among all connections within a (virtual) aggregate traffic is uncertain.

Similar performance objectives such as the selectivity, responsiveness and queueing delay introduced in the UPC of ATM networks can also be employed to verify the efficiency of the traffic conditioner. In addition, because the processing of the traffic conditioner is based on the aggregate connection, the marking *fairness* for resource share among all connections within the (virtual) aggregate one could be taken into consideration as another performance objective. On the other hand, the IP network would be a world-wide network constituted by several interworking network systems which are hosted by different network service providers (NSPs). The network management policies of different NSPs may be varied and thus the definitions of a specific DiffServ QoS class can be distinct. Therefore, the traffic profile and the associated QoS-provisioning processing of an aggregate connection corresponding to the same QoS class may change from network domains to domains. As noted above in the TRTCM scheme, the packets might be demoted due to a locally stricter traffic profile and thus the end-to-end QoS of the applications would be degraded since the only demotion processing would make the traffic rate corresponding to the high conforming level decline along the communication route when the traffic traverse across several network hops or domains [55]. Consequently, a traffic promotion function is also considered as an objective for the traffic conditioner to not only restore the conforming levels of the previously demoted packets, but also aggressively promote the packets to higher conforming levels, if possible, for better application QoSs, while the traffic contract is still be respected. The aggressive promotion processing can then be equivalently regarded as fully utilizing the network resources to drive the traffic of each conforming level to achieve as high rate as possible by packet promotions while conforming to the traffic contract.

A random early demotion and promotion (REDP) technique [55] was proposed to overcome the unfair-marking problem. It implements a packet promotion function in addition to the demotion nature of the RED-In/Out (RIO) [56] marking mechanism, and achieves marking fairness by appropriately allocating the demotion/promotion probabilities among packets during the packet demotion and promotion procedures. In order to fully utilize the network resources for better application QoSs and provide marking fairness among all connections within the (virtual) aggregate one for TRTCM, a TC_PFG marking scheme [57] was proposed. However, in TC_PFG, only the packets belonging to the yellow conforming level is allowed to be promoted and this limits its application. Moreover, TC_PFG has the problem of unjust-promotion that the previously demoted packets can not be guaranteed to be promoted first when the network resource condition is available to perform the packet promotion function.

In this chapter, we proposed an enhanced traffic marker (ETM) based on the Two-Rate-Three-Color-Marker (TRTCM) scheme for the traffic conditioner to perform traffic policing by properly determining the conforming level of the incoming packet and making a corresponding color notation on the packet. The proposed ETM scheme introduces the features of aggressive promotion and fair share marking, and incorporates them into the existing traffic policing function. Simulation results show that the ETM scheme can fairly allocates the color notations among connections (micro-flows) within an aggregate one and achieves a higher throughput for the traffic of each conforming level than the (conventional) TRTCM scheme does.

The chapter is organized as follows. In Section 6.2, the design of the proposed ETM scheme is introduced and well described. In Section 6.3, the performance measures about marking accuracy and fairness of the ETM and conventional TRTCM schemes are evaluated and compared. Finally, some concluding remarks are presented in Section 6.4.

6.2 Enhanced Traffic Marker

The ETM is based on TRTCM scheme. It adopts the concept of RED [55] and provides functions of *promotion*, *fairness-guarantee*, and *green-packet protection*. The promotion function remarks the low-conforming packets into high-conforming ones when there are excessive resources of the network, and this would improve the throughput of the aggregate flow. Based on the natural demotion capability and the proposed promotion function, the fairness-guarantee function further improves the fair share among the connections of an aggregate one by appropriately determining reasonable demotion/promotion probabilities for the green and yellow packets of each individual connection. The green-packet protection function allows the token number in the bucket to be in deficit for incoming green packets to protect them from being affected by excessive incoming yellow packets.

TRTCM is composed of two token buckets denoted as T_P and T_C. The Peak Information Rate (PIR), the Peak Burst Size (PBS), the Committed Information Rate (CIR), and the Committed Burst Size (CBS) are four parameters to be configured. The size and the token generation rate of T_P (T_C) are set to be PBS and PIR (CBS and CIR), respectively. Initially, both the token buckets T_P and T_C are set to be full. An incoming packet is marked as green if both T_P and T_C are not empty. A packet is marked as yellow if T_P is not empty and T_C is empty. If T_P is empty, the incoming packet is marked as red. After marking, the number of tokens consumed from T_P and T_C is depend on the size of the packet.

The functional block diagram of the proposed ETM is illustrated in Fig. 6.1. We adopt the same architecture and parameters used in TRTCM, but modify its marking algorithm. The ETM also consists of two token buckets, denoted as T_P and T_C, respectively, and a marking algorithm processor, the *fair traffic marker with aggressive promotion* (FTM_AP). The size and token generation rate of T_P (T_C) are also set to be PBS and PIR (CBS and CIR), respectively. The FTM_AP works with a record unit and a promotion/demotion

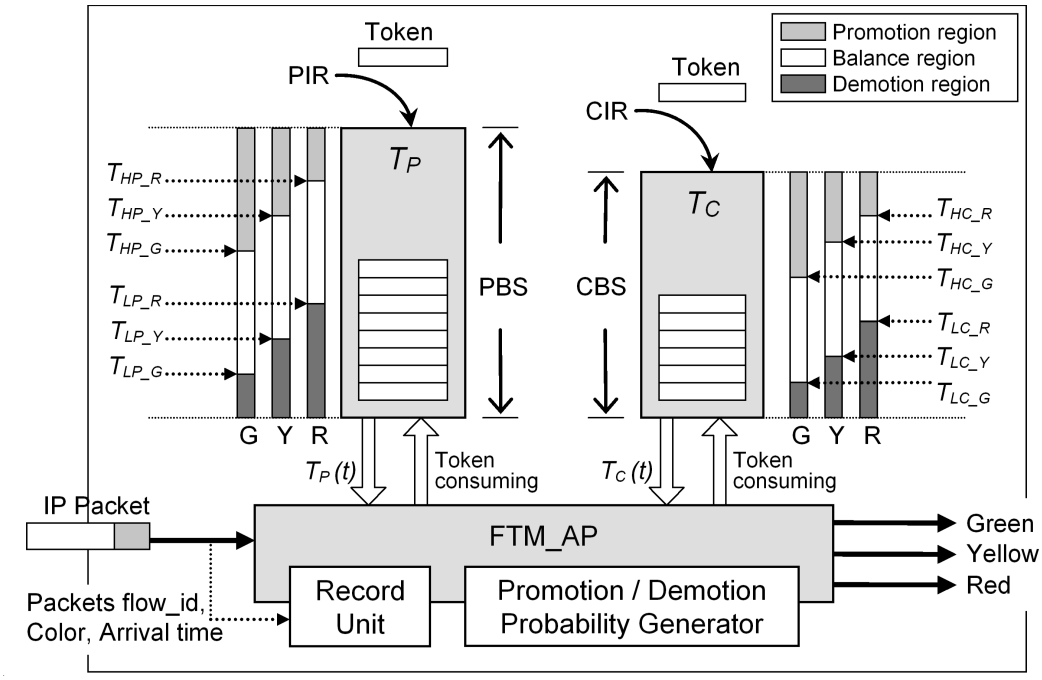


Figure 6.1: ETM scheme

probability generator. The *record unit* stores the flow-id, the existing color, and the arrival time for incoming packets every Δt. The number of green, yellow, and red packets of an individual connection j, denoted by $g(j)$, $y(j)$, and $r(j)$, respectively, are then recorded. The *promotion/demotion probability generator* uses the statistics to estimate the distribution of incoming packets and determines the promotion/demotion probability for each individual connection based on the available tokens.

FTM_AP supports the promotion of yellow and red packets to enhance the throughput as well as achieve better marking fairness. FTM_AP further the original color to reduce the unjust promotion. For each incoming packet, currently unused (CU) bits in the DS field [50] are used to store the information of its original color and current color. The original color is assigned at the source end and the current color could be remarked at any intermediate node. For simplicity, the G, Y, and R are used to denote the green, yellow, and red original colors, respectively. Y_G, for example, can be regarded as a compound color which represents

that the current color is yellow and the original color is green.

Twelve thresholds are defined and are further categorized into four groups, denoted by T_{HC}, T_{LC}, T_{HP}, and T_{LP}, respectively. T_{HC} and T_{LC} are used in T_C and they divide T_C into the promotion, balance, and demotion regions. Similarly, T_{HP} and T_{LP} divide T_P into three regions. Each threshold group defines three sub-thresholds for packets with different original color. For example, the thresholds T_{HC_G}, T_{HC_Y}, and T_{HC_R} defined in the group T_{HC} are specified for the original color of green, yellow and red packet, respectively. In order to mitigate the influence of unjust promotion, the constraint $T_{X\_G} \leq T_{X\_Y} \leq T_{X\_R}$, where $X \in \{HP, LP, HC, LC\}$, should be met. The constraint is to assure that a packet with a higher original conforming level color at the source would be demoted with a lower probability.

Assume $T_C(t)$ and $T_P(t)$ denote the number of tokens in T_C and T_P observed at time t, respectively. In our design, FTM_AP demotes an incoming packet from green to yellow when $T_C(t) < T_{LC}$. The demotion probability P_d^G is given by

$$P_d^G = Max_d^G \times \frac{T_{LC\_X} - T_C(t)}{T_{LC\_X}}, \tag{6.1}$$

where Max_d^G is the maximum demotion ratio defined by the system and $X \in \{G, Y, R\}$ is corresponding to the original color of the incoming packet. It can be found that a large T_{LC} will result in a higher demotion probability. The demotion probability is increased as the decrease of available tokens. We further use P_d^G and the packet number statistics $g(i)$ to estimate the actual demotion probability applied on the incoming green packet. At first, we can obtain the amount of green packets that can pass through ETM without demotion, denoted by g_{pass}, by

$$g_{pass} = \sum_{i=1}^{n} g(i) \times (1 - P_d^G), \tag{6.2}$$

where $g(i)$ is the amount of green packets of the individual connection i; n is the total number of the individual connection within an aggregate connection. According to the max-min fairness [58], we have to guarantee the sending rate of the "micro-flow that need less

(MFNL)" traffic first. The remaining resource is then equally shared by the "micro-flow that need more (MFNM)" traffic. It means that we shall not demote any green packet for each individual MFNL connection and then share the remaining resource of g_{pass} to the MFNM connections. Assume that there are n_{MFNM} MFNM connections and the remaining resource of g_{pass} is g_{MFNM}. Then we can recursively obtain the demotion probability of the green packet for individual connection j until the following condition is fulfilled:

$$P_d^G(j) = \begin{cases} 0 & \textit{if } j \textit{ belongs to MFNL traffic,} \\ 1 - \frac{(g_{MFNM}/n_{MFNM})}{g(j)} & \textit{if } j \textit{ belongs to MFNM traffic.} \end{cases} \tag{6.3}$$

That is, for an individual connection containing more green packets (i.e. larger $g(j)$), its green packets will have a higher probability to be demoted in ETM.

Similarly, a yellow packet is demoted to be red when $T_P(t) < T_{LP}$. The demotion probability P_d^Y is given by

$$P_d^Y = Max_d^Y \times \frac{T_{LP\_X} - T_p(t)}{T_{LP\_X}}, \tag{6.4}$$

where Max_d^Y is the maximum demotion ratio defined by the system and $X \in \{G, Y, R\}$ is corresponding to the original color of the incoming packet. The amount of yellow packets that can pass through ETM without demotion, denoted by y_{pass}, is then given by

$$y_{pass} = \sum_{i=1}^{n} y(i) \times (1 - P_d^Y). \tag{6.5}$$

And we can recursively obtain the demotion probability of the yellow packet for individual connection j as

$$P_d^Y(j) = \begin{cases} 0 & \textit{if } j \textit{ belongs to MFNL traffic,} \\ 1 - \frac{(y_{MFNM}/n_{MFNM})}{y(j)} & \textit{if } j \textit{ belongs to MFNM traffic.} \end{cases} \tag{6.6}$$

We will promote the yellow and red packets when there is available resource (i.e. sufficient token number in T_P and T_C). Based on the concept of max-min fairness, the individual connection i that consumes the smallest resource among the connections within the aggregate

one will be promoted first. In ETM, a packet is promoted by FTM_AP from yellow to green when $T_C(t) > T_{HC}$. The promotion probability P_p^Y is given by

$$P_p^Y = Max_p^Y \times \frac{T_C(t) - T_{HC\_X}}{CBS - T_{HC\_X}}, \tag{6.7}$$

where Max_p^Y is the maximum promotion ratio defined by the system and $X \in \{G, Y, R\}$ is corresponding to the original color of the incoming packet. It can be found that the promotion probability is raised along with the increase of the available tokens. The excess resource to be spent for supporting the promotion of yellow packets, denoted as y_{prom}, can be obtained by

$$y_{prom} = \sum_{i=1}^{n} y(i) \times P_p^Y. \tag{6.8}$$

The excess resource is then equally shared by the connections whose green traffic after accumulating the distributed resource does not violate their traffic profiles. We can recursively obtain the promotion probability of the yellow packet for the individual connection j, denoted as $P_p^Y(j)$, until the following condition is fulfilled:

$$P_p^Y(j) = \begin{cases} 0 & \textit{if } j \textit{ violates its traffic profile,} \\ \dfrac{\left(\left(y_{prom} + \sum_{i=1}^{k} g(i)\right) \Big/ k\right) - g(j)}{y(j)} & \textit{otherwise,} \end{cases} \tag{6.9}$$

where k is the total number of connections whose green traffic after accumulating the distributed resource still respect to their traffic profiles.

Similarly, a red packet can be promoted to be yellow when $T_P(t) > T_{HP}$. The promotion probability P_p^R is given by

$$P_p^R = Max_p^R \times \frac{T_P(t) - T_{HP\_X}}{CBS - T_{HP\_X}}, \tag{6.10}$$

where Max_p^R is the maximum promotion ratio defined by the system and $X \in \{G, Y, R\}$ is corresponding to the original color of the incoming packet. The excess bandwidth results from the promotion of red packets, denoted as

The excess resource to be spent for supporting the promotion of red packets, denoted as r_{prom}, is obtained by

$$r_{prom} = \sum_{i=1}^{n} r(i) \times P_p^R. \tag{6.11}$$

The promotion probability of red packets for the individual connection j, denoted as $P_p^R(j)$, is given by

$$P_p^R(j) = \begin{cases} 0 & \textit{if } j \textit{ violates its traffic profile,} \\ \dfrac{\left(\left(r_{prom} + \sum_{i=1}^{k} (g(i)+y(i))\right) \Big/ k\right) - (g(j)+y(j))}{g(j)+y(j)} & \textit{otherwise.} \end{cases} \tag{6.12}$$

In this equation, k is the total number of connections whose traffic profiles are still be respected after accumulating the distributed resource and $g(j) + y(j)$ is the resource volume that has been used by the individual connection j.

6.3 Simulation Results and Discussions

In this section, two simulation scenarios were presented to verify the marking *accuracy* and *fairness* of the ETM. The results were then compared with the (conventional) TRTCM. The network configuration for simulation is demonstrated in Fig. 6.2. N micro-flows belonging to the same service class originate from the sources and traverse across three DiffServ domains to reach their destinations (i.e. the "sink" node). The link capacity and delay parameter for each link are directly noted in the figure, and the round trip time (RTT) of a connection is assumed to be $36ms$.

6.3.1 Accuracy of the Marking

The first simulation scenario we take is to verify the marking accuracy of traffic markers. In this scenario, only single traffic source is necessary (i.e. $N = 1$ in Fig. 6.2.) but diverse traffic parameter conditions of the source would be considered to explore the marker's performance on accuracy. In this chapter, a Pareto traffic source with ten different traffic rate combination

Table 6.1: System parameters of scenario 1

Scheme	Parameters	Value
TRTCM	CBS	60 packets
	PBS	60 packets
ETM	T_{L_P}	17 packets
	Δt	0.432 ms
	CBS	60 packets
	PBS	60 packets
	$(Max_d^G, Max_d^Y, Max_P^Y, Max_P^R)$	(1, 1, 1, 1)
	$(T_{LC\_G}, T_{LC\_Y}, T_{LC\_R})$	(10, 17, 24) packets
	$(T_{LP\_G}, T_{LP\_Y}, T_{LP\_R})$	(10, 17, 24) packets
	$(T_{HC\_G}, T_{HC\_Y}, T_{HC\_R})$	(36, 43, 50) packets
	$(T_{HP\_G}, T_{HP\_Y}, T_{HP\_R})$	(36, 43, 50) packets

conditions is employed. The QoS profile specified at the ER1 is *CIR* equals to 5Mbps and *PIR* equals to 10Mbps and no profiles are specified at ER2 and ER3. That is, the maximum ideal green and yellow rates observed at the output of ER1 are 5Mbps and 5Mbps, respectively, and the ER2 and ER3 are transparent for the traffic. The other system parameters and the simulation results are listed in Table 6.1 and Table 6.2, respectively, and the results are observed and measured at the output of ER1.

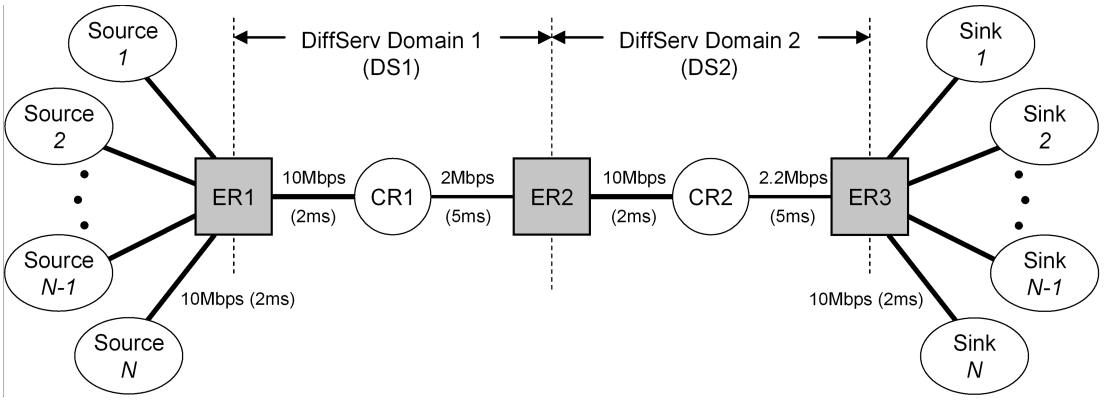


Figure 6.2: Simulation topology

In Table 6.2, it can be found that the ETM and TRTCM have similar results in traffic

conditions 1, 3, and 4. Traffic condition 2 demonstrates the case that a greedy source generates an excess amount of yellow packets than its QoS profile. In this case, the TRTCM does not take action for the excess yellow packets. Therefore, the yellow packets consume most of the tokens in T_P and result in the starvation of green packets. With ETM, it can be found that both the green and yellow packets are conformed to the QoS profile. Traffic conditions 5, 8, 9, and 10 simulate the congestion at source nodes such that the input green rate is smaller than the QoS profile. It's found that the ETM marks more green packets via aggressive promotion to meet the profile. In summary, the proposed ETM meets the traffic profile and achieves a highest throughput for the traffic of each conforming level than TRTCM does.

6.3.2 Fairness of the Marking

The second simulation scenario is to verify the fair share making capability of traffic markers. Therefore, several connections with different traffic characteristics and parameters are employed. The QoS profile in DiffServ domain 1 (DS1) is then configured as the bottleneck for the incoming green traffic. Thus, some of the green packets would be demoted or discarded. Besides, the Multiple-RED (MRED) [56] scheme is adopted in the core router (CR1 and CR2) to handle the congestion for both TRTCM and ETM. In MRED, packets marked as the lowest conforming level will be dropped first if congestion happens.

In the simulation, 45,000 packets (about 20 seconds) were simulated and the size of all packets is 512 bytes. Four UDP and two TCP connections are assumed. The UDP sources are implemented as Constant Bit Rate (CBR) traffic. The TCP sources are adaptive traffic with varied sending rates. The round trip time (RTT) of a TCP connection is assumed to be 36ms. The traffic parameters, the output (green, yellow, red) traffic rate in Mbps, for each source are as follows: UDP1=(1.9, 0.3, 0.3), UDP2=(1.0, 0.9, 0.9), UDP3=(0.7, 0.35, 0.35), UDP4=(0.3, 0.35, 0.35), TCP1=(1.0, 0.5, 0.5) and TCP2=(1.0, 0.5, 0.5). The QoS profile in

each edge router is the same and is set as $CIR = 2.0$ Mbps and $PIR = 2.5$ Mbps. The other system parameters including the parameters for both of the TRTCM and ETM schemes used in the simulation are the same with those defined in Table 6.1 for the simulation scenario 1.

In order to evaluate the marking fairness among all of the connections within the same aggregate one, we adopt the *fairness_index* defined by [59]

$$x_i = \frac{achieved\_rate_i}{ideal\_rate_i}, \tag{6.13}$$

$$fairness\_index = \frac{\left(\sum_i x_i\right)^2}{n \times \sum_i x_i^2}, \tag{6.14}$$

where $achieved\_rate_i$ and $ideal\_rate_i$ are the practical average throughput and the ideal throughput for the individual connection i, respectively; n is the number of active connections. The *fairness_index* falls into the range between 0 and 1. For the perfect fairness, the *fairness_index* should be equal to 1.

The average throughput of each conforming level for every traffic source obtained during the simulation is shown in Fig. 6.3. In Fig. 6.3, the *fairness_index* of TRTCM and ETM are 55.51% and 97.13%, respectively. It's because that TRTCM does not protect the TCP traffic and, thus, a large number of packets are demoted in ER1 and dropped in the core routers. This leads to the re-transmission mechanism of TCP and results in a low throughput for TCP users; therefore, the *fairness_index* is decreased. Here, we can also observe the unfair marking due to phase effect [60] in traditional marking algorithm among the adaptive and non-adaptive traffic sources: the CBR UDP traffic source, UDP3, suffers the lowest throughput for both of the green and yellow levels, since the packet demotion may happen in a periodic sense which is just corresponding to the constant packet rate of UDP3. The result shows that the ETM could more effectively mitigate the unfair marking caused by the phase effect than the TRTCM does. In the simulation, we also studied the effect of unjust promotion for TC_PFG and ETM. It is found that the ETM may also mitigate the unjust

promotion problem.

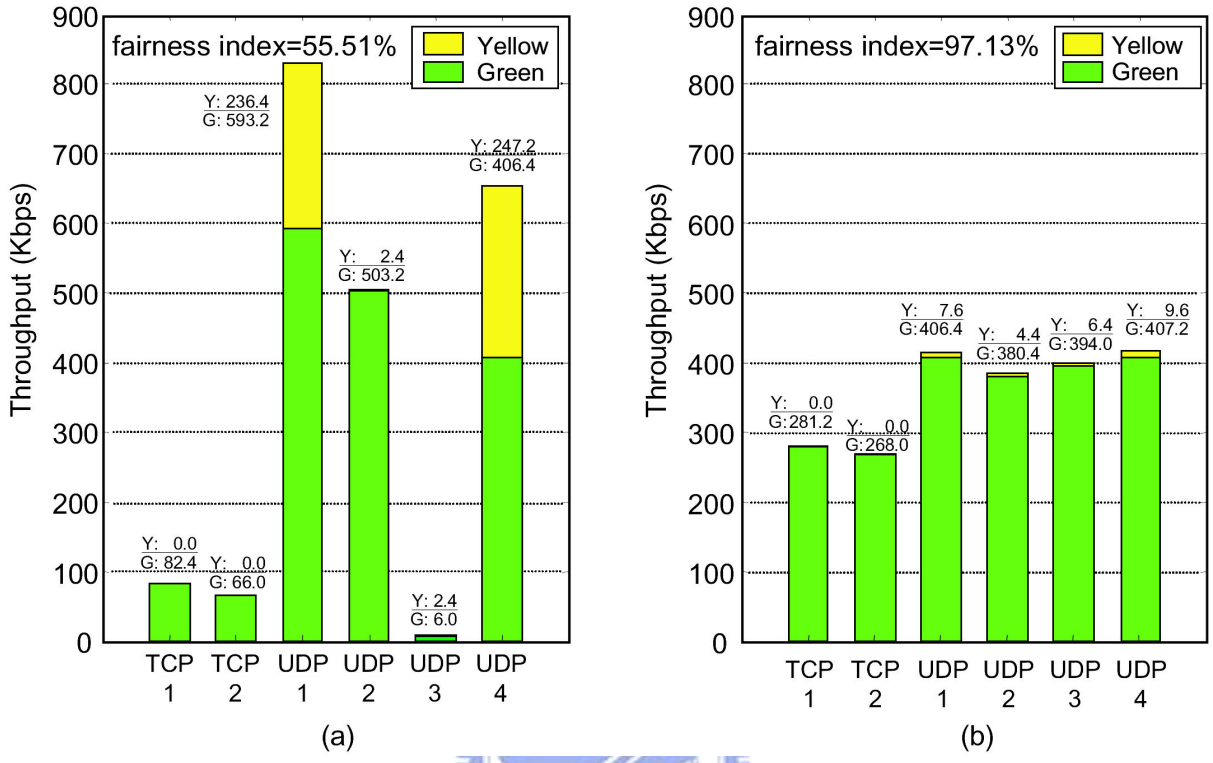


Figure 6.3: Throughput distribution of different traffic marker schemes in simulation scenario 2: (a) TRTCM scheme, (b) ETM scheme

6.4 Concluding Remarks

In this chapter, we proposed an enhanced traffic marker (ETM) for the TRTCM-based traffic conditioner to perform the traffic policing function in DiffServ IP networks. The primary feature of the proposed ETM is that it can fairly allocate the color notations among connections within an aggregate one. It also enhances the throughput of each conforming level for the aggregate connection to achieve as high rate as possible by not only restore the conforming levels of the previously demoted packets, but also aggressively promote the packets to higher conforming levels if the network resource condition is available, so that the

end-to-end QoS of the applications would be substantially improved while the traffic contract is still be respected. The operations of the ETM scheme as well as the computations of the promotion/demotion probabilities are carefully defined. The performances of the proposed ETM scheme were verified via simulations and the simulation results were compared with the conventional TRTCM scheme. Simulation results show the ETM scheme outperform the TRTCM scheme in both aspects of marking fairness and traffic throughput of each conforming level under congested and under-loaded networks.

Table 6.2: Simulation results of scenario 1

Scenario 1 Traffic Conditions		Input Rate (Mbps)	QoS Profile (Mbps)	Output rate of TRTCM (Mbps)	Output rate of ETM (Mbps)
1	Green	5.0	5.0	4.9145	4.9461
	Yellow	5.0	5.0	4.8082	4.7701
	Red	2.0	—	2.2773	2.2838
2	Green	5.0	5.0	4.5504	5.0033
	Yellow	8.0	5.0	5.1563	4.8674
	Red	2.0	—	5.2934	5.1293
3	Green	8.0	5.0	4.9371	4.9941
	Yellow	5.0	5.0	4.8152	4.8559
	Red	2.0	—	5.2477	5.1500
4	Green	10.0	5.0	4.9605	5.0010
	Yellow	12.0	5.0	4.9035	4.9313
	Red	2.0	—	14.1359	14.0678
5	Green	4.0	5.0	3.7645	4.8717
	Yellow	8.0	5.0	6.0348	4.9541
	Red	2.0	—	4.2008	4.1742
6	Green	3.0	5.0	2.9781	4.7209
	Yellow	6.0	5.0	5.8557	4.2170
	Red	2.0	—	2.1662	2.0621
7	Green	6.0	5.0	4.9578	4.9787
	Yellow	3.0	5.0	3.8930	4.0682
	Red	2.0	—	2.1492	1.9531
8	Green	3.0	5.0	3.0488	4.6262
	Yellow	2.0	5.0	2.0203	3.2920
	Red	6.0	—	5.9309	3.0818
9	Green	3.0	5.0	3.0447	4.6838
	Yellow	4.0	5.0	4.0328	3.1402
	Red	2.0	—	1.9195	1.1760
10	Green	2.0	5.0	1.9969	4.1533
	Yellow	2.0	5.0	1.9582	1.3479
	Red	2.0	—	2.0449	0.4988

Chapter 7

Conclusions and Future Works

In this dissertation, the traffic control functions involving the connection admission control and the traffic policing for multimedia high-speed networks are studied by employing the neural/fuzzy intelligent techniques or a sophisticated computation algorithm, including a neural fuzzy connection admission controller, a power-spectrum-based neural-net connection admission controller, a fuzzy increment controller, a neural fuzzy increment controller and a enhanced traffic marker. Both ATM and IP networks which can be utilized to construct the multimedia high-speed networks are considered in this dissertation. The CAC schemes which make the admission control decisions according to the *time-domain* and *frequency-domain* traffic parameters are both discussed where the intelligent techniques are chosen to implement the CAC controllers. Also, the enhanced algorithms which implement the traffic policing function by incorporating the intelligent techniques and a elaborate computation procedure into existing algorithms for ATM and IP networks respectively are both well explored.

In Chapter 3, a neural fuzzy connection admission control (NFCAC) scheme which based on the time-domain traffic parameters and provides QoS guarantees for ATM networks is proposed. The NFCAC scheme combines the linguistic control capability of a fuzzy logic controller and the learning ability of a neural network. This type of integrated neural fuzzy system can automatically construct a rule structure by learning from training examples

and can self-calibrate parameters of membership functions. It not only provides a robust framework to mimic experts' knowledge embodied in existing traffic control techniques but also constructs intelligent computational algorithms for traffic control. It can be easily trained and enhances system utilization. Simulation results show that the proposed NFCAC scheme provides system utilization about 32% and 11% higher than the EBCAC and FLCAC schemes proposed in [10] and [18], respectively, and the NFCAC scheme requires only a fraction of the 10^3 order and the 10^1 order of training cycles, consumed by the NNCAC scheme proposed in [23] and RBFCAC scheme, respectively. An NFCAC scheme such as the one introduced here may be the answer to the problem of designing a coherent call admission controller for ATM systems.

In Chapter 4, we propose a power-spectrum-based neural-net connection admission control (PNCAC) scheme for ATM networks. The PNCAC method adopts the converted power-spectrum parameters of traffic source to represent its traffic characteristics and uses neural network to implement the connection admission control. The frequency-domain power-spectrum parameters of traffic source possess additive property and can capture the correlation and burstiness behavior more than the time-domain parameters such as peak rate, mean rate, and peak rate duration. The neural network has the learning/adapting capabilities so that the boundary of the decision hyperplane for the connection admission control can be adjusted optimally and dynamically. Simulation results show that the proposed PNCAC enhances significantly the system utilization while fulfilling QoS requirements. Not only is it superior to the conventional equivalent capacity CAC scheme (ECCAC), it also obtains more flexibility and robustness than Hiramatsu's NNCAC.

However, the practical traffic characteristics of multimedia services in broadband networks may change very fast and abruptly with large volume. Also, several researches demonstrate that the multimedia traffic possesses self-similar or chaotic property, and present a

long-range dependence (LRD). The conventional traffic control algorithms based on current system performance measures may not perform well because of the fast varying dynamic traffic; the control decision would be obsolete and inappropriate due to delayed react to such a fast dynamic traffic. It is necessary to capture the next-step system performance, that is, the predicted information about the system status due to traffic change should be provided. Accordingly, a predictive intelligent traffic controller for broadband multimedia systems could be proposed as the future work for the CAC. It considers *predicted* system performance measures, instead of present ones, to well capture the oncoming effects in the future, besides also adopting the neural fuzzy network for the CAC decision making as well as the fuzzy logic controller for both of the equivalent capacity estimation of the new call and congestion estimation of the system. A pipelined recurrent neural network with extended recursive least square learning algorithm (PRNN/ERLS) [61], [62], which can efficiently reduce the prediction error for the statistical fluctuations of the system, could be employed to implement the predictors to well attain the advance information of the system. It is expected that the predictive intelligent traffic controller with predicted system measured statistics would achieves better performances than that of the conventional CAC schemes without prediction.

In Chapter 5, we employ two intelligent techniques, the fuzzy logic systems and the neural fuzzy networks, to design two intelligent leaky bucket algorithms, respectively, for sustainable-cell-rate usage parameter control of multimedia transmission in ATM networks. The first algorithm we proposed is the fuzzy leaky bucket algorithm, which as the name implies, employs a fuzzy increment controller (FIC) in conjunction with the conventional leaky bucket algorithm. The FIC monitors the long-term mean rate and the short-term mean rate of a connection and uses the fuzzification, inference rules and defuzzification to process them in order to derive the optimal increment value. The other intelligent leaky bucket

algorithm we proposed is the neural fuzzy leaky bucket algorithm, which utilizes a neural fuzzy increment controller (NFIC) to dynamically adjust the increment value. The NFIC is basically an FIC except that it further employs a neural network to optimize its fuzzy logic system through the reinforcement learning. Simulation results show that, regardless of the traffic sources chosen, both intelligent leaky bucket algorithms achieve better performances in terms of selectivity, responsiveness and mean queueing delay as compared to the conventional leaky bucket algorithm by responding about 160% faster when taking control actions against a non-conforming connection, while reducing as much as 50% of the queueing delay experienced by a conforming connection. The performance gain of the intelligent algorithms is a result of employing fuzzy logic and neural fuzzy controllers where the measured system statistics, the long-term and short-term mean rates, are introduced as the feedback information and served as the inputs of the intelligent controllers to form a robust and adaptive close-loop control system. Accordingly, both intelligent algorithms can adapt to the time-varying and non-stationary traffic, and thus enhance their performances. In addition, the simulation results also show that the neural fuzzy leaky bucket algorithm outperforms the fuzzy one by achieving better performance in all aspects especially the responsiveness.

In Chapter 6, we proposed an enhanced traffic marker (ETM) for the TRTCM-based traffic conditioner to perform the traffic policing function in DiffServ IP networks. The primary feature of the proposed ETM is that it can fairly allocate the color notations among connections within an aggregate one. It also enhances the throughput of each conforming level for the aggregate connection to achieve as high rate as possible by not only restore the conforming levels of the previously demoted packets, but also aggressively promote the packets to higher conforming levels if the network resource condition is available, so that the end-to-end QoS of the applications would be substantially improved while the traffic contract is still be respected. The operations of the ETM scheme as well as the computations of the

promotion/demotion probabilities are carefully defined. The performances of the proposed ETM scheme were verified via simulations and the simulation results were compared with the conventional TRTCM scheme. Simulation results show the ETM scheme outperform the TRTCM scheme in both aspects of marking fairness and traffic throughput of each conforming level under congested and under-loaded networks.

Bibliography

[1] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: a neuro-fuzzy synergism to intelligent systems*, Prentice–Hall, Singapore, 1996.

[2] J.-S. R. Jang, C. T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing: a computational approach to learning and machine intelligence*, Prentice–Hall, 1997.

[3] H.-J. Zimmermann, *Fuzzy set theory and its applications*, 2nd revised edition, Kluwer Academic Publishers, pp. 11–17, 1991.

[4] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller – Part I," *IEEE Tran. Systems, Man And Cybernetics*, Vol. 20, No. 2, pp. 419–435, Mar./Apr. 1990.

[5] N. E. Cotter, "The Stone-Weierstrass theorem and its application to neural networks," *IEEE Trans. Neural Networks*, Vol. 1, Iss. 4, pp. 290–295, Dec. 1990.

[6] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning internal representation by error propagation," *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MA: MIT Press, Cambridge, Vol. 1, Ch. 8, 1986.

[7] J. Moody and C. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computa.*, Vol. 1, No. 2, pp. 281-294, 1989.

[8] R. Guèrin, H. Ahmadi and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high–speed networks," *IEEE J. Select. Areas Commun.*, Vol. 9, No. 7, pp. 968–981, Sep. 1991.

[9] C. S. Chang and J. A. Thomas, "Effective bandwidth in high-speed digital networks," *IEEE J. Select. Areas Commun.*, Vol. 13, No. 6, pp. 1091–1100, Aug. 1995.

[10] G. Kesidis, J. Walrand and C. S. Chang, "Effective bandwidths for multiclass Markov fluids and other ATM sources," *IEEE/ACM Tran. Networking*, Vol. 1, No. 4, pp.424–428, Aug. 1993.

[11] A. I. Elwalid and D. Mitra, "Effective bandwidth of general Markovian traffic sources and admission control of high speed networks," *IEEE/ACM Trans. Networking*, Vol. 1, No. 3, pp. 329–343, June 1993.

[12] A. I. Elwalid, D. Heyman, T. V. Lakshman, D. Mitra and A. Weiss, "Fundamental bounds and approximations for ATM multiplexers with applications to video teleconferencing," *IEEE J. Select. Areas Commun.*, Vol. 13, No. 6, pp. 1004–1016, Aug. 1995.

[13] H. Saito, "Call admission control in an ATM network using upper bound of cell loss probability," *IEEE Trans. Commun.*, Vol. 40, No. 9, pp. 1512–1521, Sep. 1992.

[14] M. Murata, Y. Oie, T. Suda and H. Miyahara, "Analysis of a discrete-time single-server queue with bursty inputs for traffic control in ATM networks," *IEEE J. Select. Areas Commun.*, Vol. 8, No. 3, pp. 447–458, April 1990.

[15] S. Jamin, P. B. Danzig, S. J. Shenker and L. Zhang, "A framework for robust measurement-based admission control," *IEEE/ACM Trans. Networking*, Vol. 5, No. 1, pp. 56–70, Feb. 1997.

[16] M. Grossglauser and D. N. C. Tse, "A framework for robust measurement-based admission control," *IEEE/ACM Trans. Networking*, Vol. 7, No. 3, pp. 293–309, June 1999.

[17] A. R. Bonde and S. Ghosh, "A comparative study of fuzzy versus 'fixed' thresholds for robust queue management in cell-switching networks," *IEEE/ACM Trans. Networking*, Vol. 2, No. 4, pp. 337–344, Aug. 1994.

[18] R. G. Cheng and C. J. Chang, "Design of a fuzzy traffic controller for ATM networks," *IEEE/ACM Trans. Networking*, Vol. 4, No. 3, pp. 460–469, June 1996.

[19] I. W. Habib, A. A. Tarraf, and T. N. Saadawi, "Intelligent traffic control for ATM broadband networks," *IEEE Comm. Mag.,* Vol. 33, No. 10, pp. 76–85, Oct. 1995.

[20] A. Hiramatsu, "ATM communications network control by neural networks," *IEEE Trans. Neural Networks*, Vol. 1, No. 1, pp. 122–130, Mar. 1990.

[21] R. J. T. Morris and B. Samadi, "Neural network control of communications systems," *IEEE Trans. Neural Networks*, Vol. 5, No. 4, pp. 639–650, July 1994.

[22] S. A. Youssef, I. W. Habib and T. N. Saadawi, "A neurocomputing controller for bandwidth allocation in ATM networks," *IEEE J. Select. Areas Commun.*, Vol. 15, No. 2, pp. 191–199, Feb. 1997.

[23] R. G. Cheng and C. J. Chang, "Neural network connection admission control for ATM networks," *IEE Proceedings Communications*, Vol. 144, No. 2, pp. 93–98, Apr. 1997.

[24] B. Kosko, *Neural Networks and Fuzzy Systems*, Prentice–Hall, 1992.

[25] C. T. Lin, C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Computers*, Vol. 40, No. 12, pp. 1320–1336, Dec. 1991.

[26] S. T. Welstead, *Neural Network and Fuzzy Logic Applications in C++*, John Wiley & Sons Inc., 1994.

[27] Y. Lin and G. A. Cunningham III, "A new approach to fuzzy-neural system modeling," *IEEE Tran. Fuzzy Systems*, Vol. 3, No. 2, pp. 190–198, May 1995.

[28] N. Yin, S. Q. Li and T. E. Stern, "Congestion control for packet voice by selective packet discarding," *IEEE Tran. Commun.*, Vol. 38, No. 5, pp. 674–683, May 1990.

[29] D. E. Goldberg, *Genetic Algorithms*, Addison–Wesley, 1989.

[30] D. L. Gall, "MPEG: A video compression standard for multimedia applications," *Commun. of the ACM*, Vol. 34, No. 4, pp. 46–58, Apr. 1991.

[31] "Traffic control and congestion control in B-ISDN," *ITU-T Recommendation I.371,* Geneva, May 1996.

[32] "Traffic management specification: Version 4.0," *The ATM Forum Technical Committee,* Apr. 1996.

[33] S. Q. Li and C. L. Hwang, "Queue response to input correlation functions: continuous spectral analysis," *IEEE/ACM Trans. Networking*, Vol. 1, No. 6, pp. 678–692, Dec. 1993.

[34] H. D. Sheng and S. Q. Li, "Spectral analysis of packet loss rate at a statistical multiplexer for multimedia services," *IEEE/ACM Trans. Networking*, Vol. 2, No. 1, pp. 53–65, Feb. 1994.

[35] C. W. Therrien, *Discrete random signals and statistical signal processing*, Prentice–Hall, New Jersey, 1992.

[36] C. J. Chang, C. H. Lin, D. S. Guan and R. G. Cheng, "Design of a power-spectrum-based ATM connection admission controller for multimedia communications," *IEEE Trans. Industrial Electronics*, Vol. 45, No. 1, pp. 52–59, Feb. 1998.

[37] C. J. Chang, H. M. Chi, and R. G. Cheng, "A power-spectrum based connection admission control for ATM networks," *Proc. IEEE ICC '96*, Vol. 2, pp. 637–641, June 1996.

[38] C. J. Chang, S. Y. Lin, R. G. Cheng and Y. R. Shiue, "PSD-based neural-net connection admission control," *Proc. IEEE INFOCOM '97*, Vol. 3, pp. 955–962, April 1997.

[39] C. J. Chang, S. Y. Lin, Y. R. Shiue and R. G. Cheng, "A power-spectrum based neural fuzzy connection admission mechanism for ATM networks," *Proc. IEEE ICC '97*, Vol. 3, pp. 1709–1713, June 1997.

[40] E. P. Rathgeb, "Modeling and performance comparison of policing mechanism for ATM networks," *IEEE J. Select. Areas Commun.*, Vol. 9, No. 3, pp. 325–334, Apr. 1991.

[41] L. Dittmann, S. B. Jacobsen and K. Moth, "Flow enforcement algorithms for ATM networks," *IEEE J. Select. Areas Commun.*, Vol. 9, No. 3, pp. 343–350, Apr. 1991.

[42] S. Shioda and H. Saito, "Satisfying QoS standard with combined strategy for CAC and UPC," *Proc. ICC '95*, Vol. 2, pp. 965–969, June 1995.

[43] M. Butto, E. Cavallero and A. Tonietti, "Effectiveness of the leaky bucket policing mechanism in ATM networks," *IEEE J. Select. Areas Commun.*, Vol.9, No.3, pp. 335–342, Apr. 1991.

[44] T. D. Ndousse, "Fuzzy neural control of voice cells in ATM networks," *IEEE J. Select. Areas Commun.*, Vol. 12, No. 9, pp. 1488–1494, Dec. 1994.

[45] A. A. Tarraf, I. W. Habib and T. N. Saadawi, "A novel neural network traffic enforcement mechanism for ATM networks," *IEEE J. Select. Areas Commun.*, Vol. 12, No. 6, pp. 1088–1096, Aug. 1994.

[46] V. Catania, G. Ficili, S. Palazzo and D. Panno, "A comparative analysis of fuzzy versus conventional policing mechanisms for ATM networks," *IEEE/ACM Trans. Networking*, Vol. 4, No. 3, pp. 449-459, June 1996.

[47] R. G. Garroppo, S. Giordano, S. Miduri and F. Russo, "A prediction based UPC mechanism for VBR video traffic," *Proc. ICC '98*, Vol. 2, pp. 1119–1123, June 1998.

[48] F. Guillemin, C. Rosenberg and J. Mignault, "On characterizing an ATM source via the sustainable cell rate traffic descriptor," *Proc. INFOCOM '95*, Vol. 3, pp. 1129–1136, April 1995.

[49] R. Braden, D. Clark and S. Shenker, "Integrated services in the Internet architecture: an overview," *RFC 1633*, June 1994.

[50] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An architecture of differentiated services," *RFC 2475*, Dec. 1998.

[51] X. Xiao and L. M. Ni, "Internet QoS: a big picture," *IEEE Magazine on Network*, Vol. 13, No. 2, pp. 8–18 Mar./Apr. 1999.

[52] J. Heinanen and R. Guerin, "A single rate three color marker," *RFC 2697*, Sep. 1999.

[53] J. Heinanen and R. Guerin, "A two rate three color marker," *RFC 2698*, Sep. 1999.

[54] W. Fang, N. Seddigh and B. Nandy, "A time sliding window three color marker," *RFC 2859*, June 2000.

[55] F. Wang and P. Mohapatra, "A random early demotion and promotion marker for assure service," *IEEE Journal of Selected Areas in Communications*, Vol. 18, No. 12. pp. 2640–2650, Dec. 2000.

[56] D. D. Clark and Wenjia Fang, "Explicit allocation of best-effort packet delivery service," *IEEE/ACM Transactions on Networking*, Vol. 6, No. 4, pp. 362–373, August 1998.

[57] C. J. Chang, Y. H. Cheng and L. F. Lin, "The traffic conditioner with promotion and fairness guarantee schemes for DiffServ networks," *Proc. of ICC 2003*, Vol. 1, pp. 238–242, May 2003.

[58] J. M. Jaffe, "Bottleneck flow control," *IEEE Transactions on Communications*, Vol. 29, No. 7, pp. 954–962, July 1981.

[59] R. Jain, *The art of computer systems performance analysis*, John Wiley & Sons Inc., 1991.

[60] S. Floyd and V. Jacobson, "On traffic phase effects in packet-switched gateways," *Internetworking: Research and Experience*, Vol. 3, No. 3, pp. 115–156, Sep. 1992.

[61] S. Haykin and L. Li, "Nonlinear adaptive prediction of nonstationary signals," *IEEE Trans. Signal Processing*, Vol. 43, No. 2, pp. 526–535, Feb. 1995.

[62] J. Baltersee and J. A. Chambers, "Nonlinear adaptive prediction of speech signals using a pipelined recurrent neural network," *IEEE Trans. Signal Processing*, Vol. 46, No. 8, pp. 2207–2216, Aug. 1998.

Vita

Li-Fong Lin was born in Tainan, Taiwan, R.O.C., on June 10, 1974. He received B.E. degree in communication engineering from the Department of Communication Engineering, National Chiao-Tung University, Taiwan, in 1996. Currently, he is a candidate and working toward the Ph.D degree in the area of communication engineering, in the Institute of Communication Engineering, National Chiao-Tung University. His current research interests include performance analysis, traffic control over multimedia high-speed networks, and intelligent techniques involving fuzzy logic, neural networks and neural fuzzy systems. The principle considerations of his Ph.D dissertation are the analysis, simulation and optimal design of traffic control schemes in multimedia high-speed networks.

博士候選人資料

◎ 姓　名：林立峰

◎ 性　別：男

◎ 出生日期：民國 63 年 06 月 10 日

◎ 出生地：台灣省台南市

◎ 住　址：高雄市三民區本安里 11 鄰汾陽路 113 號

◎ 學　歷：

1. 國立交通大學電信工程學系

   時間：81 年 9 月 ～ 85 年 8 月，畢業

2. 國立交通大學電信工程研究所碩士班

   時間：85 年 9 月 ～ 86 年 8 月，直升博士班

3. 國立交通大學電信工程研究所博士班

   時間：86 年 9 月 ～ 迄今

◎ 論文題目：

英文：QoS-provisioning Traffic Control Schemes for Multimedia High-speed Networks Using Intelligent Techniques

中文：多媒體高速網路之服務品質保證式智慧型訊務控制機制

Publication List of Li-Fong Lin

◎ Journal Paper（期刊論文）

[1] C. J. Chang, Z. Eul, C. S. Chang, and L. F. Lin, "**Intelligent Leaky Bucket Algorithms for Sustainable-Cell-Rate Usage Parameter Control in ATM Networks**," *IEEE Transactions on Multimedia*, Vol. 6, No. 5, pp. 749-759, Oct. 2004.

[2] C. J. Chang, L. F. Lin, S. Y. Lin and R. G. Cheng, "**Power-spectrum-based neural-net connection admission control for multimedia networks**," *IEE Proceedings – Commun.*, Vol. 149, No. 2, pp. 70-76, April 2002.

[3] R. G. Cheng, C. J. Chang, and L. F. Lin, "**A QoS-Provisioning Neural Fuzzy Connection Admission Control for Multimedia High-Speed Networks**," *IEEE/ACM Transactions on Networking*, Vol. 7, No. 1, pp. 111-121, Feb. 1999.

◎ Conference Paper（會議論文）

[1] L. F. Lin, N. Y. Yan, C. J. Chang, and R. G. Cheng, "**An Enhanced Traffic Marker for DiffServ Networks**," *Proc. of ICOIN 2005* (also *LNCS 3391: Information Networking – Convergence in Broadband and Mobile Networking,* edited by Cheeha Kim, Publisher: Springer-Verlag, 2005), Jeju, Korea, Jan. 31 – Feb. 2, 2005, pp. 432-442. **(Received Best Paper Award)**

[2] C. J. Chang, Y. H. Cheng, and L. F. Lin, "**The Traffic Conditioner with Promotion and Fairness Guarantee Schemes for DiffServ Networks**," *Proc. of ICC 2003*, Anchorage, Alaska, USA, May 11-15, 2003, Vol. 1, pp. 238-242.

[3] S. Y. Chen, L. F. Lin, C. S. Chang, and C. J. Chang, "**The sustainable-cell-rate usage parameter control with adjustable window for high-speed multimedia communications**," *Proc. of ACM SAC '2001*, Las Vegas, Nevada, USA, pp. 467-471, March 11 – March 14, 2001.

[4] C. J. Chang, C. S. Chang, Z. Eul, and L. F. Lin, "**Intelligent Leaky Bucket Algorithms for Sustainable-Cell-Rate Usage Parameter Control in ATM Networks**," *Proc. of ICOIN-15*, Beppu City, Oita, Japan, pp. 453-460, Jan. 31 – Feb. 2, 2001.

[5] C. J. Chang, W. C. Cheng, L. F. Lin, and R. G. Cheng, "**Intelligent Traffic Controller with Multiple QoSs Provisioning for High-Speed Multimedia Networks**," *Proc. of ISCOM '99*, Kaohsiung, Taiwan, Nov. 7 – Nov. 10, 1999.

[6] L. F. Lin, Z. S. Eul, R. G. Cheng, and C. J. Chang, "**Implementation of an Admission Controller for High-Speed Multimedia Networks**," *Proc. of IEEE ICCE '98*, Los Anglos, CA, pp. 254-255, Jun. 2 – Jun. 4, 1998.

[7] C. J. Chang, Y. R. Shiue, R. G. Cheng, and L. F. Lin, "**A PSD-based Neural-net Connection Admission Control Scheme for ATM Systems**," *Proc. of SCCT '97*, Chung-Li, Taiwan, pp. 278-287, 1997.