An Adaptive Connection Admission Control under Generalized Processor Sharing Discipline

by
Chang He

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Masters of Applied Science

at the
Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Systems and Computer Engineering
Carleton University

Ottawa, Ontario

April 7, 2004

©Chang He, 2004

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-93938-3

Our file Notre référence

ISBN: 0-612-93938-3

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

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

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

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

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Acknowledgements

This thesis could not have been done without the help from many people. I consider myself lucky to have these people in my life.

First of all, I would like to thank my supervisor Professor Jerome Talim for his valuable guidance, encouragement, trust and patience all along my thesis study. I would not be here without him. I owe a great debt of gratitude to him.

I also thank Professor Ioannis Lambadaris as my advisor in my first year study and the people in Dept. of System and Computer Engineering for helping me to finish my study.

Many thanks to my buddies Jeni, Mark, Wei, Houjin and lots of others. The friendship, encouragement and advice gave me tremendous help.

Special thanks are also given to my landlady Katherine Kimbell, and my friend Paul Simon, not only for revising my writing, but also for the family love and care they gave to me. I owe them too much.

Finally, to my father and mother, I do not know what I can contribute to them. Wherever I go, they are always behind me. Thank you mom and dad.

Abstract

In this thesis, we propose a novel self-adaptive fuzzy connection admission control (FCAC) scheme for multi-service packet switched networks. Currently, the full statistical characterization for online traffic is very difficult to get. However, many connection admission control (CAC) schemes are based on certain traffic models which limit their implementation into practice. This thesis proposes a measurement based CAC scheme without any assumption of source characterization. It defines a source characteristic factor which is derived from the peak rate, mean rate and QoS constraint to estimate the traffic behavior in terms of the possible loss ratio and bandwidth utilization. The estimation and measurement from actual traffic are taken as the input variables to a fuzzy logic knowledge base control system in order to make the admitted control. This scheme also has the property of classifying admission connections to provide differentiated services. We evaluate our scheme with two algorithms: Mosquito algorithm and peak rate allocation CAC scheme. The comparison among these CACs shows our scheme has an improved performance especially in terms of bandwidth utilization. The FCAC also has less computational power than the effective bandwidth method.

Contents

List of Symbols	ix
List of Acronyms	xi
1 Introduction	1
1.1 Multi-service Network	1
1.2 Quality of Service	4
1.3 Traffic Management	6
1.3.1 Resource Sharing	7
1.3.2 Connection Admission Control (CAC)	9
1.3.3 Traffic Characterization	10
1.4 Problem Statement	11
1.5 Thesis Contribution	12
1.6 Organization	12
2 Literature Review	14
2.1 GPS - Ideal Scheduling Discipline	14
2.2 CAC Schemes Review	17

2.2.1	Effective Bandwidth	18
2.2.2	Artificial Intelligence (AI) Method	21
2.3	Fuzzy Logic	22
2.3.1	Crisp Set and Fuzzy Set	22
2.3.2	Membership Function	24
2.3.3	Operations on Fuzzy Sets	25
2.3.4	Rules	26
2.3.5	Fuzzy Knowledge Based Control	27
3	A CAC Scheme Using Fuzzy Logic	31
3.1	Notation definition - S , Q and μ	31
3.2	Framework	33
3.3	Queuing and Scheduling Module	37
3.4	Q and μ Estimator	40
3.4.1	Relations between S , Q and μ	40
3.4.2	Q and μ Estimation	41
3.4.3	Linear Regression	45
3.5	Real-time Measurement Module	47
3.6	Fuzzy Connection Admission Control Module	48
3.6.1	Membership Functions	51
3.6.2	Fuzzy Rule Base	54
3.6.3	Rule Firing and Defuzzification	56
4	Self Learning and Self Adaptation	61
4.1	Self Learning in Q and μ Estimation	61

4.2	Self Adaption in fuzzy control module	63
4.2.1	Scenario 1 -Evaluation of the Effect of the Lower bound and Upper Bound of Q_e and Q_a	67
4.2.2	Scenario 2-Evaluation of the Effect of the Lower bound and Upper Bound of μ_e and μ_a	72
4.2.3	Scenario 3 - Evaluation of the Effect of the “Admission” and “Re- jection” bounds	75
4.2.4	Observations and Conclusion	77
4.2.5	Self Adaption	78
5	Simulation and Comparison	79
5.1	General Simulation Environment	79
5.2	Estimation of Q and μ	81
5.3	FCAC Evaluation	84
5.4	Comparison	90
6	Conclusion	95
6.1	Summary of the Work	95
6.2	Future Work	99
	Bibliography	100

4.2	Self adaptation in fuzzy control module	63
4.2.1	Scenario 1 -Evaluation of the Effect of the Lower bound and Upper Bound of Q_e and Q_a	67
4.2.2	Scenario 2-Evaluation of the Effect of the Lower bound and Upper Bound of μ_e and μ_a	72
4.2.3	Scenario 3 - Evaluation of the Effect of the “Admission” and “Re- jection” bounds	75
4.2.4	Observations and Conclusion	77
4.2.5	Self Adaptation	78
5	Simulation and Comparison	79
5.1	General Simulation Environment	79
5.2	Estimation of Q and μ	81
5.3	FCAC Evaluation	84
5.4	Comparison	89
6	Conclusion	96
6.1	Summary of the Work	96
6.2	Future Work	100
	Bibliography	101

4.1	Adaptive fuzzy CAC framework	62
4.2	Self-learning cycle	63
4.3	Change of parameters in Scenario 1 - Case a	68
4.4	Scenario 1 - Case a: shifting the lower bound of Q_e and Q_a	69
4.5	Change of parameters in Scenario 1 - Case b	69
4.6	Scenario 1 - Case b: shifting the upper bound of Q_e and Q_a	70
4.7	Change of parameters in Scenario 1 - Case c	71
4.8	Scenario 1 - Case c: shifting the lower and upper bound of Q_e and Q_a Simultaneously	71
4.9	Scenario 2- Case a: shifting the lower bounds of μ_e and μ_a	73
4.10	Scenario 2- Case b: shifting the upper bound of μ_e and μ_a	73
4.11	Scenario 2 - Case c: shifting the lower and upper bound of μ_e and μ_a simultaneously	74
4.12	Scenario 3- Case a: shifting the “rejection” bound	76
4.13	Scenario 3 - Case b: shifting the “admission” bound	76
5.1	Estimation of Q and μ	82
5.2	Linear regression of $Q \cdot S$ empirical curve	83
5.3	Linear regression of $\frac{\mu}{S}$ empirical curve	84
5.4	Fuzzy decision	86
5.5	Customer satisfaction rate Q	87
5.6	Unused bandwidth μ	88
5.7	Distribution of unused bandwidth	89
5.8	Number of queues in progress	90
5.9	Number of connection in progress	91

5.10 Multiplexing gain	92
5.11 Bandwidth utilization comparison	93

List of Tables

2.1	True value table of “old”	23
3.1	Sample sources	59
3.2	Fuzzy rule base	60
4.1	Parameters assignment in Scenario 1	67
4.2	Parameters assignment in Scenario 2	72
4.3	Parameters assignment in Scenario 3	75
5.1	Admission rate and blocking rate for FCAC	89
5.2	Connection request blocking rate	94
5.3	Performance comparison	94

List of Symbols

Q	Customer satisfaction rate
S	Source characteristic factor
μ	Unused bandwidth
p_i	Peak rate of connection i
m_i	Mean rate of connection i
l_i	QoS constraint for bit loss ratio of connection i
a_i	Current arrival of connection i
x_i	Current queue length of connection i
b_i	Buffer size of connection i
α_i	Allocated bandwidth for connection i
$\phi_i(t)$	Weight in bandwidth allocation for connection i at time t
$r_i(t)$	Excess bandwidth during at time t in connection i
C	System bandwidth

Q_0	Customer satisfaction rate of class 0
N_0	Total connection number of class 0
Q_1	Customer satisfaction rate of class 1
N_1	Total connection number of class 1
n	Average number of concurrent connections
Q_e	Estimated customer satisfaction rate
Q_a	measured customer satisfaction rate
μ_e	Estimated unused bandwidth
μ_a	Measured unused bandwidth
$\chi(a)$	Membership function of a

List of Acronyms

AAA	Authorization, Authentication, and Accounting
AI	Artificial Intelligence
ARPA	Advanced Research Projects Agency
ATM	Asynchronous Transfer Mode
BISDN	Broadband Integrated Service Digital Network
CAC	Connection Admission Control
CCITT	Comité Consultatif Internationale de Télégraphique et Téléphonique
DiffServ	Differentiated Service
DSCP	Differentiated Services Code Point
EB	Effective Bandwidth
GPS	Generalized Processor Sharing
FCAC	Fuzzy Connection Admission Control
FKBC	Fuzzy Knowledge Based Control

IntServ	Integrated Service
ITU-T	International Telecommunications Union - Telecommunication
IPP	Interrupted Poisson Process
MMPP	Markov Modulated Poisson Process
MMBP	Markov Modulated Bernoulli Process
MMFP	Markov Modulated Fluid Process
OSI	Open System Interconnection
PGPS	Packetized Generalized Processor Sharing
PHB	Per-Hop Behavior
QoS	Quality of Service
SLA	Service Level Agreement
TCP/IP	Transmission Control Protocol / Internet Protocol
VC	Virtual Circuit
WFQ	Weighted Fair Queuing

Chapter 1

Introduction

1.1 Multi-service Network

Current telecommunication networks are expected to carry not only voice, but also video, data and multimedia applications. These applications have various transmission requirements. Voice and video need near real time delivery where small amount of bit errors are tolerable. Email and ftp don't need to be received immediately, but no errors are permitted. Some applications such as online trading are both time and error sensitive. To deal with different type of services, traditionally, two types of switched mode are applied.

Circuit Switched and Packet Switched

Typically applied in telephone networks, circuit switched mode provides a connection between two ends. This connection occupies dedicated resources. Once the connection has been set up, these resources cannot be used by other connections even though they are not currently being used. This feature guarantees the bandwidth for real time traffic.

In packet switched mode, data is chopped into small pieces before being sent via a

shared circuit to its destination. A packet switched network does not allocate dedicated resource or bandwidth to each connection. All go through the same channel. The principle behind it is *statistical multiplex* [43]. Since it is unlikely that all the transmitters send data at the same moment, less resources are needed than in circuit switched networks to process same amount of connections. Packet switched mode, therefore, increases bandwidth utilization. But on the other hand, if all connections tries to deliver the data simultaneously, although it is unlikely to happen, data delay and loss may happen.

Compared to circuit switched mode, packet switched mode evolved faster in recent decades. The oldest packet switched network - X.25, although obsolete, systematically described three-layer structure based on Open System Interconnection (OSI). It is efficient in non-real-time data transmission. Frame Relay simplified X.25 to adapt the introduced faster, reliable transmission media. ATM (Asynchronous Transfer Mode) [43], introduced in late 80s, can carry different type of services such as voice, data, and video. It has become the core technology of B-ISDN (Broadband Integrated Service Digital Network).

From another branch of packet switched technology, TCP/IP , which came from Advanced Research Projects Agency (ARPANet) and prevailed after the mid-90s, brought us the Internet revolution.

From the trend we have seen, the packet switched networks are integrating the function of traditional circuit switched networks, and are becoming a uniform platform carrying all types of services.

ATM and IP

ATM was standardized by the CCITT, currently called ITU-T, as the multiplexing and switching principle for the broadband integrated services digital network (B-ISDN). Strictly

speaking, it is a hybrid of both packet and circuit switched technologies. In the sense that ATM uses cells as its basic unit for transmission, it is closer to packet switched mode. Compare to the fact that traditional packet networks were designed primarily for non-real-time data traffic, ATM from the beginning built in the real-time Constant Bit Rate (CBR) and Variate Bit Rate (VBR)[3] traffic to support the voice and video transmissions. The concept of virtual circuits results in a clear and simple Quality of Service (QoS) [3] guarantee, which will be introduced in the following section. Not only limited in Wide Area Network, ATM also carries the fast traffic in Local Area Network which can take place of Ethernet and 802.5. The infrastructure is designed to carry all type of services, including voice, video, data, and multimedia traffic. ATM was considered to integrate all telecommunications needs on one platform.

But the birth of fast Ethernet and giga-bit switches, the slow standardization in ATM, and more importantly, the huge success of Internet defeated the dream of telephone companies. Internet Protocol (IP), the core technology of Internet, is more affordable, easier to deploy, less regulated, and therefore, more popular.

Unlike ATM designed for telecommunications industry, which focused on an exquisite infrastructure, IP is simpler in both administration and transmission mechanisms. It is easy to implement. But from the other side, the explosive increase of Internet raised many issues, such as addressing, routing, security, etc. One notable problem is the IP technology is based on *best effort* service mode, which means IP makes every effort to deliver a packet. If something goes wrong and a packet gets lost, corrupted or in any way fails to reach its destination, the network does nothing. IP makes its best effort with no delivery guaranteed. Although some applications are more tolerable and have strong error correction schemes and low requirement of delay limit such as ftp and email, the best effort mode clearly does

not satisfy the needs of many new applications such as VoIP and video streams. Therefore, the Quality of Service has been introduced in IP networks .

1.2 Quality of Service

What is QoS?

Quality of Service initially is designed in ATM technology to specify the guaranteed throughput level, which provides a great advantage over competing technologies such as Frame Relay or Fast Ethernet. The primary goal of QoS is to provide priority including dedicated bandwidth, controlled jitter and delay (required by some real-time and interactive traffic), and improved loss characteristics. Another goal is to make efficient use of network equipment by carrying all different types of traffic via the same physical network. In ATM, QoS is characterized by six parameters which are peak-to-peak cell delay variation, maximum cell transfer delay, cell loss ratio, cell error ratio, severely errored cell block ratio, and cell misinsertion rate [3]. ATM defines five service classes of services. The topology of small fixed-size cells transported over a virtual connection makes QoS a build-in function. QoS negotiation is made when a virtual circuit is set up. This is done by including information in the signaling messages that are sent at VC setup time. Borrowed from ATM, RSVP (Resource Reservation Service Protocol) [11] signaling protocol was developed by Integrated Service working group under IETF as the first attempt to provide QoS in TCP/IP networks. There are two basic approaches:

Integrated Services (IntServ)

The Integrated Services architecture [8] for IP networks is based on per-flow resource reservation. To receive resource assurance, an application has to make a reservation before it can transmit traffic into the network. There are two service models defined: the *guaranteed service* and the *controlled load service* [9]. The guaranteed service is defined to support hard real-time applications. It works under deterministic worst-case delay bound through strict admission control, fair queuing and scheduling, which means the guarantee must be preserved under all conditions. To implement this service, per-flow queuing mechanisms and signaling support must be implemented at all routers in the network. By comparison, the controlled load service supports some soft real-time applications and customers who need a better best-effort service. It uses more relaxed mechanisms than guaranteed service does. The RSVP [11] was developed for signaling an application's requirements to the network and for setting up resource reservation along the path.

The IntServ model has not been successfully implemented in the Internet. The main concern is the scalability. IntServ relies on all the nodes along the path implementing the IntServ per-flow classification and scheduling. Given the huge size and heterogeneous state of the Internet, the IntServ could be too difficult to achieve. Besides, even with the support of equipment, considering the amount of service providers, the authorization, authentication and accounting (AAA) among them are almost impossible. This is because most service providers are currently connected through bilateral agreement. An Internet-wide agreement is hard to be made.

Observing the complexity and scalability issues in IntServ, IETF formed a new working group to develop a simpler and more scalable approach to offer a better than best effort service, called Differentiated Services [5].

Differentiated Services (DiffServ)

Instead of making per-flow reservations, DiffServ [5] architecture uses a combination of edge policing, provisioning, and traffic prioritization to achieve service differentiation. The edge of a DiffServ network is responsible for mapping packets to their appropriate *forwarding classes* based on the *service level agreement* (SLA) [5][6] between the service provider and user. Rather than using a protocol like RSVP to tell all the nodes that a packet has certain forwarding class, it would be easier if the packet identifies itself during transmission. This identification is called *differentiated services code point* (DSCP) [4]. It is 6-bit information tagged in the IP header. Routers process each DSCP code in a different manner called per-hop behavior (PHB) [12], a term that indicates the behavior of individual routers rather than end-to-end services.

DiffServ does not require resource reservation setup. Since the service differentiation happens only on edge nodes, the functions that interior nodes must perform to support DiffServ are relatively simple. This architecture eliminates many concerns of scalability, but cannot provide a hard QoS guarantee.

Another technology Multiprotocol Label Switching (MPLS) is also proposed by IETF [52] which gives a labeled switching solution for QoS guarantee.

1.3 Traffic Management

Traffic management includes a set of mechanisms that enable a network to meet a diverse range of service requests efficiently. Two basic goals of traffic management are service satisfaction and resource efficiency. However these two goals are mutually contradictory. We cannot have unlimited resource to provide best services; on the other hand, only focusing

on resource efficiency will deteriorate the service quality. In this section, we will discuss traffic management from three different aspects shown in Figure 1.1.

Figure 1.1: Traffic management

1.3.1 Resource Sharing

Resource *sharing* is the fundamental feature of packet switched network. Based on a best effort service model, all applications share the same channel on a first-come-first-serve (FCFS) basis, regardless of the real time nature of each application. By contrast, circuit switched networks allocate resources in *isolation*, which guarantees the dedicated resources for the QoS. To provide QoS in packet switched networks, similar mechanisms used in circuit switched networks should be apply. Therefore, packet scheduling and queuing disciplines are required for achieving isolation in a shared environment.

In the circumstance of insufficient resources to meet demand, the so-called *max-min fair sharing* [14] is the policy of fair sharing resources. The max-min fair sharing tries to maximize the minimum share of a flow whose demand is not fully satisfied. Each flow will be given a weight and demands of flows will be satisfied based on this weighting. Once the

fair share is satisfied, the remaining capacity (if any) will again be distributed based on the weights of unsatisfied flows.

Typically, a processing server is idle only if there are no packets waiting for transmission in all active flows. Such a server is called a *work-conserving* [14] server. It means that if some flows cannot use their quota of resources, other flows can use them for their sending.

To achieve max-min fair sharing, a fluid model scheduling discipline developed called, *Generalized Processor Sharing (GPS)* [15] [18]. GPS is an ideal fair-queuing scheme based on the fluid work-conserving model. The details of GPS is explained in Section 2.1. A fluid model assumes a server can process the data with an infinitesimal amount from an infinite number of flows. Since real servers must process data packet-based, the GPS server doesn't exist in the real world. However, it provides an important tool to study traffic management via its ideal fair-queuing feature. Many scheduling schemes are also proposed to emulate the GPS, such as Weighted Fair Queuing [15](WFQ, or PGPS) and virtual clock [19].

Weighted Fair Queuing was actually proposed before GPS. The equivalent GPS version is Packetized Generalized Processor Sharing (PGPS) [15]. WFQ tries to emulate GPS in a packet-based real server instead of a fluid model. There are also many variants such as Worst-Case Fair WFQ, Weighted Round Robin, Deficit Round Robin [14], etc.

Scheduling disciplines provide QoS by isolating flows in a sharing environment, but only if there is enough resource available. Since the network resource is always limited, the admission control for accessing the network resource becomes critical in traffic management.

1.3.2 Connection Admission Control (CAC)

In order to offer guaranteed QoS for an ongoing traffic, a network has to allocate enough resources for it. When sufficient resources are not available, the network should reject the connection requests. There must be some mechanisms to control the admission of networks in order to provide QoS and bandwidth efficiency.

CAC tries to answer the following question: “Assuming a new connection request coming, can the network accept it without jeopardizing the guarantees of QoS for current connections and efficiency of bandwidth usage?” But the answer is not simple.

In circuit switched networks, the CAC is relatively simple. The CAC compares the available idle circuits in the network with the needs of new connections. If enough circuits exist, call requests are accepted, otherwise, they are rejected.

However the situation in packet switched networks is more challenging. The applications may have varying transmission rates. They could be extremely bursty. If CAC uses deterministic scheme that is applied in circuit switched networks, there would be a huge waste of bandwidth. To find a better solution, statistical multiplexing CAC schemes are used rather than deterministic schemes. One main stream is using effective bandwidth [39][40][34] or equivalent capacity [20] to determine the admission control. This effective bandwidth or equivalent capacity is a bandwidth that bursty source needs for the transmission with specified loss ratio. Since it is a rate between the peak rate and mean rate of the source, effective bandwidth scheme achieves better bandwidth utilization. In recent years, the artificial intelligence (AI) has been introduced into CAC control. In this category, neural network and fuzzy logic have been investigated for more accurate and intelligent network control. We will review these CAC schemes in the next chapter.

CAC is not isolated from other elements of traffic management. Different scheduling

disciplines can strongly affect the QoS parameters such as loss ratio and delay. Another topic in traffic management, traffic characterization, is also important because all admission control decisions depend more or less on the statistical profile of the requesting connections. The more a CAC knows about a connection request, the more accurate the decision is made.

1.3.3 Traffic Characterization

Traffic characterization is well-studied in the telecommunication industry. In telephone networks, the calling time, connection length, peak and mean volume are crucial in network design and topology. Probability theory and stochastic processes are major mathematic tools to model traffic behaviors. Traditionally, the traffic model in telephone system is based on the Poisson process. The time interval between two successive calls is exponentially distributed.

Multi-service networks like IP and ATM support many kinds of services, among which the traffic could be bursty. The traditional models, Poisson or Bernoulli distribution, did not capture the notion of burstiness of the new type of applications. Therefore, one major shift uses on/off models to represent the busy applications, such as Interrupted Poisson Process (IPP) [20]. In an on/off model, packets are generated only during on periods, but not during off periods. One problem is that an IPP does not capture the characters of correlation for successive inter-arrival packets. As a result, Markov Modulated Poisson Process (MMPP), Markov Modulated Bernoulli Process (MMBP) and Markov Modulated Fluid Process (MMFP) [20] have been developed to allow one to simulate and analyze the network performance. However, these models need complex computation and, more importantly, they cannot possibly cover all the traffic types.

To compound the mathematical models in multi-service networks, the *traffic descriptor* are standardized in ATM with a set of parameters such as peak rate, mean rate, cell delay variation tolerance and burst tolerance. These parameters are not adequate to characterize the connections, but they are easy to monitor and specify.

In general, traffic characterization is yet an open issue.

1.4 Problem Statement

This thesis focuses on the study in CAC for multi-service packet networks. CAC is the pro-active control in traffic management. It is the first step to guarantee the QoS. The goal of CAC design, as described above, is to balance two contradictory objectives, maximizing the bandwidth efficiency and delivery of QoS guarantees for admitted connections [21]. Since the CAC is the first step of traffic control in the provision of network resources to connections, it needs to work with both traffic characterization and scheduling and forwarding schemes. A good CAC design should consider the following requirements:

- **Resource utilization and QoS provision.** These are two counterparts between which CAC have to balance perfectly: Maximizing the resource utilization while achieving guaranteed QoS is the goal of CAC.
- **Implementation complexity and scalability.** Admission control decision has to be made in near real time.
- **Independency from traffic characterization.** Ideally, CAC should not depend on any traffic model, although the specific CAC for one kind of model may meet the best performance in certain circumstances. Since the full understanding of traffic

characterization is still an open issue, and unpredictable new applications are yet to come, CAC should be independent of any traffic models.

1.5 Thesis Contribution

In this thesis, we propose a novel connection admission control scheme in multi-service packet switched network using fuzzy logic. The control system is based on fuzzy knowledge based control (FKBC) [24][26][27]. This thesis has the following contributions:

- We propose a CAC scheme which is not based on any assumption made regarding the statistics of the traffic yet it maintains a good performance in terms of QoS guarantees and system utilization.
- We give a simply solution using fuzzy logic technology in CAC decision. It is easy to implement, in contrast to the complex computation requirement of effective bandwidth methods.
- We show the proposed CAC with self-adaptive and self-learning abilities. It provides great flexibility and can apply “thumb of rules” into automation control.
- We classify accepted connections into several classes. Therefore, customers with different demands can get different services. The service operators can also maximize the resource utilization.

1.6 Organization

This thesis is organized as follows: Chapter 1 introduces the importance of connection admission control, states the requirement of CAC schemes and our objective. Related

background literature is reviewed in Chapter 2. In chapter 3, we propose an admission control scheme based on fuzzy knowledge based control (FKBC). The framework is given first, then each module is explained in detail. Chapter 4 gives simulation results, performance evaluation and comparisons, while Chapter 5 draws a conclusion and addresses future work.

Chapter 2

Literature Review

2.1 GPS - Ideal Scheduling Discipline

Scheduling discipline is an important mechanism in traffic management. A scheduler determines when and which packets get transmitted. It provides QoS, guaranteed throughput, bounded end-to-end delay and jitter and limited loss probability. It also allows statistical multiplexing. In traffic management, the connection admission control is the first phase of traffic control which has to cooperate with scheduler. Thus, scheduling discipline directly affects the design of connection admission control scheme. A good scheduling discipline should be fair and easy to implement. It must provide performance guarantee and flow isolation.

Generalized Processor Sharing (GPS) is an ideal scheduling discipline. It was proposed in 1993 by Abhay K. Parekh and Robert G. Gallager [15]. The idea comes from the natural generalization of uniform processor sharing [18]. GPS gives the ideal scheduling properties - perfect flow isolation and genuine fairness. To understand the GPS scheduling model, let us first see the difference between a real server and an ideal server.

Figure 2.1: Fluid service model

In the real world, the minimum transmission unit that a server processes is a packet. (Figure 2.1(a)). The packet cannot be disassembled. The server cannot process packets from other flows until the current packet has been transmitted. Figure 2.1(b) is a GPS server. The GPS multiplexing is defined with respect to a fluid-model, where packets are considered to be infinitely divisible. A GPS server can process an infinitesimal amount of data from each non-empty flow. Thus it is also called Fluid Fair Queuing (FFQ).

A GPS server is a work-conserving server which means the server must be busy if there are packets awaiting in the system. It operates at a fixed rate. Parekh defined *backlogged flow* as a non-empty queue at time t . Let $B(t)$ denote the set of backlogged flow. For any time interval $[t_1, t_2]$, where $B(t)$ is non-empty during this interval, the bandwidth is shared equally among the backlogged flows, which is called *processor sharing*. Since some flows expect to receive more service than others, to generalize the above sharing, each flow is assigned a weight. The share of bandwidth reserved to flow i is represented by a real number ϕ_i .

Let $W_i(t_1, t_2)$ denote the amount of traffic from flow i during the time period $[t_1, t_2]$.

Given an interval $[t1, t2]$, and for any pair of backlogged flows i and j during this interval, they have

$$\frac{W_i(t1, t2)}{W_j(t1, t2)} = \frac{\phi_i}{\phi_j} \quad (2.1)$$

This is known as a *Generalized Processor Sharing* (GPS) server. The flow i has a guaranteed rate of

$$\alpha_i = \frac{\phi_i}{\sum_j \phi_j} C$$

If we consider ϕ_i represents the portion of the service rate C assigned to flow i , then

$$\sum_{i=1}^n \phi_i = 1$$

As an ideal model, GPS serves flows in a round-robin fashion [14], with an infinitesimal amount of data from each non-empty flow. In any finite time interval, it visits every flow at least once. It provides ideal fairness and complete isolation of flows. But it is not implementable because the real server cannot serve infinitesimal amount in a non-empty flow. Therefore various GPS emulations have been proposed during the past decade. Packetized GPS (PGPS, also known as Weight Fair Queuing) [15][17] and Virtual Clock [19] are both implementable approximation for GPS.

Since GPS is an ideal scheduling discipline, it provides a best platform for us to perform CAC study. It also gives a reference point to be used as benchmark to evaluate different CAC schemes.

2.2 CAC Schemes Review

In conventional circuit switched networks, call admission control is based on the availability of circuits for the desired route. Multiplexing is on circuit level. Compared to circuit switched networks, in multi-service packet switched networks like ATM and IP networks, the CAC is more complex due to the burstiness of the traffic, different service quality demands, and insufficient traffic statistical information.

The *deterministic* or *peak rate* allocation [20] is based on the maximum bit rate specified by the sources during call setup time. A new connection is accepted only if the sum of the peak rates of all ongoing connections and of the requesting one is less than the capacity of total bandwidth, just like the CAC in dedicated circuit switched networks. Deterministic allocation is easy to accomplish and provides absolute QoS guarantees, also requires minimal processing. It is suitable for constant bit rate (CBR) [3] transmission such as uncompressed audio and video. Since most traffic in ATM or IP networks are not of this type, poor bandwidth utilization is inevitable.

CAC based on the peak rate allocation results in waste of valuable network resources. To increase the utilization, some CAC schemes allow the sum of all peak rates to be greater than the total bandwidth. This is called statistical multiplexing. The resource utilization can be substantially improved especially when there is large number of bursty flows in the network.

Generally, statistical allocation admission control can be classified into two categories: *parameter based* and *measurement based* [14]. In the parameter based approach, a traffic descriptor is used to characterize traffic flows. It specifies the connection's statistical profile and includes peak rate, average rate, cell delay variation for the peak rate, maximum burst length, etc. Based on the traffic descriptor as well as traffic model, such as

MMPP/D/1/K, MMFP/D/1/K [20], etc. CAC sets up a set of parameters to make the decision. The strength is that this approach deals with specific type of traffic, which can maximize the QoS guarantee rate and system utilization. The weakness is also obvious. First, acquiring an exact knowledge of traffic statistics is difficult. Traffic characterization is still an open issue. Second, since it is traffic model oriented, every new traffic model needs a new set of parameters for CAC decision, which raises scalability concerns. Third, the lack of real time traffic monitoring may undermine the efficiency because of the users misbehavior i.e. sending more traffic than it is declared.

The measurement based approach does not rely on a *priori* traffic characterization. It measures the real traffic load to make admission control. It has drawn more attention because of increasingly unpredictable network traffic. Since the network applications are not static, the measurement approach cannot predict the prior traffic beyond the measuring time. Therefore, it is always probabilistic in nature. The measurement based approach cannot one hundred percent guarantee the required QoS.

Among the various CAC approaches proposed in the literature, we focus on two major classes: effective bandwidth method and artificial intelligence method.

2.2.1 Effective Bandwidth

The effective bandwidth scheme is similar to the admission control in a multi-rate circuit switched network. In a circuit switched network, the admission decision is based on the comparison of the sum of the current connection rate and the requesting connection rate with the trunk capacity. The procedure is very simple and straightforward. Since, it does not work well in bursty environment because of the low bandwidth utilization, people started to find out the bandwidth that a bursty source really needs, which is called effective

bandwidth. Effective bandwidth represents the bandwidth a connection actually needs in a statistical multiplexing and queuing environment. It is a rate between its peak rate and average rate. As a result, the sum of peak rates of connections admitted into a system may be greater than the total bandwidth, which provides statistical multiplexing. Since this CAC scheme is very simple, it draws a lot of attention.

After the first paper of effective bandwidth published by Hui [39] in 1988, many different methods have been proposed to compute effective bandwidth. A widely accepted mathematical framework for effective bandwidth of a stationary arrival process has been defined by D. Kelly [40] as

$$a(s, t) = \frac{1}{st} \log E[e^{sX[0,t]}] \quad 0 < s, t < \infty$$

Where s is the space-scale parameter and t is the time-scale parameter, $x[0, t]$ denotes the amount of data that arrives from a source during the interval of length t .

However, the computation of effective bandwidth is difficult. It needs to have sufficient statistical profile of coming requests, which is often very difficult to get. Another barrier is that the computation of effective bandwidth is complex. The fast decision is hard to make.

Since full statistical profile of traffic is hard to obtain, the measurement based effective bandwidth method has drawn more attentions. One of them issued by S. Crosby, *et al.* is called Mosquito algorithm[34][35].

The Mosquito algorithm proposed by S. Crosby, *et al.* avoids the necessity of fully understanding the traffic's statistical behaviour. Instead, a near approximation of effective bandwidth is theoretically derived from Large Deviation principle according to online measurement. The algorithm makes no assumption for the characterization of a new connection request. It uses online estimates of the *entropy* to derive an estimation of the bandwidth

requirement of the existing load. By simply adding the peak rate of a new connection to the estimated bandwidth, the Mosquito algorithm compares the sum of these bandwidths with the system bandwidth. If there is enough system resource, then the new connection is accepted; otherwise, it is rejected.

Crosby, *et al*, [34][35] approximated effective bandwidth based on the function

$$\hat{s}(b, c) = -b \frac{\lambda(-\log(c/b))}{\log(c)} \quad (2.2)$$

where b is the buffer size and c is the target bit loss ratio constraint. $\lambda(\theta)$ is called scaled cumulant generating function (SCGF), which is defined by

$$\lambda(\theta) = \lim_{t \rightarrow \infty} \frac{1}{t} \log E(e^{\theta A_t}) \quad (2.3)$$

Here A_t is the total arrivals up to time t . Crosby, *et al*, [35][34] found that the thermodynamic entropy of the traffic is nothing more than a large deviation rate-function.

The Mosquito algorithm estimates the value of SCGF of an observed traffic stream by dividing the time axis into blocks of finite length B and calculating the number of arrivals $\tilde{X}_k$ in each block $k = 1, 2, \dots$. An estimation of SCGF is then given by

$$\hat{\lambda}(\theta) = \frac{1}{B} \log \frac{1}{K} \sum_{k=1}^K e^{\theta \tilde{X}_k} \quad (2.4)$$

As a new connection request arrives, the system estimates the bandwidth requirement of the current multiplex of connections, then adds the needed bandwidth the new connection requires, which could be as simple as peak rate. If the estimated bandwidth requirement is less than the link capacity, then the connection can be accepted without violating the QoS of any calls. As soon as the new call starts to transmit, the estimator revises the estimation

of the current new complex using online measurement.

2.2.2 Artificial Intelligence (AI) Method

Recently, artificial intelligence applied to automation control has drawn more attention because of its adaptive and self-learning properties [21][23]. Among these, fuzzy logic and neural networks have been studied extensively. Both of them are proven robust. AI techniques have been used successfully in engineering automation control. Once sufficiently programmed and trained, they are accurate and quick. The implementation and computation complexity is low. Both fuzzy logic and neural networks are known to have modest computational requirements [21]. These characteristics make fuzzy logic and neural networks very attractive in real control systems.

Fuzzy logic and neural networks have been studied for computer networks such as congestion control [49] and active queue management [50].

Fuzzy logic systems are especially useful when the full traffic profiles are practically unavailable, which introduces considerable vagueness to the CAC decision. L. Barolli, *et al*, proposed a fuzzy admission control scheme for high speed networks [42]. They gave a theoretical framework for applying fuzzy logic to CAC. R. Srinivasan and J. Talim proposed a bandwidth allocation scheme based on the fuzzy logic with limited knowledge of traffic profiles [41]. The bandwidth allocation is based on a *User Requirement* [41] which uses peak rate, mean rate and QoS constraint to classify the traffic. In this thesis, our CAC scheme is the extension of the bandwidth allocation to connection admission control.

2.3 Fuzzy Logic

In 1964, Lotfi Zadeh previewed his first paper about fuzzy logic to his Berkeley colleagues. It was attacked immediately, [24] consequently, had not even appeared in print. Until 1970s with successful stories in Japan, fuzzy logic began to be accepted as a serious discipline.

Fuzzy logic, an interesting and confusing name, is not a "fuzzy" logic, but logic that defines and describes fuzziness in mathematical method. The fundamental concept of fuzzy logic is the idea that all things come in degrees. Height, age, color, beauty, sadness - all come on a continuously sliding scale. The NBA stars are tall. Algonquin Park is beautiful. The winter in Ottawa is cold. Such sliding scales often make it impossible to distinguish members of a class from non-members.

Traditional logic, set theory, and philosophy have compelled sharp distinctions. For instance, a traditional way to define "mountain", let us say, is a natural elevation of the earth's surface having a height 500 meters or more, while a "hill" is less than 500 meters height. By this standard, a 500.01m height elevation would be a mountain, and a 499.99m is a hill. Thus, such a hill 499.99m height measured in the summer would become a mountain in the winter just because 2mm snow covers the peak.

Fuzzy logic avoids such absurdities by introducing a new set concept - fuzzy set.

2.3.1 Crisp Set and Fuzzy Set

Traditional set theory was founded by German mathematician Georg Cantor (1845-1918) in the late 19th century. In set theory, the notions "set" and "element" are primitive. They are not defined in terms of other concepts. Any element in the universe of discourse can unambiguously be determined whether it belongs to the set or not. A classical set is a collection of objects of any kind and may be finite, countable or uncountable.

The basic of set theory is clear and simple [24]: a set is a circle which divides the universe into two halves. Each element is either in the circle or not in it. No element straddles the line. Cantor's set theory went on to triumph among mathematicians, but there are some paradoxes traditional set theory cannot explain, like mountain and hill, young and old. There is no proper way to define such kind of objects in traditional set theory.

Since, there's no such dictating line that simply divides the world into true and false, Zadeh founded a way which introduced a new notion - graded memberships. In fuzzy set theory, traditional set is called *crisp* set in contrast to fuzzy set. Fuzzy set could have members which belonged to it partly, in degrees. One simple example is the question of "old man". Suppose John is 61-year-old. Does he belong to old men? Crisp sets ask, "is John old?" and draw a standard, say, 60. Old lies over this age, not old is below. Thus, at age of 61, "John is old". In contrast, fuzzy sets ask, "How old is John?" The answer is a partial membership in the fuzzy set, such as 0.52. So "John is 0.52 old."

Thus, candidates for old men might have the following memberships:

	Crisp	Fuzzy
Tom (90) is old	1	0.9
John (61) is old	1	0.52
Alex(59) is old	0	0.48
Chris(35) is old	0	0.2

Table 2.1: True value table of "old"

The crisp set contains Tom and John. Although Alex is almost the same age as John, he is not in the set. The fuzzy set contains all four and shows the similarity between John and Alex. Fuzzy sets, as a matter of fact, discriminate much better and supply more profile. They are more precise.

The master key to fuzzy sets is the membership values. Fuzzy sets are easier to grasp as

sets of degree rather than sets of members. In fact, fuzzy sets include crisp sets. A crisp set is just a fuzzy one with membership values of 1 and 0. If an item is in a crisp set, it must have a value of 1; if it is “in” a fuzzy set, it can have any value between 0 and 1. Hence, fuzzy operations always work with crisp sets.

2.3.2 Membership Function

Following traditional set theory, to define the set of “old”, let U indicate all people and subset A be the set of “old” people. The subset A of a set U is determined by its indicator function, or characteristic function χ_A defined by

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \quad (2.5)$$

The indicator function of a subset A of a set U specifies whether or not an element is in A . It either is or is not. This notion is generalized in fuzzy set theory by allowing images of elements to be in the interval $[0, 1]$, rather than being restricted to the two element set $\{0, 1\}$ [23][24].

A fuzzy subset A of a set U is determined by characteristic function $U \rightarrow [0, 1]$. For a fuzzy set, this generalized characteristic function

$$\chi_A : U \rightarrow [0, 1] \quad (2.6)$$

is called the *membership function* associated with the fuzzy set A , and the value $\chi_A(x)$ is called the degree of the member x in the fuzzy set A .

Membership function for a fuzzy concept is not unique. Different membership functions can be considered. For example, both of the membership functions shown in Figure

2.2 can represent the fuzzy concept “old”. The choice of the function χ is subjective and context dependent.

Figure 2.2: Two membership functions of “old”

2.3.3 Operations on Fuzzy Sets

In fuzzy set operations, some are the same as traditional set operations, such as commutativity, associativity and distributivity. However due to the graded membership characters in fuzzy sets, some other operations, which include union, intersection and complement, are not like traditional set theory. Zadeh defined them in [30]

$$\forall x \in X : \quad \chi_{A \cap B} = \min(\chi_A(x), \chi_B(x))$$

$$\forall x \in X : \quad \chi_{A \cup B} = \max(\chi_A(x), \chi_B(x))$$

$$\forall x \in X : \quad \chi_{A'}(x) = 1 - \chi_A(x)$$

Other extensions are also possible such as

$$\forall x \in X : \quad \chi_{A \cap B} = \chi_A(x) \cdot \chi_B(x)$$

$$\forall x \in X : \quad \chi_{A \cap B} = \max(0, \chi_A(x) + \chi_B(x) - 1)$$

$$\forall x \in X : \quad \chi_{A \cup B} = \chi_A(x) + \chi_B(x) - \chi_A(x) \cdot \chi_B(x)$$

$$\forall x \in X : \quad \chi_{A \cup B} = \min(1, \chi_A(x) + \chi_B(x))$$

More generally, *triangular norms* are applied to represent fuzzy operations [32], [31]. T-norm represents the most general intersection operator and S- norm (T-conorm) represents the most general union operator. Having the T-norm and S-norm, it can have a great many different models to represent the connectives for fuzzy set. Noticeable is the fact that this situation enables us to choose the most suitable one for our purpose.

2.3.4 Rules

To apply fuzzy logic to automation control, fuzzy rules are applied to express the relationship between process states and control output variables [28]. Each rule is characterized with an IF part, called the *antecedent*, and with a THEN part, called the *consequent*. The antecedent of a rule contains a set of conditions and the consequent contains the conclusion. If the conditions of the antecedents are satisfied, then the conclusions of the consequent apply. For instance, the public transportation equips low-floor buses for the senior people and the disabled. A fuzzy system controls the operations. One rule in the fuzzy control system may be: IF a passenger is an old person and with mobility difficulty, THEN the floor lowers to the curb so that the first step is easier. The age and mobility difficulty are the inputs, and

the lowering distance is the output.

Fuzzy if-then rules have been used extensively in both modeling and control. Through the use of linguistic labels in membership function, a fuzzy if-then rule can easily capture the spirit of a "rule of thumb" used by humans.

To representing the meaning of if-then rules, (most of these relations are derived from multi-valued logic), various kinds of implications are used. A commonly used one is the Mamdani implication [27]. The rule "if x is A , and y is B , then z is C ." is represented as $\chi_C(z) = \min(\chi_A(x), \chi_B(y))$. In the above low-floor buses, if a passenger is old with grade of 0.65, and with mobility difficulty with grade 0.9, then the conclusion is lowering floor with grade $\min(0.65, 0.9) = 0.65$. The implication methods are being used in inference operations in the so called fuzzy knowledge based control (FKBC) which will be introduced below.

2.3.5 Fuzzy Knowledge Based Control

Fuzzy knowledge based control is composed of six functional blocks described in Figure 2.3:

- A scaling block performing a scale transformation which maps the physical values of the current process state variables into a normalized scale.
- A fuzzification interface which transform the crisp inputs into degrees of fuzzy sets, in order to match fuzzy if-then rules.
- A knowledge base containing a number of fuzzy if-then rules and the membership functions of the fuzzy sets used in the fuzzy rules.

- A rule firing unit which performs the inference operations on the rules to make the decision.
- A defuzzification interface transforming the fuzzy results of the inference into a crisp output.
- A denormalization block which maps the normalized crisp output back to its physical domain.

Figure 2.3: The structure of a FKBC

Generally to say, the design of a FKBC system needs following jobs to do.

1. Choose variables and content of rules.
2. Find proper term set.
3. Define membership functions.
4. Choose scaling factors.
5. Define inference implication.

6. Choose defuzzification methods.

The following example shows how to design a FKBC system.

A fuzzy control system is to operate the descending height of low-floor bus. The bus lowers floor for senior people with mobility difficulties. The adjustable range of the floor height is 30cm. The input is a person's age and degree of mobility difficulty. The output is the descending height of a bus floor. The term sets we choose here are "old", "not old", "mobility difficulty" and "no mobility difficulty". As a simple example, the control is based on the following rules:

1. if the passenger is "old" and he/she shows "mobility difficulty", then lower the floor to lowest height.
2. if the passenger is "old" but he/she has "no mobility difficulty", then lower the floor half way to its adjustable distance.
3. if the passenger is "not old", then do not lower the floor.

The next step is to define membership functions in order to map crisp inputs into grade of memberships. The membership function of "old" is defined as follows,

$$\chi_{\text{old}}(a) = \begin{cases} 1 & (a \geq 80) \\ \frac{a-50}{30} & (50 \leq a < 80) \\ 0 & (a < 50) \end{cases}$$

where a is the age of the person.

The membership function to describe “mobility difficulty” is a function of his walking speed v ,

$$\chi_{\text{md}}(v) = \begin{cases} 1 & (v < 1\text{km/h}) \\ \frac{5-v}{4} & (1 \leq v < 5\text{km/h}) \\ 0 & (v \geq 5\text{km/h}) \end{cases}$$

The membership function of “not old” and “no mobility difficulty” are the complement of χ_{old} and χ_{md} .

$$\chi_{\overline{\text{old}}}(a) = 1 - \chi_{\text{old}}(a)$$

$$\chi_{\overline{\text{md}}}(v) = 1 - \chi_{\text{md}}(v)$$

Therefore, for an 60-year-old passenger, who walks very slowly (2km/h), the grades of “oldness” is $\frac{60-50}{30} = 0.33$ and his/her “mobility difficulty” grades $\frac{5-2}{4} = 0.75$. According to the rule base, three rules fire. Using Mamdani implication, the first rule gives a decision to lower the floor 30cm with grade $\min(0.33, 0.75) = 0.33$. The second rule decide to lower the floor 15cm with grade $\min(0.33, 0.25) = 0.25$. The third rule is doing nothing with grade 0.67.

The next step defuzzification combines the decisions made by all fired rules. The method of defuzzification can be different. The height defuzzification method [27] is proper for this example. This method calculates the output as $\frac{0.33 \cdot 30 + 0.25 \cdot 15 + 0.67 \cdot 0}{0.33 + 0.25 + 0.67} = 10.92\text{cm}$. Therefore, the bus lowers the floor 10.92cm for this person.

In this example, no scaling and denormalization are needed since the there is no change in the domain.

Chapter 3

A CAC Scheme Using Fuzzy Logic

In this chapter, we proposed a novel CAC scheme based on fuzzy logic control. The study uses a single node GPS server. The QoS criteria is BLR [3], the bit loss ratio due to buffer overflow. This CAC is a measurement based scheme, without any assumption of traffic statistics. It is also simple and fast in the CAC process. In the following sections, we will discuss it in detail.

3.1 Notation definition - S , Q and μ

As mentioned in Section 1.4, two fundamental objectives of a CAC is the service satisfaction for customers and resource utilization. The former can be defined by the proportion of connections whose QoS requirements are satisfied. We name it *customer satisfaction rate* and it will be denoted by Q

$$Q = \frac{(\text{Number of connections whose QoS are satisfied})}{(\text{Total number of connections})} \times 100 \quad (\%) \quad (3.1)$$

The latter deals with the unused bandwidth, denoted by μ , which is defined as follows,

$$\mu = \frac{100}{t} \int_0^t \frac{(\text{Unused bandwidth at time } i)}{C} di \quad (\%) \quad (3.2)$$

where t indicates the measurement interval. Note that $(1 - \mu)$ provides the resource utilization.

The primary job of a CAC is to maintain a high level of Q , while at the same time tries to keep the unused bandwidth μ as low as possible.

While making no assumption of traffic model, there is limited information of the sources that can be obtained. Each source is characterized by its peak rate, mean rate and QoS constraint. The first two can be observed through leaky bucket [38] measurement, while the QoS constraint is a set of parameters specified by customers. In this thesis, we focus on one QoS parameter - BLR [3].

Since the BLR is the single parameter for QoS, we define the value of QoS as a positive integer which represents the $-\log(BLR)$. For example, a QoS of 5 means that the BLR is 10^{-5} .

We may think that Q represents how successful the system is in the guarantee of the QoS for admitted connections, while $(1 - \mu)$ represents the status of system utilization. None of the two parameters describe the traffic profile. We introduce a third factor describing the burstiness of the source and we define $\frac{\text{Peak_rate}}{\text{Mean_rate}}$ ratio as the burstiness index. It is easy to observe that a higher $\frac{\text{Peak_rate}}{\text{Mean_rate}}$ ratio means sources burstier; the system probably needs to allocate more resource to guarantee the QoS, which may yield low utilization of the bandwidth. If the resource is limited, QoS guarantee for bursty sources are harder to satisfy. On the other hand, the BLR requirement should be taken into consideration because the more restrictive the loss ratio is, the higher the capacity requirement, and the

more difficult the guarantee of the QoS is. Therefore, using known peak rate p_i , mean rate m_i and bit loss ratio 10^{-l_i} , given n number of connections currently existing in the single node system with service rate C , we define the *Source Characteristic Factor*,

$$S = \frac{\sum_{i=1}^n \frac{p_i}{m_i} l_i}{n \cdot C} \quad (3.3)$$

We notice that S represents the aggregated statistical characterization of all ongoing sources. When S is a large number, it means that either the source is burstier or loss ratio requirement is stricter. The value of S represents some of the statistical characterizations which are easy to obtained. The CAC that we propose here will use this factor S to estimate the Q and μ values. The estimations form the input to a fuzzy control module to achieve an easy, fast and effective connection control.

3.2 Framework

The framework of the fuzzy CAC is shown in Figure 3.1. It consists of four functional modules: queuing and forwarding module, Q and μ estimation module, real-time measurement module and fuzzy logic control module. Let us first describe these four modules briefly.

The top block is the queuing and forwarding module. It works under the GPS scheduling discipline. Each connection is allocated an individual buffer. GPS sets up different weights for each connection according to its QoS requirement and the occupancy of the buffer. Here, we consider that the bit loss only happens due to the buffer overflow.

The lower left block is the estimation module. This module has the task to estimate QoS satisfaction rate Q and unused bandwidth μ . This estimation is based on the factor S which

Figure 3.1: System framework of fuzzy logic control in CAC

takes into account the peak rate, mean rate and QoS constraint of a new arriving source, and the empirical curves that show how the burstiness of traffic can affect the bit loss ratio and bandwidth utilization. The generation of empirical curves is investigated in Section 3.4. The noticeable aspect is that there is no assumption for the traffic model and characterization. Therefore, the estimation has no dependency on known traffic statistics. This module can function as a simple CAC. It calculates the source characteristic factor, then estimates the possible loss ratio and bandwidth utilization. If both meet the requirement, the new request is accepted, otherwise, the request is rejected.

Besides the estimation of the system behavior, the measurement of real time statistical profile is also used in CAC decision. The real time measurement module is shown at the lower right corner of Figure 3.1. This module monitors two metrics: the loss ratio and unused bandwidth. From these metrics, real time Q and μ values are obtained. These two values, along with the estimated Q and μ , form the input variables of the admission control module.

In the center block, the fuzzy CAC controller takes the estimated customer satisfaction rate Q and unused bandwidth μ from the estimator module as inputs, along with the corresponding actual values from the real time measurement module. Using these variables, the CAC module outputs the decision of “accept” or “reject”. An “accept” decision can further be classified according to different levels. In this thesis, we implement a two class admission. The internal process in this module is implemented in a fuzzy way, which means, the inputs need to be fuzzified into fuzzy membership grades. Also the output comes from a defuzzification procedure. The whole process follows the FKBC structure [27] which is explained in Section 2.3.5. In this module, the fuzzy rules are essential to describe what kind of sources can be accepted according to the current system utilization and loss ratio. A so-called inference engine matches the inputs with conditions in fuzzy rules. The matched rules are fired. Since each single input has more than one state in a fuzzy world, there might be several rules fired simultaneously. The output of these fired rules are combined together through certain defuzzification methods. Therefore, a unique admission decision is made. The detail of this module is described in Section 3.6.

The complete admission procedure is as follows:

1. A coming request activates the CAC mechanism. During this time, the system collects the basic statistical profile, which are peak rate and mean rate. This profile

can be either given by the customer, or measured through some mechanisms such as the leaky bucket method [38]. The system also need to know the QoS constraint indicated by the customer. This information is input to the estimation module.

2. The estimation module gets the peak rate, mean rate and QoS constraint from the requesting connection. The source characterization factor S is calculated. This value is the aggregation of all current and requesting connections. From estimation curves, the estimated value of Q and μ are generated through an empirical function. These values show the predicted customer satisfaction rate and unused bandwidth if the connection were to be admitted.
3. The real time measurement keeps monitoring the entire system bandwidth usage and the loss ratio for each connection. The actual values of Q and μ are provided to the fuzzy CAC module to carry on the admission decision.
4. From estimation module and real time measurement module, four variables Q_e , μ_e , Q_a and μ_a form the input of the fuzzy logic control module. Through membership functions, these variables are fuzzified to grades of states such as “high” and “low”. An inference engine matches rules in fuzzy rule base and fires them. Through a defuzzification procedure is to output an admission or rejection decision.
5. If the new connection request is accepted, this connection will be scheduled through the GPS scheduling discipline. A CAC procedure is completed.

Not only each connection has an individual buffer, but also every connection has a service priority class. We propose this CAC scheme providing the differentiated class of services. In the implementation of this thesis, we apply two-class service types. The method can be extended to multiple-class admission type. The more details are described in Section 3.6.

By applying fuzzy logic, this CAC has a self adaptive and self learning mechanism. This is important to keep the decision database updated and accurate. We will discuss them in a separate chapter. In the following sections, we will describe these modules in detail.

3.3 Queuing and Scheduling Module

Queuing and scheduling module is the basis of the entire system. It is not the part of CAC, but a CAC has to be incorporated into certain queuing and forwarding discipline to function. As we mentioned before, the queuing and forwarding discipline used in this thesis is the GPS scheduling discipline. In this module, individual buffers are allocated for each admitted connection. According to the occupancy of buffer and the QoS constraint for each connection, GPS dynamically assigns weight in the allocation of bandwidth. When some connections cannot use up their own portion of bandwidth, the excess bandwidth is allocated to other connections in order to maximize the utilization.

As detailed in Section 2.1, in this GPS scheduler, let $W_i(t)$ denotes the amount of data served in queue i during time slot t , ϕ_i denotes the weight of queue i in processor sharing. We have,

$$\frac{W_i(t)}{W_j(t)} = \frac{\phi_i}{\phi_j}$$

The queue i has a guaranteed bandwidth of

$$\alpha_i = \frac{\phi_i}{\sum_j \phi_j} C \quad (3.4)$$

among the capacity of service rate C . One essential issue is to determine the value of ϕ_i to achieve a fair resource allocation.

Determining the Weight ϕ_i

In this module, each connection can be considered as a queue. To meet the guaranteed QoS of queue i in terms of loss ratio, in time slot t , the weight of each queue is defined as,

$$\phi_i(t) = \frac{a_i(t) + x_i(t)}{b_i} l_i \quad (3.5)$$

where, a_i denotes the current arrival in queue i , x_i denotes the system queue i length, b_i denotes the buffer size, (which is related to the peak rate of this queue,) and l_i shows the bit loss ratio is 10^{-l_i} .

The reason to define ϕ_i as shown in 3.5 is that: 1. Weight depends on time t , weights need to be various in different time. 2. If we have a very large buffer, it is less likely that we will lose packet due to buffer overflow. Therefore the long buffer can be given a lower priority. $\frac{a_i(t) + x_i(t)}{b_i}$ is the occupancy in the buffer of queue i at time t . if $\frac{a_i(t) + x_i(t)}{b_i} = 1$, it means buffer is full, and bit loss is expected; if $\frac{a_i(t) + x_i(t)}{b_i} \rightarrow 0$, the buffer is almost empty, no bit loss possibly happens. 3. The bit loss ratio, represents by l , indicates the QoS requirement. The lower loss ratio is, the higher the weight needs to be set.

Bandwidth Allocation

The bandwidth allocated to queue i at time t is,

$$\alpha_i(t) = \frac{\phi_i(t)}{\sum_j \phi_j(t)} C \quad (3.6)$$

Since GPS scheduler is work-conserving, which means the server is idle only if none of queues are backlogged, $\alpha_i(t)$ is the guaranteed minimum bandwidth allocated to queue i . The excess bandwidth available from the queues not using their reserved portion will be

distributed among all the backlogged queues, as in Section 2.1, in proportion with $\phi_i(t)$.

The excess bandwidth is allocated in the following procedures.

1. The possible excess bandwidth $r_i(t)$ is the allocated service rate $\alpha_i(t)$ minus backlog amount $a_i(t) + x_i(t)$, which is expressed as,

$$r_i(t) = [a_i(t) + x_i(t)] \left(\frac{l_i C}{b_i \sum_j \phi_j(t)} - 1 \right) \quad (3.7)$$

2. If $r_i(t) > 0$, there is excess bandwidth from queue i at time t to be allocated. Sorting $r_i(t)$ in descending order, we have a re-indexed $r'_k(t)$. where, $r'_1 > r'_2 > \dots > r'_n$. At time t , $\sum_j \phi_j(t)$ is a constant. $r'_k(t)$ is therefore in descending order of $\frac{l_k}{b_k}$.
3. Starting from $k = 1$, if $r'_k > 0$, which means queue k cannot use up its allocated bandwidth. The actual needed bandwidth of queue k is

$$\alpha'_k(t) = \alpha_k(t) - r'_k(t) \quad (3.8)$$

The excess bandwidth $r'_k(t)$ is allocated to other queues. The rest queues $l = k + 1, k + 2, \dots, n$ have adjusted bandwidth of

$$\alpha'_l(t) = \alpha_l(t) + \left[\frac{\phi_l(t)}{\sum_{j=k+1}^n \phi_j(t)} \right] \cdot r'_k(t) \quad (3.9)$$

4. Repeat the step 3 with $k = 2, 3, \dots$ until $r'_k \leq 0$, which means no available excess bandwidth could be re-allocated.
5. If there is still excess bandwidth after the bandwidth re-allocation. It is the unused bandwidth in the system during the transmission at time t .

3.4 Q and μ Estimator

3.4.1 Relations between S , Q and μ

In section 3.1, we suggest possible relations between S , Q and μ . This section numerically identifies relations between $S - Q$ and $S - \mu$, based on simulations. The reason to find the relations between S , Q and μ is we need to use S to estimate Q and μ values.

Let us consider an experiment. In a given period of time t , n number of connections are transmitted by the system. The value of n is exponentially distributed with mean 10. Each connections has the peak rate p_i , mean rate m_i and QoS constraint l_i .

We choose sources of connections from thirty traffic samples. One of these has been widely used in performance studies of ATM in the literature, a MPEG coding of the Star Wars movie; some sources are coming from the Internet real traffic [51]; some others are based on stochastic models. Table 3.1 shows examples of traffic sources used in simulations.

The Figures 3.2 and 3.3 show the results of two thousand times of simulation. Each dot represents one simulation result. We plot these results in $S - Q$ and $S - \mu$ planes.

The figures do not show any explicit relations between $S - Q$ and $S - \mu$. Partly, the reason is that we do not have enough sample traffic files which are diverse enough in terms of the value of S . Therefore, we need an alternative perspective to explore their relations. From the observations, we can find with a larger source characteristic factor S , the source tends to be more bursty or less loss tolerable. In either case, the demand of bandwidth for this connection is more than the need for a connection with less S . That causes the unused bandwidth μ to increase. On the other hand, a bursty source makes its QoS constraint more difficult to satisfy, since the behavior of this connection is difficult to predict. Thus, when S

Figure 3.2: Q as a function of S

increases, μ (resp. Q) is likely to increase (resp. decrease). In the next section, we further investigate the $S - Q$ and $S - \mu$ relations from probability distribution perspective.

3.4.2 Q and μ Estimation

Q Estimator

Since the factor S is used to estimate the customer satisfaction factor Q , we investigate a “simple” relationship between the two of them. As mentioned previously, we conjecture that as S increases (or equivalently $\frac{1}{S}$ decreases), Q is likely to decrease. Thus, we introduce the following equation:

Figure 3.3: μ as a function of S

$$Q = v \cdot \frac{1}{S}$$

where v is random variable whose empirical probability function plotted in Figure 3.4. In this figure, the x-axis is in logarithm scale. Given a point on the curve, denoted as (η, ϵ) , we have,

$$\text{Prob} (Q \cdot S < \eta) = \epsilon \quad (3.10)$$

From Equation 3.10, we can derive,

Figure 3.4: $Q \cdot S$ empirical distribution curve

$$\text{Prob} (Q \cdot S \geq \eta) = 1 - \epsilon \quad (3.11)$$

Equation 3.11 shows us that when ϵ is a small number close to 0, the probability that $Q_e \geq \frac{\eta}{S}$ is close to 1, where Q_e refers to the estimated Q value. Therefore, suppose the new connection is arrived, the aggregated S for all connections are calculated. From the value of S , we can estimate that customer satisfaction rate Q_e is greater than $\frac{\eta}{S}$ with the probability close to 1.

From the estimated value of Q , a simple CAC decision can be made: When the estimated customer satisfaction rate Q_e is higher than a threshold to be defined, the new connection request is accepted, otherwise, it is rejected.

Figure 3.5: $\frac{\mu}{S}$ empirical distribution curve

μ Estimator

Similarly, we expect the unused bandwidth μ to increase when S increases, we define the random variable w as:

$$\mu = w \cdot S$$

and whose distribution is derived numerically in Figure 3.5. In this figure, x axis has a log scale. For any point on the curve, denoted as (η, ϵ) , we have,

$$\text{Prob} \left(\frac{\mu}{S} < \eta \right) = \epsilon \quad (3.12)$$

Therefore, for a given value of ϵ close to 1, there is a corresponding value of η . we can get,

$$\text{Prob} (\mu < \eta \cdot S) = \epsilon \quad (3.13)$$

From the Equation 3.13, if the new connection request is accepted, the unused bandwidth μ is estimated to be lower than the product of η and the aggregated S for all connections with a high probability.

The estimation of Q and μ from empirical curves provides a very simple but effective method to predict the coming traffic, but it is not accurate because of the limited knowledge of traffic statistical profile. In the later this chapter, fuzzy logic is applied to make more accurate decision through the estimation and real measurement. The estimation of Q and μ are very important variables as the inputs of fuzzy CAC module.

3.4.3 Linear Regression

Figure 3.4 and Figure 3.5 give the relations between the S , Q and μ . In a real time situation, the estimations need to be done in short time. Therefore, using linear function to approximate the empirical probability curves are necessary. In this section, linear regressions are used to find these fit lines of $Q - S$ and $\mu - S$ curves.

The commonly used method in linear regression [44] to find a best fit line for the curve is the method of *least squares*. This method calculates the best-fitting line for the observed data by minimizing the sum of the squares of the vertical deviations from each data point to the line. Because the deviations are first squared, then summed, there are no cancellations between positive and negative values.

Given a collection of n points (x_i, y_i) , $(i = 1, 2, \dots, n)$, we define a line $\bar{y} = m\bar{x} + b$.

To make this line meet the least squares fit, the slope m and intercept b are calculated as follows [44],

$$m = \frac{\sum_{i=1}^n (x_i \cdot y_i) - \frac{1}{n} \sum_{i=1}^n (x_i) \cdot \sum_{i=1}^n (y_i)}{\sum_{i=1}^n (x_i^2) - \frac{1}{n} \sum_{i=1}^n (x_i) \cdot \sum_{i=1}^n (x_i)}$$

$$b = \frac{1}{n} \left(\sum_{i=1}^n y_i - m \sum_{i=1}^n x_i \right)$$

In practical simulation, to minimize the least squares errors, we cut the heads above 0.95 and tails of lower than 0.05 when finding the fit lines for empirical $S - Q$ and $S - \mu$ curves. The regression lines are shown in figure 3.6 and Figure 3.7.

Figure 3.6: Linear regression for $Q \cdot S$ empirical distribution curve

Figure 3.7: Linear regression for $\frac{\mu}{S}$ empirical distribution curve

3.5 Real-time Measurement Module

Real time monitor measures the activity of the buffer and output port. There are two functions in this module: (shown in Figure 3.8) one is to record the unused bandwidth. Since instantaneous bandwidth utilization changes very fast, a low pass filter is applied to smoothen the curve and to generate the average bandwidth occupancy. The other function is to monitor the bit loss due to buffer overflow, and the total transmitted bits for the each flow. From these data, the bit loss ratio (BLR) is calculated. The BLR is compared with the required loss ratio in QoS constraint. The module counts how many connections are guaranteed with their QoS constraint. The actual customer satisfaction rate, denoted as Q_a , is then calculated. The actual unused bandwidth, denoted as μ_a , and customer satisfaction rate Q_a

are the input to the fuzzy control module for the decision of CAC.

Figure 3.8: Real-time measurement module

3.6 Fuzzy Connection Admission Control Module

The fuzzy admission control module gets the input from real time measurement block and estimator block. This module is based on FKBC structure.[27] It consists of four blocks, shown in Figure 3.9. They are fuzzification, inference engine, FAC rule base and defuzzification module. The function of these blocks are as follows.

- **Fuzzification Block.** This block is the interface of FCAC module. This block needs to convert input variables, which are Q_e , Q_a , μ_e , and μ_a , to fuzzy memberships of “high” or “low” with corresponding grades. For instance, a Q_e value of 0.9 can be represented as “high” Q_e with grade of 0.8, and at the same time can be stated as “low” with grade of 0.2. After fuzzification, there is no numerical value of Q_e , Q_a , μ_e , and μ_a . They are all represented as different states with certain grades.

- **Inference Engine.** It is the engine that implements the control according to the input states. The main function of it is to match the input states with the conditions in the fuzzy rule base. If one rule is matched, the consequence of this rule is executed. Multiple rules may be fired and the results may be not unique. For instance, the fired rules can show the result of the combination of “accept the request with grade 0.1” and “reject the request with grade 0.5”. It will not lead to a self-contradiction. An element can belong to several fuzzy sets with different membership grade.
- **FCAC Rule Base.** It is the database that stores all the fuzzy rules. Fuzzy rules are represented as “if <conditions> then <consequence>” form. When the conditions are true, the consequence is valid. These rules are translations of thumb of rules from human intelligence. Through these rules, the fuzzy logic achieves artificial intelligence.
- **Defuzzification Block.** Since the output has to be restored to a crisp value to implement the control, the defuzzification block reverses the fuzzification procedure to convert the grade of membership to a numeric value. Since more than one rule may be fired, the defuzzification block has to consider all the fired rules to make a final decision. It is more like a human decision made from multiple uncertainty.

Since the input variables Q_e , Q_a , μ_e , and μ_a have “high” and “low” two states, the rules in the fuzzy rule base need to cover all the possibilities. Therefore, the number of rules in fuzzy rule base is 16. For the output decision, three different states are “accept”, “weak accept” and “reject”.

The “accept” class connections have higher priority in scheduling and forwarding. For this class, the system allocates as much resources as possible to guarantee their QoS. The

CAC only “accepts” this class type when it considers that there are enough resources for QoS requirement of these connections.

The “reject” action is made when the CAC decides that for the current available resources, there is no way to guarantee its QoS without deteriorating other ongoing connections.

The “weak accept” is the decision between “accept” and “reject”. There are resources available to accept the new request. But the CAC cannot guarantee the resources are enough for this connection. Therefore, the CAC still accepts the connection request but does not guarantee the QoS constraint it requires. If the resources are not enough during transmission, the “accept” class has priority to use the resources which leaves “weak accept” vulnerable with bit loss.

Figure 3.9: Fuzzy connection admission control module

3.6.1 Membership Functions

Membership functions need to fit the requirement of fuzzy system. Considering that the real time operation needs fast decision, we apply the commonly used Γ -function, L -Function and Π -function [26][27], with underlying domain $[0, 1]$, shown in Equations 3.14, 3.15, 3.16.

$$\Gamma(x; \alpha, \beta) = \begin{cases} 0 & (x < \alpha) \\ \frac{x-\alpha}{\beta-\alpha} & (\alpha \leq x < \beta) \\ 1 & (x \geq \beta) \end{cases} \quad (3.14)$$

$$L(x; \alpha, \beta) = \begin{cases} 1 & (x < \alpha) \\ \frac{x-\alpha}{\beta-\alpha} & (\alpha \leq x < \beta) \\ 0 & (x \geq \beta) \end{cases} \quad (3.15)$$

$$\Pi(x; x_0, x_1, \alpha, \beta) = \begin{cases} \frac{x-\alpha}{x_0-\alpha} & (\alpha \leq x < x_0) \\ 1 & (x_0 \leq x < x_1) \\ \frac{x-\beta}{x_1-\beta} & (x_1 \leq x < \beta) \\ 0 & otherwise \end{cases} \quad (3.16)$$

To keep it simple, term sets for each variable, Q_e, Q_a, μ_e, μ_a , consists of two elements. They are “high” and “low”, denoted as H and L. The membership functions for Q and μ apply L -function and Γ -function as follows,

$$\chi_L(Q_e) = L(Q_e; q_{el}, q_{eh});$$

$$\chi_H(Q_e) = \Gamma(Q_e; q_{el}, q_{eh});$$

Figure 3.10: Γ , L and Π membership function

$$\chi_L(Q_a) = L(Q_a; q_{al}, q_{ah});$$

$$\chi_H(Q_a) = \Gamma(Q_a; q_{al}, q_{ah});$$

$$\chi_L(\mu_e) = L(\mu_e; \mu_{el}, \mu_{eh});$$

$$\chi_H(\mu_e) = \Gamma(\mu_e; \mu_{el}, \mu_{eh});$$

$$\chi_L(\mu_a) = L(\mu_a; \mu_{al}, \mu_{ah});$$

$$\chi_H(\mu_a) = \Gamma(\mu_a; \mu_{al}, \mu_{ah}).$$

The linguistic out of FCAC, admission decision, denoted as D , has three states which are “accept”, “reject” and “weak accept”. They are denoted as A, R, and WA. The membership functions are,

$$\chi_A(D) = \Gamma(D; d_a, d_{w1});$$

$$\chi_R(D) = L(D; d_{w0}, d_r);$$

$$\chi_{WA}(D) = \Pi(D; d_r, d_a, d_{w0}, d_{w1}).$$

Figure 3.11 shows the membership functions for Q_e , Q_a , μ_e , μ_a and D . There is a total of twelve parameters in these membership functions. They are

q_{el} - lower bound in the membership function of Q_e

q_{eh} - upper bound in the membership function of Q_e

q_{al} - lower bound in the membership function of Q_a

q_{ah} - upper bound in the membership function of Q_a

μ_{el} - lower bound in the membership function of μ_e

μ_{eh} - upper bound in the membership function of μ_e

μ_{al} - lower bound in the membership function of μ_a

μ_{ah} - upper bound in the membership function of μ_a

d_a - admission bound

d_r - rejection bound

d_{w0} and d_{w1} - weak admission bounds

The selection of values for these parameters is discussed in the later chapter.

Figure 3.11: FCAC membership functions

3.6.2 Fuzzy Rule Base

From the term sets definition, Q_e , Q_a , μ_e and μ_a have two states. In order to cover all the possible states of these four variables, the rule base needs $2 \times 2 \times 2 \times 2 = 16$ entries. As mentioned above, we classify admission into two types: QoS strong guaranteed service and QoS weak guaranteed service. These two types correspond to “accept” and “weak accept”

of $\chi(D)$.

The ideal conditions for the CAC to make an admission decision is:

1. The system currently should have enough bandwidth, indicated by a high μ_a , to accept the new connection.
2. The current customer satisfaction rate Q_a is high, which means most of the current services have been guaranteed.
3. The estimated customer satisfaction rate Q_e is high, which means if the CAC accepts this connection, the Q still meets our requirement.
4. The estimated unused bandwidth μ_e is relatively low. It means that the bandwidth will be used effectively if the connection request is accepted.

When more than one condition, for the ideal situation stated previously, is not satisfied, the decision would be either “reject” or “weak accept” .

The fuzzy Rule base is set of if ... then... statements. As shown in Table 3.2, H and L represent “high” and “low”; A, WA and R are “accept, weak accept” and “reject” respectively. Every entry of rule base table represents a statement of: if “ Q_e ...” and “ Q_a ...” and “ μ_e ...” and “ μ_a ...”, then “ D ...”. For instance, the rule 2 is explained as: IF Q_e is high AND Q_a is high AND μ_e is low AND μ_a is high THEN the CAC controller accepts the request. This rule is the case for ideal “accept” situation.

Rule 0, 3, 6, 10 have only one condition differing from ideal “accept” situation, we define the rule still as “accept” since 3 out of 4 satisfied conditions are good enough. Rule 1, 4, 7, 8, 11, 14 have two conditions differing from rule 2, the decision is defined as “weak accept”. For the rest rules, only one or none of the conditions is same as rule 2, therefore, the “reject” decision is made.

3.6.3 Rule Firing and Defuzzification

When the fuzzy inference engine receives the fuzzified input, it retrieves the fuzzy rule base. Based on the corresponding fuzzy rules, certain fuzzy decisions are fired. We apply the Mamdani implication [27] to represent the rules, which means,

$$\chi(D) = \min(\chi(Q_e), \chi(Q_a), \chi(\mu_e), \chi(\mu_a))$$

Example

An example to illustrate the rule firing procedure is as follows.

At time t_0 , a new connection request comes. The fuzzy CAC controller is working with the parameters as follows: $q_{el} = 0.5$, $q_{eh} = 0.9$, $q_{al} = 0.5$, $q_{ah} = 0.9$, $\mu_{el} = 0.2$, $\mu_{eu} = 0.7$, $\mu_{al} = 0.2$, $\mu_{au} = 0.7$, $d_r = 0.4$, $d_a = 0.6$, $d_{w0} = 0.3$, $d_{w1} = 0.7$.

From the estimation module, the estimated values yield: $Q_{e0} = 0.8$, $\mu_{e0} = 0.25$. From the real time measurement module, the current values are: $Q_{a0} = 0.92$, $\mu_{a0} = 0.77$. After fuzzification, we have $\chi_L(Q_{e0}) = 0.25$, $\chi_H(Q_{e0}) = 0.75$, $\chi_L(Q_{a0}) = 0$, $\chi_H(Q_{a0}) = 1$, $\chi_L(\mu_{e0}) = 0.1$, $\chi_H(\mu_{e0}) = 0.9$, $\chi_L(\mu_{a0}) = 0$, $\chi_H(\mu_{a0}) = 1$. Therefore, 4 rules are fired.

- Rule 0, Q_{e0} is grade 0.75 of H , Q_{a0} is grade 1 of H , μ_{e0} is grade 0.1 of H and μ_{a0} is grade 1 of H , then D is grade 0.75 of A .
- Rule 2, Q_{e0} is grade 0.75 of H , Q_{a0} is grade 1 of H , μ_{e0} is grade 0.9 of L and μ_{a0} is grade 1 of H , then D is grade 0.1 of A .
- Rule 8, Q_{e0} is grade 0.25 of L , Q_{a0} is grade 1 of H , μ_{e0} is grade 0.1 of H and μ_{a0} is grade 1 of H , then D is grade 0.1 of WA .

Figure 3.12: An example of fuzzy rule firing

- Rule 10, Q_{e0} is grade 0.25 of L , Q_{a0} is grade 1 of H , μ_{e0} is grade 0.9 of L and μ_{a0} is grade 1 of H , then D is grade 0.25 of $W A$.

We use the *Centre of Gravity* [27][26] method for defuzzification to derive the crisp output $\hat{D}$

$$\hat{D} = \frac{\sum_{k=1}^m cog(k) \cdot \chi_k(D)}{\sum_{k=1}^m \chi_k(D)} \quad (3.17)$$

where $cog(x)$ is the Centre of Gravity of x and m is the number of fired rules.

Following the example above, $\hat{D}$ is $\frac{0.75 \cdot 0.825 + 0.1 \cdot 0.825 + 0.1 \cdot 0.5 + 0.25 \cdot 0.5}{0.75 + 0.1 + 0.1 + 0.25}$, which has a value of 0.73. This value is always between 0 and 1. The value smaller than $\frac{d_{w0} + d_r}{2} = 0.35$ makes a “reject” decision. The value larger than $\frac{d_a + d_{w1}}{2} = 0.65$ gets “accept” decision. Therefore the crisp output is “accept”. See Figure 3.13.

Figure 3.13: Defuzzification using Centre of Gravity method

Table 3.1: Sample sources

Rule	Q_e	Q_a	μ_e	μ_a	D
0	H	H	H	H	A
1	H	H	H	L	WA
2	H	H	L	H	A
3	H	H	L	L	A
4	H	L	H	H	WA
5	H	L	H	L	R
6	H	L	L	H	A
7	H	L	L	L	WA
8	L	H	H	H	WA
9	L	H	H	L	R
10	L	H	L	H	A
11	L	H	L	L	WA
12	L	L	H	H	R
13	L	L	H	L	R
14	L	L	L	H	WA
15	L	L	L	L	R

Table 3.2: Fuzzy rule base

Chapter 4

Self Learning and Self Adaptation

In Chapter 3, a fuzzy logic CAC scheme is proposed. As mentioned previously, the estimation of Q and μ needs experimental curves (see Figure 3.4 and 3.5). These curves are generated in the beginning training phase. Since the CAC works in a real time environment, the initial learning may not be enough to truly reflect the relations between $Q - S$ and $\mu - S$. A continuing learning is essential for making accurate control.

On the other hand, the fuzzification for Q and μ is based on the membership functions. These functions have up to 12 parameters yet to decide. How to choose the proper values for them needs to be studied further.

Furthermore, with learning ability and adaptivity, the framework of CAC is updated on Figure 4.1.

4.1 Self Learning in Q and μ Estimation

Since there is no assumption for the traffic characterization, the system starts from knowing nothing about coming flows. The estimation curves are generated during the initial training

Figure 4.1: Adaptive fuzzy CAC framework

phase when the system starts to serve. Two reasons make the continuing learning necessary. One is that the CAC operates in real time system. Only a short period can be allowed in the initial training for estimation curves. In such a short time interval, the accumulation of sample data may be not enough to discover the distribution features of Q , μ and S . Second, the curves are more accurate if the types of traffic have more variety. To ensure that the traffic sample is large enough to cover as much applications as possible, the estimator has to keep learning and updating.

Figure 4.2 shows the learning cycle in the CAC. The controlled traffic goes through the

queuing and forwarding scheduler. From the scheduler, the real time measurement module continuously monitors the values of Q , μ and S currently in the system. Besides being transmitted to the fuzzy controller, this information is also passed to Q and μ estimator. The estimator updates its sample database, and re-calculates the estimation curves periodically.

Figure 4.2: Self-learning cycle

4.2 Self adaptation in fuzzy control module

In the fuzzy control module, the membership functions are applied in the fuzzification procedure to transfer crisp values into fuzzy grades of membership. As described in the previous chapter, there is a set of parameters deciding the “high” and “low” states of Q and μ . The assignment of numerical values to these parameters is arbitrary at the beginning.

One important task in the development of the FCAC is to find a set of proper parameters to achieve better performance. Letting the FCAC adapt itself according to the current situation is more advanced step. At the beginning of this section, we first study how these parameters influence the FCAC decision and system performance.

Shown in Figure 3.11, there are totally 12 parameters in membership functions. These parameters are:

q_{el} - lower bound in the membership function of Q_e

q_{eh} - upper bound in the membership function of Q_e

q_{al} - lower bound in the membership function of Q_a

q_{ah} - upper bound in the membership function of Q_a

μ_{el} - lower bound in the membership function of μ_e

μ_{eh} - upper bound in the membership function of μ_e

μ_{al} - lower bound in the membership function of μ_a

μ_{ah} - upper bound in the membership function of μ_a

d_a - admission bound

d_r - rejection bound

d_{w0} and d_{w1} - weak admission bounds

To make our experiments simple, we consider that, for weak admission bound, $d_{w0} = d_r$ and $d_{w1} = d_a$ which mean the lower weak accept is the same as the rejection bound and the upper weak accept bound is the same as the admission bound.

From the membership functions, we can divide the parameters into three groups:

1. q_{el} , q_{eh} , q_{al} and q_{ah} are parameters for turning the customer satisfaction rate Q_e and Q_a from the numerical inputs to grades of “high” and “low” states. The assigning of these parameters mainly affect membership grades of the customer satisfaction rate Q . Of course, the other measurement unused bandwidth μ is affected as a consequence.
2. μ_{el} , μ_{eh} , μ_{al} and μ_{ah} are parameters to fuzzify the unused bandwidth μ_e and μ_a . The change of these parameters affect the unused bandwidth μ . The customer satisfaction rate Q is affected indirectly.
3. d_a and d_r control the FCAC decision of “admission” and “rejection”. The change of these parameters will lead to stricter or looser decision.

Based on these three groups, we isolate the influence of each group by changing parameter values only in one group and fixing the values in other groups. Therefore, we can observe how the system performance is affected with each single groups of parameter. There are three scenarios designed for this purpose.

1. The effect of change of the lower and upper bounds of Q_e and Q_a .

We observe the effect to the system performance in terms of the customer satisfaction rate and number of connections in class 0 - “admission”, class 1 - “weak admission” and in total. The unused bandwidth are also shown as part of the results.

In this scenario, only the first group of parameters are changed. The others are unchanged. Three cases in this scenario are studied.

- Fix the upper bounds of Q_e and Q_a and shift the lower bounds only.

- Fix the lower bounds of Q_e and Q_a and shift the upper bounds only.
- Shift lower and upper bounds of Q_e and Q_a simultaneously. Fix the distance between lower and upper bounds.

2. The effect of change of the lower and upper bounds of μ_e and μ_a

We mainly observe the change of the unused bandwidth affected by the change of the bounds of μ_e and μ_a . The other parameters are unchanged. There are also three cases in this scenario.

- Fix the upper bounds of μ_e and μ_a . Shift the lower bounds only.
- Fix the lower bounds of μ_e and μ_a . Shift the upper bounds only.
- Shift lower and upper bounds of μ_e and μ_a simultaneously. Fix the distance between lower and upper bounds.

3. The effect of change of the bounds of “admission” and “rejection” d_a and d_r

The change of these bounds affect the decision of FCAC. We will fix the other parameters to see how the bounds of “admission” and “rejection” change the system decisions and performance. Two cases are considered:

- Shift the “rejection” bound only.
- Shift the “admission” bound only.

4.2.1 Scenario 1 -Evaluation of the Effect of the Lower bound and Upper Bound of Q_e and Q_a

All the simulations run in the Solaris platform. The code is written in C++ language, while the compile environment is GNU C++ compiler. All the simulations run by sole user loginning from console without other telnet connections from remote.

In this scenario, we focus on how the changing the lower and upper bound of Q_e and Q_a , which are q_{el} , q_{eh} , q_{al} and q_{ah} , can influence the customer satisfaction rate. We fix the other parameters with values shown in Table 4.1.

μ_{el}	μ_{eh}	μ_{al}	μ_{ah}	d_a	d_r
0.2	0.4	0.2	0.4	0.33	0.67

Table 4.1: Parameters assignment in Scenario 1

We implemented the CAC described in Figures 3.1 and 4.2 and simulate the three scenarios. The connection requests arrive in an exponential inter-arrival time distribution with the mean rate of 50 seconds. In the absence of real world data, we use connection lengths which are exponentially distributed. The average length is 2000 seconds. Each of them randomly selects a start point from traffic source data files. Like in the estimation simulations, every connection also comes with a QoS constraint. Here we define bit loss ratio (BLR) as the only measurement of QoS parameters, which is uniformly distributed from 5 to 9, corresponding to the BLR from 10^{-5} to 10^{-9} .

The FCAC classify the accepted connections into two classes. Class 0 type is QoS strongly guaranteed. It corresponds to the “accept” admission decision. They have high priority in scheduling and forwarding. The system tries to meet their requirement as much as possible. The other type of connection is class 1. Class 1 connections are admitted by the FCAC but only with QoS weak guaranteed. The connections in class 1 are served only

when there is no packet waiting in class 0 queues. This increases the probability of bit loss in this class of connections. The class 1 admission is defined as “weak accept” in FCAC term sets and rule base (See Section 3.6).

Case a: Changing the lower bounds of Q_e and Q_a only.

In this case, the upper bounds of Q_e and Q_a are set to 0.9. The lower bounds shift from 0.1 to 0.8 with the increment of 0.1 (See Figure 4.3).

Figure 4.3: Change of parameters in Scenario 1 - Case a

Figure 4.4 shows the simulation results. Each dot on the lines represents one simulation result. From the left diagram, when lower bounds q_{el} and q_{al} increase from 0.1 to 0.8, the customer satisfaction rate of class 1 Q_1 increases from 0.27 to 0.67. The overall customer satisfaction rate Q_{all} increases from 0.65 to 0.96. The unused bandwidth also increases from 0.25 to 0.43. The right diagram shows the connection numbers. The overall processed connection number N_{all} decreases from 90 to 62 and in class 1, N_1 drops from 43 to 4.

From the results, we can conclude, when the lower bounds increase and are close to the higher bounds, the connections are graded to smaller values as the membership of “high” estimated and measured customer satisfaction rate. Therefore, the FCAC tends to reject more connections. It especially affects the class 1 connections more, because the increase of the lower bounds make the membership grades drop sharply. More connections, which

are considered to be class 1, are rejected since the estimated and measured customer satisfaction rates are considered more as “low” states than as “high” when the lower bounds increase. Since more requests are rejected, especially in class 1, the overall customer satisfaction rate and unused bandwidth increases.

Figure 4.4: Scenario 1 - Case a: shifting the lower bound of Q_e and Q_a

Case b: Changing the upper bounds of Q_e and Q_a only.

In this case, the lower bounds of Q_e and Q_a are set to 0.1. The upper bounds shift from 0.2 to 0.9 with the increment of 0.1 (See Figure 4.5).

Figure 4.5: Change of parameters in Scenario 1 - Case b

Figure 4.6 shows the simulation results. Each dot on the lines represents one simulation

result. From the left diagram, when the upper bounds, q_{eh} and q_{ah} , increase from 0.2 to 0.9, the customer satisfaction rate of class 0 Q_0 increases from 0.92 to 1. In class 1, Q_1 also increases from 0.11 to 0.56. The unused bandwidth increases from 0.20 to 0.42. The right diagram shows the connection numbers. The processed connection number in class 0 N_0 decreases from 97 to 37 and in class 1, N_1 increases from 4 to 34.

Figure 4.6: Scenario 1 - Case b: shifting the upper bound of Q_e and Q_a

Case c: Changing the lower and upper bounds of Q_e and Q_a simultaneously.

In this case, the q_{el} and q_{al} , representing the lower bounds of Q_e and Q_a respectively, are from 0.1 to 0.55 with increment 0.05. We set that the intervals between the lower and upper bound of Q_e and Q_a are 0.4. Thus, the upper bounds q_{eh} and q_{ah} are from 0.5 to 0.95 with increment 0.05 along with the increase of their lower bounds. (See Figure 4.7).

Figure 4.8 shows the simulation results. The x-axis is q_{el} and q_{al} values. The y-axis in the left figure is the customer satisfaction rate Q and the unused bandwidth μ . There are three lines in the left figure: One is Q_0 which indicates the customer satisfaction rate of class 0 connections; one is the overall customer satisfaction rate, denoted as Q_{all} . The other

Figure 4.7: Change of parameters in Scenario 1 - Case c

is the system unused bandwidth μ . From the figure, we can find that when the bounds shift from (0.1, 0.5) to (0.55, 0.95), the overall customer satisfaction rate Q_{all} increases from 0.61 to 0.79. The Q_0 increases from 0.92 to 1. The unused bandwidth μ increases from 0.18 to 0.44. The right figure in Figure 4.8 gives the total processed connection number. In the figure, N_{all} drops from 110 to 71, , N_0 decreases from 99 to 36.

Figure 4.8: Scenario 1 - Case c: shifting the lower and upper bound of Q_e and Q_a Simultaneously

4.2.2 Scenario 2-Evaluation of the Effect of the Lower bound and Upper Bound of μ_e and μ_a

In this scenario, instead of changing the lower and upper bound of Q_e or Q_a , we move the bounds μ_e and μ_a . The parameter values that are changed here are: μ_{el} , μ_{eh} , μ_{al} and μ_{ah} . The parameters in other groups are kept unchanged. The values of fixed parameters are shown in table 4.2. In this scenario, same traffic sources as in scenario 1 are used. The following are the three cases that we study.

q_{el}	q_{eh}	q_{al}	q_{ah}	d_a	d_r
0.5	0.9	0.5	0.9	0.33	0.67

Table 4.2: Parameters assignment in Scenario 2

Case a: Changing the lower bounds of μ_e and μ_a only.

In this case, the upper bounds of μ_e and μ_a are set to 0.5. The lower bounds shift from 0.1 to 0.45 with the increment of 0.05. Figure 4.9 shows the simulation results. Each dot on the lines represents one simulation result. From the left diagram, when lower bounds μ_{el} and μ_{al} increase from 0.1 to 0.45, the unused bandwidth μ increases from 0.30 to 0.46. The overall customer satisfaction rate Q_{all} also increases from 0.73 to 0.85. From the right diagram, the overall processed connection number N_{all} decreases from 80 to 61. Correspondingly, N_0 increases from 27 to 47, and N_1 drops from 53 to 14.

Case b: Changing the upper bounds of μ_e and μ_a only.

In this case, the lower bounds of μ_e and μ_a are set to 0.1. The Upper bounds shift from 0.2 to 0.55 with the increment of 0.05. Figure 4.10 shows the simulation results. Each dot

Figure 4.9: Scenario 2- Case a: shifting the lower bounds of μ_e and μ_a

on the lines represents one simulation result. From the left diagram, when upper bounds μ_{eh} and μ_{ah} increase from 0.2 to 0.55, the unused bandwidth μ increases from 0.21 to 0.32. The overall customer satisfaction rate Q_{all} slightly increases from 0.74 to 0.83. From the right diagram, the overall processed connection number N_{all} decreases from 88 to 77. Correspondingly, N_0 decreases from 75 to 50, and N_1 increases from 13 to 27.

Figure 4.10: Scenario 2- Case b: shifting the upper bound of μ_e and μ_a

Case c: Changing the lower and upper bounds of μ_e and μ_a simultaneously.

In this case, the distance between lower and upper bounds of μ_e and μ_a are fixed to 0.2. The Upper bounds shift from 0.3 to 0.75 while the lower bounds move from 0.1 to 0.55. The increment is 0.05. Figure 4.11 shows the simulation results. Each dot on the lines represents one simulation result. From the left diagram, when lower and upper bounds μ_{eh} and μ_{ah} increase from (0.1, 0.3) to (0.45, 0.65), the unused bandwidth μ increases from 0.26 to 0.48. The overall customer satisfaction rate Q_{all} slightly increases from 0.84 to 0.89. From the right figure, the overall processed connection number N_{all} decreases from 83 to 62 and N_0 decreases from 59 to 45.

Figure 4.11: Scenario 2 - Case c: shifting the lower and upper bound of μ_e and μ_a simultaneously

4.2.3 Scenario 3 - Evaluation of the Effect of the “Admission” and “Rejection” bounds

In this scenario, the lower and upper bounds of Q_e , Q_a , μ_e and μ_a are fixed. We change the “admission” and “rejection” bounds to see the effect of these parameters. The unchanged parameters are shown in table 4.3. In this scenario, same traffic sources as in scenario 1 are used.

q_{el}	q_{eh}	q_{al}	q_{ah}	μ_{el}	μ_{eh}	μ_{al}	μ_{ah}
0.5	0.9	0.5	0.9	0.1	0.3	0.1	0.3

Table 4.3: Parameters assignment in Scenario 3

Case a: Changing the “rejection” bound.

In this case, the “admission” bounds is set to 0.7. The “rejection” bound shift from 0.1 to 0.55 with the increment of 0.05. Figure 4.12 shows the simulation results. Each dot on the lines represents one simulation result. From the left diagram, when “rejection” bound increases from 0.1 to 0.55, the overall customer satisfaction rate Q_{all} also increases from 0.59 to 0.84. The unused bandwidth μ increases from 0.36 to 0.43. From the right figure, the overall processed connection number N_{all} decreases from 110 to 81 and in class 1, N_1 drops from 62 to 33.

Case b: Changing the “admission” bound.

In this case, the “rejection” bound is set to 0.1. The “admission” bound is shifted from 0.3 to 0.75 with an increment of 0.05. Figure 4.13 shows the simulation results. Each dot on the lines represents one simulation result. From the left diagram, when “admission” bound

Figure 4.12: Scenario 3- Case a: shifting the “rejection” bound

increases from 0.3 to 0.75, the overall customer satisfaction rate Q_{all} also increases from 0.63 to 0.84. Q_0 increases from 0.93 to 1. The unused bandwidth μ increases from 0.21 to 0.29. From the right figure, the overall processed connection number N_{all} decreases from 106 to 97 and in class 0, N_0 drops from 96 to 45.

Figure 4.13: Scenario 3 - Case b: shifting the “admission” bound

4.2.4 Observations and Conclusion

From the above three scenarios, we have the following conclusions.

1. The customer satisfaction rate Q is mainly decided by the upper and lower bounds of Q_e and Q_a . More specifically, the area under the membership function of $\chi_H(Q_e)$ and $\chi_H(Q_a)$ decides the total number of connections accepted and the overall Q . This is because when these areas dwindle, more connection requests are considered having less grades of "high" Q memberships. Therefore, the FCAC considers that these requests do not satisfy the admission condition, and rejects them. This action increases the overall Q and at the same time decreases the total connection number.

2. The upper bounds of Q have more influence in the class 0 connections. and the lower bounds affect the total accepted connections within two classes.

3. The unused bandwidth is largely affected by changing the bounds of μ_e and μ_a . When the bounds increase, either lower bounds or upper bounds, or both of them, the unused bandwidth will increase. This is because the unused bandwidths, which are previously considered as "high", are not as much "high" as before when the bounds shift to high ends. Therefore, the FCAC starts to reject more connection requests to reserve the bandwidth for current connections.

4. The bound of "admission" control the class 0 admission, and "rejection" bound decides the both classes of connection. This is very straight forward because only when the FCAC output is over "admission" bound, a request can be accepted as class 0. Similarly, the "rejection" bound decides what output is considered to be enough for a request to be accepted in either class.

4.2.5 Self Adaptation

Based on the previous analysis, an adaptive fuzzy CAC can adjust the parameters according to the demands and system resource utilization. Some basic adaptive adjustment can be made as follows.

- Increase/decrease the “admission” bound to decrease/increase class 0 connections.
- Increase/decrease the “rejection” bound to accept less/more connection requests.
- Increase/decrease the upper bounds of Q_e and Q_a to improve/degrade the class 0 customer satisfaction rate.
- Increase/decrease the lower bounds or both lower and upper bounds of Q_e and Q_a to improve/degrade the overall customer satisfaction rate.
- Increase/decrease the lower or upper bounds of μ_e and μ_a to decrease/increase the system bandwidth utilization.

Since the customer satisfaction rate Q and unused bandwidth μ are always related with each other, and the class 0 and class 1 connections are also affect each other, more studies still need to be done in the future.

Chapter 5

Simulation and Comparison

5.1 General Simulation Environment

We implement our FCAC scheme in a single node case with fixed bandwidth capacity of 20 kbit/sec. The node is running under ideal GPS scheduling discipline. This enables the forwarding for each connection can be an infinitesimal amount of data. The weight assigned to each connection is dynamically calculated according to the buffer usage and QoS constraint for each connection (See Section 3.3). Each connection has an individual buffer. The buffer size is assigned as the value of 80 percent of the peak rate of this connection.

The source traffic are chosen from thirty sample traffic files. Some sample sources are shown in Figure 3.1. These sources represent different type of applications on ATM networks or the Internet. They have various statistical characterizations. The mean rates for these sources vary from 13.26 bit/s to 8801.4 bit/s. Some sources are very bursty such as the MPEG video data of movie “Star War”. It has peak rate of 200361 bit/s and the mean rate only 190.3 bit/s. There are also some traffic sources based on stochastic process, while some others are from the real traffic on the Internet [51].

In this thesis, we consider the bit loss ratio as the only parameter of QoS constraint, and only the bit loss due to the buffer overflow possibly happens in the simulations. There is no bit loss caused by other reasons. Thus, the QoS constraint can be expressed as a positive integer l which indicates the BLR is 10^{-l} . The higher l is, the less BLR is allowed.

All the simulations run in the Solaris platform. The code is written in C++ language, while the compile environment is GNU C++ compiler. All the simulations run by sole user loginning from console without other telnet connections from remote.

The simulations include three parts.

- The first part is to generate the to empirical estimation curves for estimating the customer satisfaction rate Q and unused bandwidth μ , which are used for the FCAC decision procedure. In this part of simulations, no admission control is committed and all connections are accepted. By varying the source types, connection numbers and QoS constraints, this part of simulations yield estimated values of Q and μ , which are not limited to certain type of traffic and can be more accurate in an unknown situation.
- The second part is to simulate FCAC in a real-time system. The connection requests arrive in a exponential distribution with various connection lengths and QoS constraints. The accepted connections are classified into two types: class 0 has high priority with strong QoS guarantee; class 1 has low priority with weak QoS guarantee. From this part, we evaluate the performance in terms of the customer satisfaction ratio Q and unused bandwidth μ , as well as the blocking rate, and the ratio of two classes.
- The Third part is the comparison. We compare the performance for the FCAC with

and without learning and adaptiveness. Both FCACs are also compared with an effective bandwidth CAC method - Mosquito algorithm [34][35], and peak rate CAC method [20][21]. To make them comparable, we implement a traffic generator to feed the same traffic to these CAC schemes in one set of simulations.

5.2 Estimation of Q and μ

This part of simulations is to emulate the training phase of the CAC. The training procedure is operated before the system is ready to accept connection requests. Since there is no admission control implemented in this phase, any type of sources are accepted by the system. For each simulation, every time a new source coming, we compute the source characteristic factor S , also derive the customer satisfaction rate Q and unused bandwidth μ (See Section 3.1). Then the distribution of $S \cdot Q$ and $\frac{\mu}{S}$ are generated. Through them, we get the linear functions of S to estimate the Q and μ . These estimation functions are used in the Q and μ estimator module 3.4.2. Figure 5.1 shows the procedure of this part of simulations.

The simulation duration is set to 2000 seconds. During this period, n number of sources are transmitted through the system simultaneously. These sources are chosen randomly from thirty sample traffic files and n is in a uniform distribution from 2 to 25. Each connection has a QoS constraint uniformly distributed from 5 to 9. That indicates the BLR is from 10^{-5} to 10^{-9} . Scheduling and forwarding obey GPS discipline which described in Section 3.3. For every simulation, an aggregated value of S is calculated. The loss ratio for each connection and the average system unused bandwidth are also recorded in order to get the values of Q and μ . The simulation is repeated 2000 times. Therefore, there are 2000 sample points distributed in the plane $S - Q$ and $S - \mu$, as we can see from Figure 3.2 and Figure 3.3.

Figure 5.1: Estimation of Q and μ

The distribution curves of $S \cdot Q$ and $\frac{\mu}{S}$ and their linear regression functions are plotted in Figure 5.2 and 5.3. Since the head and tail of the curves are not necessarily used in our computation, we only calculate the linear function to fit the stem of empirical curve where ϵ is between 0.05 and 0.95 in Equation 5.1.

$$\text{Prob}(Q \cdot S > a) = 1 - \epsilon \quad (0.05 < \epsilon < 0.95) \quad (5.1)$$

The fit line for Equation 5.1 is $\epsilon' = 1.2341a + 1.1808$ where $0.05 < \epsilon' < 0.95$ and ϵ' denotes the approximated value of ϵ (refer to Section 3.4.2 for the definition of ϵ). The sum of square errors is,

$$\sum_{i=1}^n (\epsilon'_i - \epsilon_i)^2 = 1.6680$$

In the simulations, we fix the ϵ' with the value 0.1, a is calculated as -0.88 . From

Figure 5.2: Approximation of the empirical distribution of $Q \cdot S$, using linear regression

Equation 3.11, the estimated value of Q is

$$Q_e > \frac{10^{-0.88}}{S} \quad (5.2)$$

In figure 5.3, the linear function to fit the empirical curve

$$\text{Prob} \left(\frac{\mu}{S} < a \right) = \epsilon \quad (0.05 \leq \epsilon \leq 0.95) \quad (5.3)$$

is $\epsilon' = 0.7910a - 2.6949$, where $0.05 \leq \epsilon' \leq 0.95$ and ϵ' denotes the approximated value of ϵ (refer to Section 3.4.2 for the definition of ϵ). The sum of square errors is,

$$\sum_{i=1}^n (\epsilon'_i - \epsilon_i)^2 = 2.4122$$

estimated value

$$\mu_e < 10^{4.54} \cdot S \quad (5.4)$$

Figure 5.3: Approximation of the empirical distribution of $\frac{\mu}{S}$, using linear regression

As the simulation results, the estimation functions of Q_e and μ_e yield and are applied to the FCAC simulations in the next section.

5.3 FCAC Evaluation

This part of the simulations is to evaluate the performance in a real time situation. Under the same circumstance listed in Section 5.1, The system forwards packets with GPS scheduling discipline. The connection requests arrive in an exponential inter-arrival time distribution with the mean rate of 50 seconds. In the absence of real world data, we use connection lengths which are exponentially distributed. The average length is 2000 seconds. Each of

them randomly selects a start point from traffic source data files. Like in the estimation simulations, every connection also comes with a QoS constraint. Here we define bit loss ratio (BLR) as the only measurement of QoS parameters, which is uniformly distributed from 5 to 9, corresponding to the BLR from 10^{-5} to 10^{-9} .

The FCAC classify the accepted connections into two classes. Class 0 type is QoS strongly guaranteed. It corresponds to the “accept” admission decision. They have high priority in scheduling and forwarding. The system tries to meet their requirement as much as possible. The other type of connection is class 1. Class 1 connections are admitted by the FCAC but only with QoS weak guaranteed. The connections in class 1 are served only when there is no packet waiting in class 0 queues. This increases the probability of bit loss in this class of connections. The class 1 admission is defined as “weak accept” in FCAC term sets and rule base (See Section 3.6).

FCAC procedure

As discussed in Section 3.2, the FCAC decision is based on the four input variables, which are estimated Q , actual Q , estimated μ and actual μ . Figure 5.4 shows the FCAC procedure with a 2000s simulation. There are three diagrams in this Figure. The bottom diagram gives the crisp values of Q_e , Q_a , μ_e and μ_a . Each point represents a measured or estimated value when a connection request arrives. These four variables are fuzzified to membership degrees of “high” or “low” states. FCAC compares these states with conditions in fuzzy rule base. Matched rules are fired. The fired rules generate a fuzzy decision, $\chi(D)$. This value is shown in the middle diagram. It is a value between 0 and 1. 1 represents the absolute admission and 0 is the absolute rejection. There are two bounds deciding the admission which are d_a and d_r . In Chapter 4, we have studied the effects of all of the

parameters in membership function functions. The values applied in this sample simulation is: $d_a = 0.67$, $d_r = 0.33$. This means the if $\chi(D)$ value in $[0.67, 1]$, the current connection request is accepted as class 0 type; the value in $[0.33, 0.67)$ makes FCAC accept this request as a class 1 type; the value less than 0.33 means a rejection. The top diagram in Figure 5.4 shows the FCAC decisions.

Figure 5.4: Fuzzy decision

Customer Satisfaction rate Q

Customer satisfaction rate Q is one of the two important metrics to measure the FCAC performance. The objective of the FCAC is to guarantee all the QoS in class 0 and to try the best for the service quality in class 1. Figure 5.5 shows the customer satisfaction rate Q in two classes and the average. In class 0, Q value is 1 constantly which means QoSs

are fully guaranteed. In Class 1 the satisfaction rate close to 78 percent. It is to say, three fourth of connections in class 1 meet the bit loss ratio threshold 10^{-9} to 10^{-5} . The average Q value is 85%.

Figure 5.5: Customer satisfaction rate Q

Bandwidth utilization

Unused bandwidth is the other objective for the FCAC. Figure 5.6 shows the mean unused bandwidth in a sample simulation. The average unused bandwidth is 19.5%, which means the bandwidth utilization is over 80% while the system guarantees 100% of class 0 services and 78% of services in class 1. The unused bandwidth distribution for 10-time simulations is shown in figure 5.7. We can see the major portion of unused bandwidth is between 0.1 and 0.3.

Figure 5.6: Unused bandwidth μ

The Active Queues in Progress

The Figure 5.8 shows the active queues in progress during simulation period. The average queue size in progress of class 0 is 6.3. The average queue size in class 1 is 14.1. The average of total queue number is 20.4. Around one third of admitted connections belong to class 0.

Admission Rate and Blocking Rate

Table 5.1 shows a group of simulations to examine the admission rate and blocking rate. N_{req} refers to the total number of request. N_{acpt} is the number of all admitted. N_0 or N_1 represent the number of admitted in class 0 and 1 respectively. The admission rate R_{acpt}

Figure 5.7: Distribution of unused bandwidth

for class 0 and 1 are get from $\frac{N_0}{N_{req}}$ and $\frac{N_1}{N_{req}}$. The blocking rate R_{blk} is $\frac{N_{req}-N_{acpt}}{N_{req}}$. From the table, around half of requests are accepted and more than one fourth of them are in class 0.

	1	2	3	4
N_{req}	1974	1926	1821	1935
N_{acpt}	1066	956	907	996
N_0	279	258	251	275
N_1	757	698	656	721
$\frac{N_0}{N_1}$	0.37	0.37	0.38	0.38
R_{acpt}	0.52	0.50	0.50	0.51
R_{acpt0}	0.14	0.13	0.14	0.14
R_{acpt1}	0.38	0.36	0.36	0.37
R_{blk}	0.48	0.50	0.50	0.49

Table 5.1: admission rate and blocking rate for FCAC

Figure 5.8: Number of queues in progress

5.4 Comparison

To compare the performance of the fuzzy CAC scheme with others, an effective bandwidth algorithm called Mosquito is used. Mosquito CAC algorithm, [34][35] is proposed by S. Crosby, *et al*, from U. of Cambridge, U.K. (See Section 2.2) It is a measurement based scheme using the approximation of effective bandwidth. Peak rate admission control is also used as a benchmark in performance evaluation.

We also compare the FCAC with and without self-learning and self-adaptive. The one without self-learning and self-adaptive is denoted as “simple FCAC”. The other is denoted as “adaptive FCAC”.

Figure 5.9 shows the number of concurrent flows in the system. We can observe the

peak rate allocation CAC scheme has an average 6.1 connections in progress during simulation period, while mosquito CAC has 19.6 connections. Simple fuzzy CAC has 17.8 and adaptive fuzzy CAC has 20.4. The adaptive fuzzy CAC has slightly higher connection number in progress.

Figure 5.9: Number of connection in progress

Multiplexing gain G is an important performance metric. It is defined as the ratio of maximum number of connections that can be admitted to the number of connections that can be admitted purely on peak-rate allocation basis. Obviously, for deterministic peak-rate allocation CAC scheme, the multiplexing gain $G = 1$. The multiplexing gain gives an indication of how better the statistical multiplexing is in the CAC scheme. Figure 5.10 shows the multiplexing gains of simple fuzzy, adaptive fuzzy and mosquito CAC schemes. The G values of simple fuzzy CAC is 2.62, adaptive CAC is 2.91 and mosquito CAC is

2.85.

Figure 5.10: Multiplexing gain

From the queue size in progress and multiplexing gain, the adaptive fuzzy CAC has better performance than simple fuzzy CAC. Both of them accept more connections than peak rate CAC scheme does. The queue size in progress in adaptive fuzzy CAC scheme is close to mosquito CAC scheme, while simple fuzzy CAC has slightly less queue numbers. But if we compare the bandwidth utilization, we will find that both fuzzy CAC scheme have better bandwidth efficiency than mosquito CAC algorithm. The comparison of bandwidth utilization in Figure 5.11.

From the figure, peak rate allocation has a 33.4% utilization, mosquito CAC has 69.1%, while simple fuzzy CAC and adaptive fuzzy CAC have 76.3% and 80.8% respectively. Fuzzy CACs have better bandwidth utilization than mosquito effective bandwidth algo-

Figure 5.11: Bandwidth utilization comparison

rithm.

We also found that in terms of the bandwidth utilization, both fuzzy schemes converge more slowly than mosquito method and peak rate allocation scheme. This is because that the fuzzy schemes choose sources more “carefully”. For the more bursty sources, FCAC rather choose to reject them even though the bandwidth is still enough. FCAC reverse the bandwidth for “good” sources which are more regular.

Table 5.2 shows the comparison of admission rate and blocking rate in different CAC schemes. From the same traffic pattern, peak rate scheme blocked 81% of connection requests. Mosquito effective bandwidth algorithm and adaptive fuzzy CAC blocked half of requests, slightly higher than simple fuzzy CAC scheme.

Since all the simulations run in the same platform which is SunOS 5.8. During the simulations, there’s single user loginning from the console without any remote connection

	Total request	Class 0	Class 1	Acpt rate	Rejected	Blocking rate
Peak	1974	375	-	0.19	1599	0.81
mosquito	1974	988	-	0.50	986	0.50
simple fuzzy	1974	228	682	0.46	1054	0.54
adaptive fuzzy	1974	279	757	0.52	938	0.48

Table 5.2: Connection request blocking rate

sessions on the workstation. We compare the processing time for these four schemes. The results show that the peak rate allocation has the fastest processing time, while the mosquito is the slowest. We conclude the FCAC schemes have moderate computation needs than mosquito EB scheme does.

From the comparison above, we can find simple fuzzy, adaptive fuzzy and mosquito algorithm all have high multiplexing gain. They have higher queue size in progress and less blocking rate than peak rate scheme. In terms of bandwidth utilization, both simple and adaptive fuzzy CAC have better performance than mosquito algorithm, although the blocking rates are closer. The results show that fuzzy CAC scheme is able to select less bursty connections which increase the resource efficiency. Compare to the implementation complexity, fuzzy CAC need less CPU processing time than mosquito CAC, while, of course, the peak rate admission scheme is the fastest. Table 5.3 shows the comparison results.

	peak	mosquito	simple fuzzy	adaptive fuzzy
QoS guarantee	very good	good	good	good
Differentiated service	no	no	yes	yes
Connections in progress	low	high	medium	high
Blocking rate	high	low	medium	low
Bandwidth utilization	low	medium	high	high
Processing time	very fast	slow	fast	fast

Table 5.3: Performance comparison

Chapter 6

Conclusion

6.1 Summary of the Work

The current trend of integrating all type of services into uniform multi-service networks requires on the provision of Quality of Service. To guarantee the QoS for each application, traffic management is studied extensively. As one of the most important aspects in traffic management, connection admission control is the first step to control traffic. The basic function of a CAC is to decide whether or not to accept a new connection request without deteriorating service qualities for the connections already existing. Furthermore, CACs are also used to maximize the resource utilization and to avoid waste of bandwidth.

Many CAC schemes are proposed recently. Some are very simple in terms of the theory and implementation. For instance, the peak rate CAC scheme compares the sum of the peak rates of current connections and the coming request with the total system bandwidth. If the system bandwidth is more than the sum of peak rates, then the request is admitted, otherwise, it will be rejected. But such simple schemes often lead to low utilization of system bandwidth, especially given irregular bursty applications on the Internet. Therefore,

other more complicated schemes are proposed, such as the effective bandwidth method and the artificial intelligence method.

In general, two challenges for CAC design must be faced. The first is that current applications on packet networks are unpredictable. There are always new type of applications appearing, and even for the existing ones, the fully description for some applications with certain traffic model is yet an open issue. Therefore, the dependency of certain traffic models for a CAC should be limited to the minimum. The second challenge is the implementation complexity and process time. Since CACs have to make admission decisions in a real-time basis, fast decisions are needed. The computation cannot be too complex.

The effective bandwidth CAC method has been well studied in recent years. It has simple principle to make admission decision similar to circuit switched CAC schemes. But most effective bandwidth CAC schemes are based on full understanding of traffic characterization. Even if some measurement based schemes collect information from real time measurement, the computation complexity obstructs its practical implementation.

To overcome such challenges, in this thesis we proposed a novel self-adaptive fuzzy connection admission control scheme. This scheme is based on fuzzy logic [26][27][28][30], a promising technology which has been extensively used in automation control. The character of graded membership in fuzzy logic gives more accurate control than traditional non-linear control. The major advantages in our proposed FCAC scheme are as follows.

- It does not rely on any traffic models or statistical characterization. Most information for traffic sources can be obtained through simple online measurement. There is no assumption for traffic types.
- It is simple and fast. Fuzzy logic is proven to have modest computation complexity and is easy to implement.

- It classifies service type. The proposed FCAC introduces differentiated classes during admission. Different classes get different services, which also maximizes the system utilization.
- It has self learning and self adaptive ability. The FCAC keeps learning from current transmission and adapts parameters to achieve improved performance.

We introduce two metrics to measure the CAC performance: one is the customer satisfaction rate, denoted as Q . It is the proportion of connections with guaranteed QoS among all connections processed. The other is the unused bandwidth μ , which represents the system utilization. From the observation, we find that these two metrics are related to the traffic burstiness and QoS constraints. To define the characters that affect Q and μ , notation S is introduced. We further investigate these relations and derive linear empirical functions to estimate Q and μ from S .

The fuzzy CAC, therefore, is based on these estimations, as well as the real time measurement, to determine whether or not to admit a new connection request. There are four input variables which are estimated Q , estimated μ , actual Q and actual μ . We define membership functions to convert these values into certain states “high” or “low” with grade from 0 to 1. We also construct a rule base which defines what decision should be made according to the states of these four variables. By combining fired rules together, a decision is finally made to “accept” or “reject” the request. Since we applied two class admission types, a “weak accept” decision is given to the admitted connection with less priority of QoS guarantee.

Assigning appropriate numerical values to the parameters is a non-trivial task. It is critical since fuzzy logic provides great flexibility in membership definition and rule description. Poor choice of parameters can deteriorate system performance.

In the evaluation and comparison of the FCAC, we choose the peak rate CAC scheme [20] as a benchmark. The Mosquito algorithm [34] [35], which is a measurement based effective bandwidth scheme, is used in the performance comparison. From the simulations, we have made the following observations.

- The proposed FCAC achieves our two primary goal: the guarantee of QoS and bandwidth utilization.
- The adaptive FCAC has better performance than the FCAC without self-learning and self-adaptiveness.
- Compared to Mosquito effective bandwidth method, the FCAC has the similar performance in terms of the connection number in progress, blocking rate and multiple gain. But the FCAC has better bandwidth utilization than Mosquito effective bandwidth method.
- FCAC is easy to implement and uses less CPU time.

Overall, the proposed FCAC is a fast and simple CAC scheme especially efficient in the unpredictable bursty environment. It does not rely on any traffic model and statistical characterization. It provides differentiated service which is not available in most CAC schemes. The FCAC achieves better bandwidth utilization than some effective bandwidth methods. In the matter of service satisfaction rate, the FCAC “hard” guarantees the QoS for high priority class and satisfies majority of the overall QoS constraints.

6.2 Future Work

The proposed FCAC is far from a perfect connection admission control method. In this thesis, we give a fundamental infrastructure under which some future work will be required, such as the following:

- In Chapter 4, we compared the effect of different sets of parameters. In this thesis, the adaptiveness is based on the observation from the comparison simulations. The theoretical analysis of the effect of parameters is an important issue for further study.
- The membership functions, fuzzy rules and implication method that we have chosen may not be the best in a CAC. In this thesis, we applied them in a basic form. Using other forms of membership functions may improve the service quality and system utilization.
- Neural network is another important AI method in automation control. Fuzzy logic is powerful at adaptiveness while neural network is very good at learning from the unknown. One of the next improvements for the FCAC is to introduce a neural network into the self-learning procedure. Other learning methods such as reinforcement learning [53] may also be investigated.

Bibliography

- [1] ITU-T I.371, Traffic Control and Congestion Control in B-ISDN, November, 2000
- [2] ITU-T I.356, B-ISDN ATM Layer Cell Transfer Performance, March, 2000
- [3] ATM Forum, Traffic Management 4.1, March, 1999
- [4] K. Nichols, S. Blake, F. Baker, D. Black, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, RFC2474, Dec. 1998.
- [5] S. Black et al., An Architecture for Differentiated Services, RFC2475, Dec. 1998.
- [6] K. Nichols, V. Jacobson, L. Zhang, A Two-bit Differentiated Services Architecture for the Internet, RFC2638, July 1999.
- [7] B. Braden et al., Recommendations on Queue Management and Congestion Avoidance in the Internet, RFC 2309, April 1998.
- [8] R. Braden, D. Clark, S. Shenker, Integrated Services in the Internet Architecture: an Overview, RFC 1633, June 1994.
- [9] J. Wroclawski, Specification of the Controlled-Load Network Element Service, RFC 2211, Sept. 1997.

- [10] S. Shenker, C. Partridge, R. Guerin, Specification of Guaranteed Quality of Service, RFC 2212, Sept. 1997.
- [11] A. Mankin, B. Braden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, L. Zhang, Resource ReServation Protocol (RSVP) Version 1 Applicability Statement Some Guidelines on Deployment, RFC 2208, Sept. 1997.
- [12] S. Brim, B. Carpenter, F. Le Faucheur, Per Hop Behavior Identification Codes, RFC 2836, May 2000
- [13] Z.L. Zhang, Z. Liu, J. Kurose and D. Towsley, Call Admission Control Schemes under the Generalized Processor Sharing Scheduling Discipline, Telecommunications Systems, Vol. 7, No. 1-3, pp. 125-152, July 1997
- [14] Z. Wang, Internet QoS - Architecture and Mechanisms for Quality of Service, Morgan Kaufmann, 2001
- [15] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-node Case. IEEE/ACM Transactions on Networking, Vol. 1, No. 3, pp. 344-357, June 1993
- [16] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case. IEEE/ACM Transactions on Networking, Vol. 2, No.2, pp. 137-150, April 1994
- [17] A. Demers, S. Keshav and S. Shenker, Analysis and Simulation of a Fair Queuing Algorithm, Journal of Internetworking: Research and Experience, pp. 3-26, 1990. Also in Proceedings of ACM SIGCOMM '89, pp. 3-12.

- [18] L. Kleinrock, *Queuing System Vol. 2: Computer Applications*. New York, Wiley, 1976
- [19] L. Zhang, Virtual Clock: A New Traffic Control Algorithm for Packet Switched Networks, in *ACM Transactions on Computer Systems*, 9(2), May 1991, pp. 101-125.
- [20] H. G. Perros, K. M. Elsayed, Call Admission Control Schemes: A Review, *IEEE Communication. Mag.* 34 (11) (1996) 82–91.
- [21] M. A. Rahin, Mourad Kara, Call Admission Control Algorithms in ATM Networks: A Performance Comparison and Research Directions, Early Draft, ATM-MM Research Group, September 1998.
- [22] Esko Turnen, *Mathematics Behind Fuzzy Logic*, Physical-Verlag Heidelberg, 1999
- [23] Hung T. Nguyen and Elbert A. Walker, *A First Course in Fuzzy Logic (Second Edition)*, Chapman & Hall/CRC, 2000
- [24] Daniel McNeil and Paul Freiburger, *Fuzzy Logic*, Simon & Schuster, 1993
- [25] S. Wang, D. Xuan, R. Bettati and W. Zhao, Providing Absolute Differentiated Services for Real-Time Applications in Static-Priority Scheduling Networks, *INFOCOM*, 2001
- [26] D. Dubois, and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Orlando, FL, Academic Press, 1980
- [27] D. Driankov, H. Hellendoorn and M. Reinfrank, *An Introduction to Fuzzy Control*, Springer-Verlag, 1993

- [28] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*, John Wiley & Sons, Inc. 1994
- [29] K. Shimoto, N. Yamanka, and T. Takahashi, *Overview of Measurement-Based Connection Admission Control Methods in ATM Networks*, IEEE Communications, 1999
- [30] L. A. Zadeh, *Fuzzy sets, Information and Control*, 1965
- [31] B. Schweizer and A. Sklar, *Associative Functions and Statistical Triangle Inequalities*, Publicationes Mathematicae Debrecen, 1961
- [32] S. Weber, *A General Concept of Fuzzy Connectives, Negations and Implications Based on t-norms and t-co-norms*, Fuzzy Sets and Systems, 1983
- [33] L.A. Zadeh, *A Theory of Approximate Reasoning*, Machine Intelligence, Halstead Press, NewYork, 1979
- [34] S. Crosby, I. Leslie, B. McGurk, J. Lewis, R. Russell and F. Toomey, *Statistical Properties of a Near-Optimal Measurement-Based CAC algorithm*, In Proceeding of IEEE ATM'97, June 1997
- [35] J. Lewis, R. Russell, F. Toomey, B. McGurk, S. Crosby and I. Leslie, *Practical Connection Admission Control for ATM Networks Based on On-line Measurements*, Proceedings IEEE ATM '97, June 1997, Lisbon
- [36] I. Hsu and J. Walrand, *Dynamic Bandwidth Allocation for ATM Switches*, February 1995

- [37] P. Barham, S. Crosby, T. Granger, N. Stratford, M. Huggard and F. Toomey, Measurement Based Resource Allocation for Multimedia Applications, Multimedia Computing and Networking (MMCN), 1998
- [38] R.J. McEliece, J. Murphy, M. Jennings and Z. Yu, On Simplified Modelling of the Leaky Bucket, Proceeding of the IEE 13th UK IEE Teletraffic Symposium, Strathclyde, UK, 18-20 March 1996.
- [39] J. Y. Hui, Resource Allocation for Broadband Networks. IEEE Journal on Selected Areas in Communications, Vol. 6, No.9, pp.1598-1608, December 1988
- [40] F. Kelly, Stochastic Networks: Theory and Applications, pp.141-168, Clarendon Press, Oxford, 1996
- [41] R. Srinivasan and J. Talim, Fuzzy Connection Admission Control for Self-managed Networks, in Proceedings of IEEE Canadian Conference on ECE, 2002
- [42] L. Barolli, A. Koyama, T. Yamada, S. Yokoyama, T. Suganuma and N. Shiratori, A Fuzzy Admission Control Scheme for High-Speed Networks, IEEE, 2001
- [43] L. Peterson and B. S. Davie, Computer Networks: A Systems Approach, San Francisco, Calif. Morgan Kaufmann Publishers, c2000.
- [44] R. C. James, Mathematics dictionary, New York: Van Nostrand Reinhold, c1992.
- [45] M. A. Marsan, A. Bianco, C. Casetti, P. Castelli, R. Lo Cigno, M. Mellia, and M. Munafo, A CAC Algorithm for ATM Networks Supporting Different QoS Classes, Performance Modelling and Evaluation of ATM Networks, Vol.4, Chapman and Hall, London, 1998.

- [46] D. Nandita, J. Kuri, and H.S. Jamadagni, Optimal Call Admission Control in Generalized Processor Sharing Schedulers, Tech. Rep. TR-00-02, Center for Electronics Design and Technology, Indian Institute of Science, Jun. 2000.
- [47] D. Xuan, C. Li, R. Bettati, J. Chen and W. Zhao, Utilization-Based Admission Control for Real-Time Applications, The IEEE International Conference on Parallel Processing, Canada, Aug. 2000.
- [48] G. A.F. Seber and A. J. Lee. Linear Regression Analysis, Hoboken, N.J., Wiley-Interscience, c2003.
- [49] A. Pitsillides, A. Sekercioglu, Fuzzy Logic Based Congestion Control, COST 257: Impacts of new services on the architecture and network performance of broadband networks, Sept. 1999.
- [50] Y. A. Phillis and R. Zhang, Fuzzy service rate control of queuing systems', IEEE Trans. Systems, Man and Cybernetics, Part B, Aug. 1999.
- [51] <http://pma.nlanr.net/PMA>
- [52] E. Rosen, *et al*, Multiprotocol Label Switching Architecture, RFC 3031, Jan. 2001
- [53] D. E. Moriarty, A. C. Schultz and J. J. Grefenstette, Evolutionary Algorithms for Reinforcement Learning, Journal of AI Research, Sept. 1999