INTELLIGENT ADAPTIVE BANDWIDTH PROVISIONING FOR QUALITY OF SERVICE IN UMTS CORE NETWORKS

TIMOTHY HUI CHEE KIN

(B. Eng. (Hons.), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2003

ACKNOWLEDGMENTS

I would like to take this opportunity to thank the many who have been alongside me, supporting me in various ways, and who have contributed in one way or another to the production and success of this thesis. Of these people, the following deserve special mention.

Our Father in Heaven for providing His ever-presence, for His never-ending source of strength, and for His over-shadowing guidance. He has indeed been my source of inspiration and is the sole purpose which I owe my existence and dedication to.

Dr Tham Chen Khong for his supervision, his insights into the area of research, and his trust in my work.

Liu Yong for his partnership in the same area of research, superior wisdom and great analytical mind.

My parents for their love and support and for their belief in letting their children follow their own dreams.

Last but not least, *Elise, my beloved girlfriend,* for her unending support, love and prayers, for her companionship that has always so warmed my heart.

To

My Family

And

GOD

TABLE OF CONTENTS

CHAPTER 3: Bandwidth Provisioning

CHAPTER 4: Reinforcement Learning-based Provisioning

CHAPTER 5: Reinforcement Learning-based Provisioning for Core Backbone Network

SUMMARY

The issue of bandwidth provisioning is imperative for differentiated quality of service (QoS) to be achieved in UMTS core networks. As UMTS is to offer various classes of services that require different QoS levels, careful bandwidth provisioning is needed to ensure that the QoS of every class is met in the converged UMTS core network. The Differentiated Services model has been chosen as the service model for implementing UMTS networks. The UMTS service classes can be mapped onto various DiffServ classes. By adaptively controlling the bandwidth allocated to each DiffServ class, service providers are able to quantitatively control the level of QoS provisioned. This is crucial since each class of service offered would be governed by service level agreements contracted between service providers and mobile subscribers that spell out exact QoS assurance in terms of throughput, latency and packet loss bounds.

The UMTS core network is divided into two portions – the UMTS core backbone network and the UMTS terrestrial radio access network (UTRAN), which will be provisioned using different schemes. This is because the UTRAN is topologically different from the UMTS core backbone network. The traffic in the UTRAN is also more dynamic; since in a mobile access network traffic is less aggregated and handoff traffic can cause large changes in overall traffic patterns. In this work, a bandwidth provisioning solution is presented that is bandwidth efficient, scalable, easily implemented and able to provision bandwidth in an objective manner. To meet the first criteria, the weighted fair queuing method is used to provision bandwidth as it offers high bandwidth utilization. The DiffServ framework that is used allows the scheme to be scalable. The algorithms used in the scheme can be implemented in bandwidth managers such as a DiffServ bandwidth broker. In order to provision bandwidth in a manner that requires no complex control mechanisms and little expert knowledge, and yet meet the service requirements contracted in SLAs, a reinforcement learning (RL) method is used. The advantage of an RL method is that RL agents are able to adaptively learn policies that map measured traffic conditions to WFQ weight settings through reward and penalty feedback. By designing the reward and penalty feedback based on the pricing of services and the SLA, the RL-based scheme, which is presented in this work is capable of intelligently provisioning bandwidth.

Two bandwidth provisioning schemes are presented for UMTS core backbone networks. The Reinforcement Learning Adaptive Provisioning (RLAP) scheme aims to maximize revenue for the service provider based on a novel multi-tier pricing plan that is designed to maximize utilization and manage subscriber satisfaction. Alternatively, the Reinforcement Learning Dynamic Provisioning (RLDP) scheme provisions bandwidth such that QoS assurance levels are strictly met. Since most of today's SLAs contract assured levels of QoS rather than strict 100% guarantees, service providers can use this leeway to improve utilization and at the same time adaptively manage QoS. But since high penalties in monetary terms as well as reputation are at stake, the bandwidth provisioning must be intelligent enough to manage the different classes of traffic in a heterogeneous network.

Provisioning bandwidth in the UTRAN is different from the UMTS core backbone network, since hand-off traffic is an issue. The RLDP scheme is modified by considering neighboring traffic as well. With the modification, the resulting Reinforcement Learning Bandwidth Provisioning (RLBP) scheme thus manages to meet the QoS assured levels even under high hand-off situations. Simulation studies on all three schemes show that the solutions presented can meet QoS requirements efficiently.

LIST OF ILLUSTRATIONS

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

LIST OF TABLES

CHAPTER 2

CHAPTER 5

CHAPTER 6

LIST OF ABBREVIATIONS

ATM	Asynchronous Transfer Mode
AF	Assured Forwarding
BB	Bandwidth Broker
BE	Best Effort
BSC	Base Station Controller
CBR	Constant Bit Rate
EF	Expedited Forwarding
GA	Genetic Algorithm
GGSN	Gateway GPRS Support Node
FTP	File Transfer Protocol
IEEE	Institution of Electronic and Electrical Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
LAN	Local Area Network
MLP	Multi-Layer Perceptron
MPLS	Multi-Protocol Label Switching
NDP	Neuro-Dynamic Programming
NS	Network Simulator
PDA	Personal Digital Assistant
PHB	Per-Hop Behavior
QoS	Quality of Service
RAN	Radio Access Network
RED	Random Early Detection

RL	Reinforcement Learning
RNC	Radio Network Controller
RNS	Radio Network Subsystem
RSVP	Resource ReSerVation Protocol
SGSN	Serving GPRS Support Node
SLA	Service Level Agreement
SLS	Service Level Specifications
SRV	Stochastic Real-Valued
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications Service
UTRAN	UMTS Terrestrial Radio Access Network
VoIP	Voice over IP
WAN	Wide Area Network
WFQ	Weighted Fair Queuing

LIST OF PUBLICATIONS RELATED TO THIS THESIS

1. Timothy Chee-Kin Hui and Chen-Khong Tham, *"Adaptive Provisioning of Differentiated Services Networks based on Reinforcement Learning"*, IEEE Transactions on Systems, Man and Cybernetics, Special Issue, Autumn 2003.

2. Timothy Chee-Kin Hui and Chen-Khong Tham, *"Reinforcement Learning-based Dynamic Bandwidth Provisioning for Quality of Service in Differentiated Services Networks"*, Proceedings of IEEE International Conference on Networks 2003 (ICON 2003), 29 Sep – 1 Oct 2003, Australia.

CHAPTER 1

AIMS AND OBJECTIVES OF THIS RESEARCH

1.1 INTRODUCTION

UMTS (Universal Mobile Telecommunications Service) [1] is a defined standard by the 3rd Generation Partnership Project (3GPP) [2] group. It covers all the necessary details for the implementation of 3rd Generation (3G) mobile networks. Under the UMTS standard, 3GPP Release 1999, Release 4 and Release 5 [6-8] map the evolution of the UMTS core network from a circuit-switched based architecture to an all-IP packet-switched based architecture.

With an all-IP network, a variety of services with varying Quality of Service (QoS) requirements could be transported over the same core network [16]. To ensure that the QoS requirements of all classes can be met efficiently, the proportion of bandwidth provisioned to each class has to be optimized. Though there have been solutions to the bandwidth provisioning issue, such solutions are tailored for fixed networks and are usually designed for Asynchronous Transfer Mode (ATM) backbone networks [50-55]. With the convergence of mobile and fixed networks (backbone networks being shared by both mobile network and fixed network operators), the bandwidth provisioning issue is made more complex by the continuously varying patterns of traffic and the constant altering of routes due to the mobility of end nodes.

This research focuses on a solution to the bandwidth provisioning problem for the UMTS network. The UMTS network has been split into two portions – the core backbone network (which is expected to be converged with the fixed backbone network) and the radio access network as the characteristics of both portions are quite different. The thesis presents a solution to each of the portions.

1.2 SCOPE OF THE THESIS

The scope of the thesis covers the following areas:

(i) The use of adaptive weighted-fair bandwidth proportions as a means to achieve QoS within the UMTS core network and the resulting formulation of the bandwidth provisioning optimization problem.

(ii) The development of a self-tuning algorithm based on Reinforcement Learning (RL) that adaptively converges towards the solution to the formulated bandwidth provisioning problem.

(iii) The implementation of the RL algorithm to the core backbone network.

(iv) The implementation of the RL algorithm to the radio access network.

1.2.1 Adaptive Bandwidth Provisioning for QoS in UMTS Core Network

Proper handling of Quality of Service is required for the UMTS core network to handle multiple services ranging from Voice over IP (VoIP) [13-16] to multimedia applications to e-commerce transactions. There have been proposals to map various QoS classes from other standards to the UMTS QoS classes [11-15]. Various QoS management solutions [23-25] have also been presented to enable QoS over UMTS networks.

Work has also been done at the lower layers [26-28], but these have been mainly focused on the wireless portion of the network for obvious good reasons. Since the wireless bandwidth is limited, scarce resources have to be efficiently allocated so that QoS can be achieved over the air. For some reason, network-layer QoS has not received much attention. This could be due to the assumption that core network bandwidth is much larger than the wireless bandwidth. This might be the case now, but as wireless devices become ubiquitous and wireless services start sprouting, the core network would have to support the traffic from hundreds of base stations carrying thousands of sessions. As such, the efficient provisioning of bandwidth in the UMTS core network would be much needed in the near future.

Due to the dynamic nature of mobile traffic, bandwidth cannot be statically provisioned. Over-provisioning is a method commonly used today. By allocating more than enough bandwidth to meet the heaviest of traffic, QoS can be ensured. However, this is an inefficient method as there would be low bandwidth utilization due to large variations in traffic. Another commonly used method is to give priority to traffic requiring strict QoS. This is at best a temporary solution, as it only provides a two-level differentiation of service. With more varied applications being developed, varying degrees of differentiation would be required. A priority-based provisioning solution would not be insufficient to maintain quantitative QoS. Therefore, a form of adaptive bandwidth provisioning would be needed. The adaptive bandwidth provisioning optimization problem is formulated, but it is shown that the problem is infeasible to solve optimally and an approximate method has to be used.

1.2.2 Reinforcement Learning-based Solution to the Bandwidth Provisioning Optimization Problem

Reinforcement Learning (RL) [60,61] is a machine learning theory that can be used in control problems such as the bandwidth provisioning optimization problem; where the amount of bandwidth to provision for each aggregate class of traffic has to be adaptively controlled. In RL, a learning agent has to formulate a policy, which determines the appropriate action to take in each state in order to maximize the expected cumulative reward over time. The reward is derived from how favorable the outcome is of the action taken by the agent in a particular state. RL thus provides a way to relate state, action and penalty.

When RL is applied to the bandwidth provisioning optimization problem, traffic conditions (state) can be related to provisioning settings (action) and adjusted through the use of QoS feedback (penalty). Through a gradient-based algorithm, the policy (solution) is adaptively learned such that QoS penalties are minimized and revenue is maximized. Since the provisioning problem has a continuous state and action space, an appropriate continuous state-action space RL method [76] was used.

The application of continuous state-action space RL to network management is a novel work. There have been no previous applications published explicitly that employ a similar technique.

1.2.3 Adaptive Bandwidth Provisioning for Core Backbone Network

The UMTS core backbone network [8], as defined in this thesis, is the portion of the core network that includes the SGSN (Serving GPRS Support Node) at one edge and the GGSN (Gateway GPRS Support Node) at the other. The core backbone network functions as a packet network that connects the UMTS Terrestrial Radio Access Networks (UTRAN) around the provider’s area of coverage to the fixed core network (or carrier-network). The SGSN serves the various UTRANs by connecting to the Radio Network Controllers (RNC) within each UTRAN. The GGSN provides the access to the other providers’ networks by connecting to the carrier-network. The topology of the UMTS core backbone network is usually one that is geographicallydetermined. As such, the topology can be modeled as a network of edge and core nodes, similar to a Differentiated Services (DiffServ) network [9].

The solution presented is one that provisions bandwidth at each node based on the amount of traffic measured on each outgoing link. The provisioning policy is adapted based on the service-level agreement (SLA) [29] contracted between the provider and the subscribers. A DiffServ model is used as it is the recommended service model for the UMTS core network.

1.2.4 Adaptive Bandwidth Provisioning for Radio Access Network

The UMTS radio access network [6], as defined in this thesis, is the portion of the core network that includes the RNC (Radio Network Controller) at one edge and the base station (Node B) at the other. The radio access network is commonly known as the UMTS Terrestrial Radio Access Networks (UTRAN) [21] and it provides radio coverage over the provider's entire network. As such, the topology of the UTRAN is usually pyramid or hierarchical. The base stations provide the wireless connection to the user devices (UE) and are controlled and attached to the RNCs.

As UE users move from one place to another, handovers from one base station to another may occur. This may occur frequently in densely populated areas or on high-speed transport routes. Handovers require a change of route and hence fresh provisioning. This causes great fluctuations in traffic patterns in the UTRAN. Currently, mobile providers handle this situation by reserving channels (bandwidth) in advance before handover. However, this causes low bandwidth utilization when many UEs are attached to each base station and mobility rate is high.

To solve this problem, bandwidth provisioning is proposed as an alternative to bandwidth reservation. The difference is that bandwidth is shared within a class and reservations do not have to be made for each UE. When handovers occur, the new UEs entering the base station's coverage share the bandwidth with the other present UEs in the same class. To ensure QoS is maintained, the bandwidth provisioned for the class is adaptively adjusted. Bandwidth is provisioned at each node from the SGSN at the top of the hierarchical topology to the RNCs, and from the RNCs down to the base stations. The provisioning policy is adapted based on the service-level agreement (SLA) contracted between the provider and the subscribers; but emphasis is given to the meeting of service-level requirements (SLS) within the SLA.

1.3 ORGANISATION OF THE THESIS

In the next chapter an introduction to next-generation UMTS networks is presented. Focus would be given to QoS in the UMTS core network. The Differentiated Services (DiffServ) service model is introduced and a mapping of the service model to the UMTS service model is described. In chapter 3, the use of bandwidth provisioning to enable multi-class QoS is examined. This is followed by a survey of the various bandwidth provisioning methods that are used today. The chapter ends with a formulation of the bandwidth provisioning optimization problem. In chapter 4, an introduction to the Reinforcement Learning (RL) theory is presented; including previous applications of RL in network control. Emphasis is then given to continuous state-action space RL methods, which are used in the proposed RL-based bandwidth provisioning algorithms. The chapter concludes by formulating the bandwidth provisioning optimization problem as an RL optimization problem.

Chapter 5 presents the proposed RL-based adaptive bandwidth provisioning solution for the UMTS core backbone network. The chapter includes a literature survey of previous adaptive bandwidth provisioning methods. Following that, details of implementation and simulation results of two proposed methods – Reinforcement Learning Adaptive Provisioning (RLAP) and Reinforcement Learning Dynamic Provisioning (RLDP) are described. In chapter 6, the RL-based adaptive bandwidth provisioning solution for the UMTS radio access network is presented as the second portion of the entire solution for the UMTS core network. A literature survey of previous mobility-based QoS provisioning methods is given prior to the implementation details and simulation results of the proposed Reinforcement Learning Bandwidth Provisioning (RLBP) solution. Finally, this thesis is summed up in chapter 7 with a brief conclusion emphasizing the contribution of this thesis and some recommendations for future work.

CHAPTER 2

NEXT-GENERATION UMTS NETWORKS

2.1 INTRODUCTION

Imagine the possibility of communicating and accessing information anytime, anyhow and anywhere. This dream is soon to be a reality with 3rd Generation (3G) mobile telecommunications systems [2-4]. 3G promises to deliver broadband wireless access up to speeds of 2Mbps. This would enable mobile devices to have connection speeds that are similar to fixed networks, thereby achieving the same quality of applications familiar to fixed network users. Furthermore, the promise of seamless global roaming would provide users with the effect of having that same high-speed connection regardless of location. 3G brings about the convergence of mobile communications and the Internet; a concept known widely as mobile internet.

The 3G Partnership Project (3GPP) [2] was formed to implement 3G by putting together a 3G mobile standard known as Universal Mobile Telecommunications Service (UMTS) [1]. UMTS is intended to form part of the International Mobile Telecommunications- 2000 (IMT-2000) [5] family of 3G standards. UMTS covers standards for the wireless transmission and protocols, the core network architecture, services and systems aspects and mobile terminals.

UMTS supports the use of the CDMA2000 and WCDMA wireless access technologies; two of the more prominent high-speed wireless transmission protocols anchoring 3G. In the core network, UMTS networks will initially be similar to current GSM/GPRS networks [6], but will evolve to an entirely new architecture that is IP-based [7,8]. This would enable mobile terminals to access the wealth of IP applications available today on the Internet. The radio access network (RAN) portion of the UMTS network will however remain similar to current architectures. This will allow the underlying core network to support both current and future RANs. UMTS systems will support a variety of services ranging from video-conferencing to m-commerce to multimedia applications. To enable support for such a wide range of services, 3GPP has adopted the IETF Differentiated Services (DiffServ) [9] service model, which defines how different classes of service can be supported in the core network. UMTS also defines 4 classes of service [10] that are to be supported over the wireless network. These 4 classes can be mapped to DiffServ per-hop behaviors (PHB) [11,12] to provide seamless service provisioning.

2.2 MOBILITY AND UBIQUITY

Mobility and ubiquity are the key concepts that 3G networks promise to provide. Mobility means that users are able to stay connected anytime and anywhere. Connecting without wires to a global network that can be reached at any point brings about the freedom for users to roam to any location. This could be as simple as from the kitchen to the garage, or as far-reaching as from the local main office to an overseas client's office. Users will no longer be bound by the need to find a fixed access to the network. More than this, 3G also promises ubiquity – the ability to access this network through a variety of means. This could be the plain voice call from your mobile phone or even an m-commerce transaction through a terminal in the moving taxi. Ubiquity means users can now stay connected anyhow they want, whichever way is the most natural and best-suited to the circumstances.

In this age of mobility, people are accustomed to communicating anytime and anywhere. Naturally, with the Internet so much a part of our lives today, people would want to access the Internet in the same way that they would communicating to each other. The need for real-time information anytime is already evident in today's time. Real-time information such as the latest soccer results and the current stock market prices are already available through mobile phones. This service could be extended in the future to watching live soccer matches on our mobile phones and having real-time charts of the stock market activity displayed on our laptops while on the bus.

Mobility also brings about the possibility of working from anywhere. This helps to make workers more mobile and less reliant on the office. The concept of the mobile office encompasses the objective of being able to access corporate email and databases from anywhere. It also allows workers to collaborate and run applications without being back in the office. This greatly increases productivity and enhances communication.

Sometimes, it may be more appropriate to have data displayed in different form factors. This is when ubiquity is important. The connection to the Internet should be readily available in whatever form best suits the circumstances. For example, a person who is on his way to a meeting may be on the mobile phone discussing details with his secretary. He may then require certain details to be sent to his PDA for his assessment. Upon arriving at the meeting room, the full presentation would need to be displayed on the projector screen. Members in the meeting may then have to access the same presentation, collaborate and change certain points via their own laptops. This scenario presented would require information to be accessed anyhow.

With the need to have services in various forms, at various times and with different requirements, 3G networks have to be designed with multi-layered quality of service (QoS) in mind. Services would require different levels of latency response times, information integrity and bandwidth. QoS is gradually being implemented in today's fixed networks. However, 3G networks bring in the dimensions of mobility and diversity. As described in the scenarios above, as users move from one place to another, they change contexts and may also be access networks using different access technologies. Users may be moving from a high bandwidth environment like their office to a wireless network in a crowded place, which may exhibit bandwidth congestion. There is therefore a need to provision for QoS to be maintained regardless of where the user moves. In a congested environment with limited bandwidth, users who require higher QoS would be given more bandwidth than users who only require minimal service.

2.3 CONVERGED BACKBONE NETWORK

In 2nd generation (2G) networks, mobile providers built their mobile networks on top of their fixed telephone network. These networks are mainly circuit-switched. This means that upon call connection, a fixed route is established and a dedicated channel has to be reserved from point-to-point for the entire call duration. This was sufficient for voice calls that have rather constant data rates. However, in 3G networks, data is to be carried on the network as well. Data transmission, unlike voice transmission, exhibits data rates that have high fluctuations. Using a circuit-switched network would mean very low bandwidth utilization as bandwidth would not be maximized during portions of the connection when there is little data being transmitted. Another problem with circuit-switched networks is that as long as transmission is required, the connection has to be maintained. Once the connection has ended, further transmission would require a re-establishment of the connection. This would not be appropriate for 3G, where users require to be always connected.

On the other hand, the Internet has been developed on a packet-switched network. As the Internet is used mainly for data purposes, this is highly efficient. However, in a packet-switched network, all data transmissions share the same network lines. This means that there would be contention for bandwidth if too much traffic is transmitted at the same time along the same routes. This is fine as long as users do not require any specific quality of service (QoS). However, this is not the case, as different services would require different QoS. Therefore, some form of QoS provisioning would be needed to grant services the required amount of QoS.

With the convergence of mobile voice and data in 3G, it would be more efficient and cost-saving to merge the telecommunications network with the data network [16]. In this way, infrastructure can be shared and data can be transmitted to mobile networks more efficiently through the packet-switched network. This means that voice service would need to be merged with data services. As voice traffic has strict latency requirements, QoS has to be provisioned such that voice traffic receives higher service as compared to data services. In this way, voice traffic can be transmitted over packet-switched networks without loss of QoS. The technology that enables this is known as Voice over IP (VoIP) [17-20].

The UMTS core network is specified to be a converged voiced/data IP network. UMTS supports most of the VoIP standards and QoS service models.

2.4 UMTS NETWORK

The UMTS network forms the transport backbone for all voice and data traffic, regardless of access technologies. The network therefore needs to support multimedia traffic. In this thesis, the UMTS network refers to both the UMTS Terrestrial Radio Access Network (UTRAN) [6] portion and the UMTS core network (CN) portion [7].

2.4.1 Architecture

The UTRAN architecture is shown in Fig. 2.1 as given in the first release of UMTS specification – 3GPP Release 1999 [6]. Recently, the 3GPP has proposed that the UTRAN be IP-based [21]. The work done in this thesis uses this architecture as IP is slated to be used in the core network as well. Therefore, it makes sense to have a unified system. The UTRAN comprises two types of nodes – the Radio Network Controller (RNC) and the Node B, which is the base station. The RNC is similar to the Base Station Controller (BSC) in today’s GSM networks. The RNC is responsible for the control of the radio resources within the network. It interfaces with one or more base stations, known as Node Bs. Together an RNC and the set of Node Bs that it supports are known as a Radio Network Subsystem (RNS). The topology of the UTRAN is usually hierarchical, with the top node being the Serving GPRS Support Node (SGSN) that the RNCs are connected to. In some large topologies, several UTRANs are needed to provide the coverage.

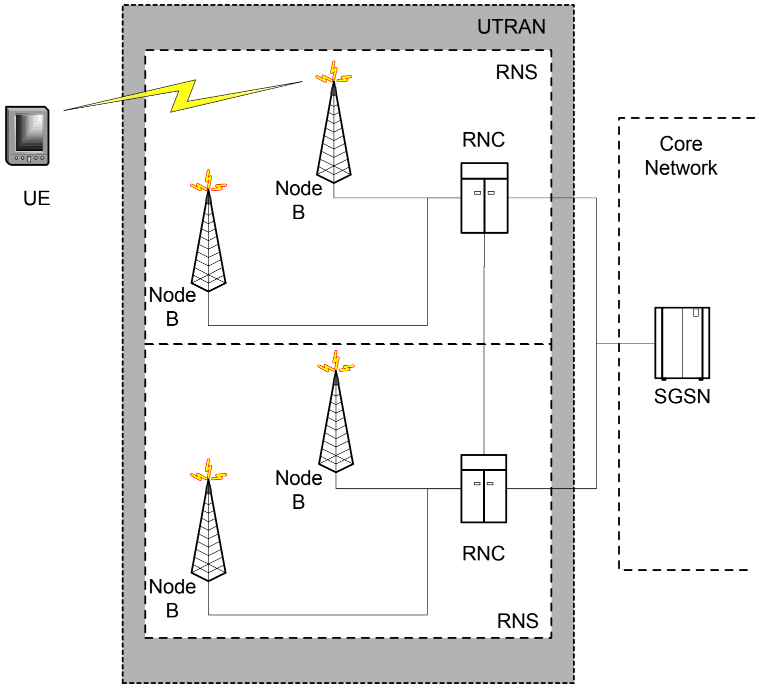

Figure 2.1: UTRAN Architecture

The mobile devices that connect to the Node Bs are known as UEs (made up of two parts; the TE and the MT). UEs may move from one Node B's coverage to another. This would trigger a soft handover (handovers that do not require disconnection). Soft handovers are achieved through the use of the Mobile IP protocol [22], which is adopted by 3GPP. Another possible type of handover is between RNCs. This occurs when a UE moves from one RNS's coverage to another. When a handover occurs, traffic directed to the previous Node B has to be re-routed to the new Node B. When mobility is high or when radio coverage is small, handovers can occur frequently, causing drastic changes in network patterns within the core network.

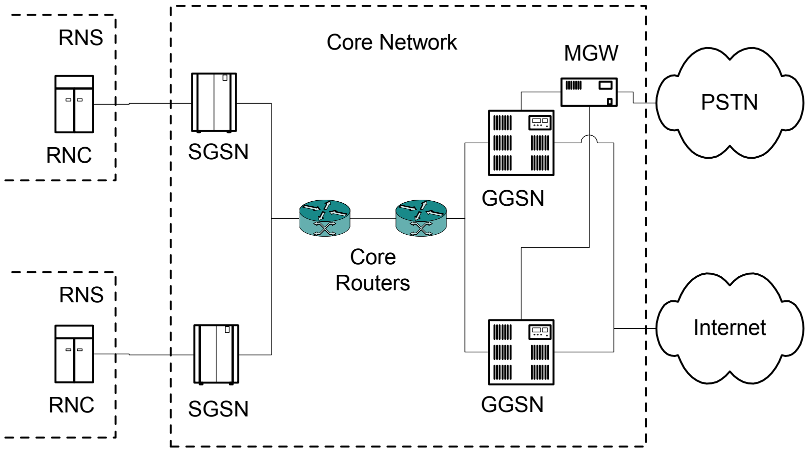

Figure 2.2: IP Core Network Architecture

The UTRAN is connected to the backbone network through the UMTS core network (CN). In 3GPP Release 5 [7], the UMTS core network makes use of an all-IP multimedia architecture as shown in Fig. 2.2. (Only the data plane is shown). In this architecture, both voice and data are largely handled in the same manner all the way from the UE to the ultimate destination. The UTRAN is connected to the core network through the connection between the RNC and the Serving GPRS Support Node (SGSN). Data traffic flows through the core network and exits to the Internet backbone via the Gateway GPRS Support Node (GGSN). Voice traffic flows through the GGSN as well, but is required to go through a Media Gateway (MGW) before heading out to the Public Switched Telephone Network (PSTN).

The GGSN may support one or more SGSNs, which in turn support several RNCs. Depending on how the UMTS core network is connected to the fixed backbone network, SGSNs may be connected to several GGSNs through core routers. This is especially the case when the network topology is widespread or when the traffic load exceeds the load capacity of a single GGSN. Hence, the UMTS core network topology tends to be a bit more mesh-like, rather than hierarchical like the UTRAN.

2.4.2 UMTS Quality of Service

Quality of Service (QoS) support in UMTS is based on a layered bearer service structure shown in Fig. 2.3 as defined in the 3GPP specification [10]. End-to-end QoS is provisioned by 3 layers. At the topmost layer, terminal equipment (TE) such as a laptop, PDA or mobile phone is connected to the UMTS network via a mobile terminal (MT). The UMTS bearer service then provides the QoS inside the UMTS network and performs functions necessary for QoS interworking with external networks. The external bearer service provides the QoS support outside of the UMTS network. This could be the familiar Differentiated Services (DiffServ) [9] or simply best-effort service.

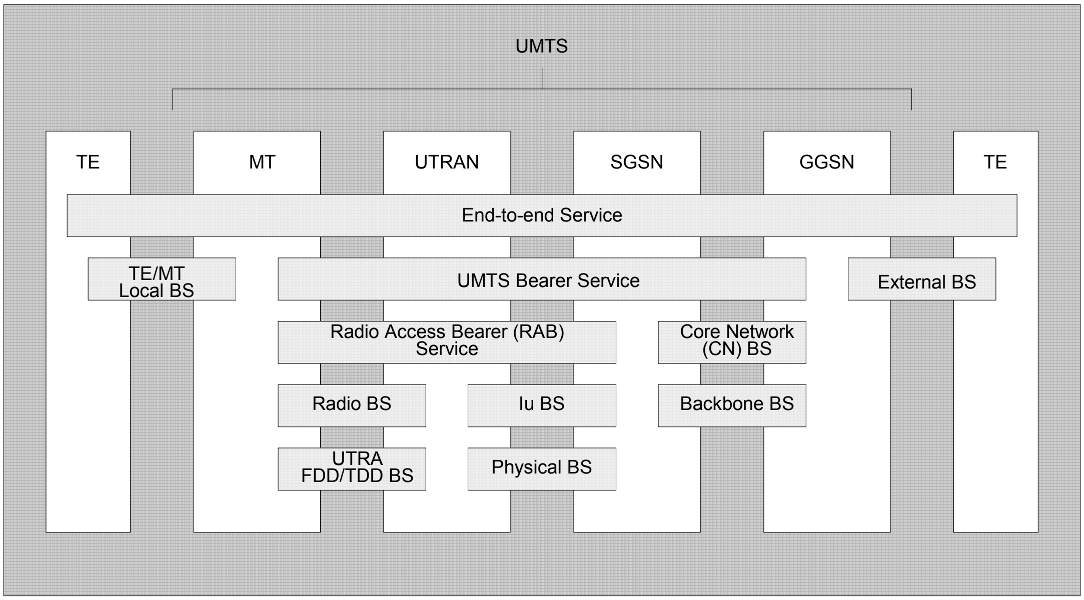

Figure 2.3: UMTS QoS Architecture

At the second layer, the UMTS bearer service is serviced by the radio access bearer (RAB) and the core network (CN) bearer. The RAB involves the air interface, the UTRAN and the link to the SGSN. The CN bearer, on the other hand, provides transport services within the core network segment located between the SGSN and the GGSN. At the lower layers, the RAB service itself consists of a radio bearer service between the MS and the UTRAN and an *Iu* bearer service between the RNC and the SGSN. The core network bearer service relies on the backbone network service, which may use different layer 2 and layer 1 transmission technologies.

UMTS specifications define four QoS classes, corresponding to different traffic QoS requirements:

- Conversational Class: This class of service is mainly for real-time applications such as voice and video communications. It is characterized by a very low delay tolerance.
- Streaming Class: Multimedia streaming applications come under this class, e.g., video streaming. For this class, a certain amount of delay is tolerable due to application level buffering.
- Interactive Class: Applications that require a “reasonable” response time come under this class. A higher scheduling priority compared to the background class is usually needed to ensure the round-trip delay requirement. Examples of applications are interactive web applications, database access and m-commerce.
- Background Class: This class takes the lowest priority as delay is not so much a concern for the applications under it. Traditional best-effort services like email and background file transfer come under this class.

Traffic Parameters	**Conversational Class**	**Streaming Class**	**Interactive Class**	**Background Class**
Maximum bit rate	×	×	×	×
Delivery order	×	×	×	×
Maximum SDU size	×	×	×	×
SDU format information	×	×		
SDU error ratio	×	×	×	×
Residual bit error ratio	×	×	×	×
Delivery of erroneous SDUs	×	×	×	×
Transfer delay	×	×		
Guaranteed bit rate	×	×		
Traffic handling priority			×	
Allocation/Retention priority	×	×	×	×

Table 2.1: QoS Attributes Defined for UMTS Bearer Service

The applicable QoS profile parameters for each class are shown in Table 2.1. As it can be seen, not all attributes are applicable to all QoS classes. The attributes are specified in ranges in the specification, depending on the QoS requirements usually associated with applications in the class.

2.4.3 Constraints and Challenges

The demand for diverse mobile services and the drive to cut infrastructure costs have fueled the need for efficient QoS provisioning. There has always been an argument that QoS can be achieved by provisioning more than sufficient bandwidth. However, this is not the case. Even till today, multimedia applications over wireline networks face network congestion. Perhaps only those who can afford fixed bandwidth lines are the exception. VoIP has also been implemented over wireline. As a public service over the Internet, VoIP has been given the image of a “cheaper than fixed line” alternative service. It is well-known that the quality of Internet VoIP is at best mediocre and unreliable. Only in the enterprise do we see VoIP being effectively implemented. The reason for this is that companies are able to afford expensive trunk lines across the Internet backbone. However, this cannot be feasible for a mobile network as virtually the entire Internet would have to be "booked" in order for clients to roam around globally!

With a growing need to cut costs as requirements for services increases, service providers can longer maintain separate networks for each service rolled out. Instead, service providers would need to converge multiple services onto a single infrastructure and manage them as a single entity. This would increase bandwidth efficiency and save infrastructure costs, while having the flexibility to add on new services at any point without building a new network. It is with this management concept that this thesis is presented.

The constraints and challenges that face a provider building a multi-service UMTS core network are as follows.

- Infrastructure costs: A service provider may find it expensive to acquire bandwidth, especially over a large network topology and in dense metropolitan areas. Therefore, bandwidth is a constraint that has to be managed and utilized efficiently. Furthermore, network equipment may be expensive to deploy, so a simple and yet effective system is needed.
- Traffic fluctuations: QoS management becomes critical when there is high traffic volume. Due to daily traffic patterns, fluctuations are inevitable. Service providers cannot overprovision bandwidth too much as it would be inefficient. Therefore when traffic volume is high, traffic with stricter QoS requirements must be able to obtain preferential treatment, like higher bandwidth provisions.
- Diversity of services: Different applications have different QoS requirements. Some may require low latency, while others may require

high assurance. While for other applications like interactive web-browsing, users may tolerate up to a reasonable amount of round-trip delay. To provision these different levels of service such that all QoS requirements are met is a challenging task. It is even more challenging when all the different classes of traffic travel through the same core network.

- On-demand services: When users require services, it is almost always on-demand. This makes usage very unpredictable and the traffic mix within the core network may change constantly. This makes QoS provisioning a dynamic problem. At times, there may be a high level of conversational class traffic. This does not necessarily lead to poor quality of service, unless the volume of other classes of traffic is equally high. Even then, only when there is a high volume of say streaming class traffic would there be a problem. Since QoS provisioning for one class would be at the expense of another.
- Mobility of users: This is the greatest concern in mobile networks. Users expect to have the same QoS wherever they roam to. This means that QoS has to be provisioned in advanced. Each time as users move, there is a route change and one or more links may be affected. The links to which the user's traffic would be transferred to must have enough bandwidth available in order for QoS to be maintained. Reservation and admission control is a simple way of providing this. However, reserving bandwidth for a large number of users is highly inefficient, as this would require vast amounts of bandwidth to be reserved. An intelligent way of handling the reservation problem is needed for service providers to be profitable.

- Links with different capacities: In a UMTS network, there are links with capacities ranging from below 2Mbps for wireless links to OC-3 links (155Mbps) in the core network. These links are usually arranged in a hierarchical topology, where the lower bandwidth links are served by a higher bandwidth link. Therefore, the router that controls the traffic from a high bandwidth to a low bandwidth link has to provision downstream bandwidth efficiently and effectively (maintaining end-to-end QoS).

2.5 DIFFERENTIATED SERVICES

The Differentiated Services (DiffServ) architecture [9] was defined by the International Engineering Task Force (IETF) DiffServ Working Group as a simple and scalable service model for service differentiation within an IP network. This means that traffic can be classified and treated with different QoS levels whilst being transported through the same network. The DiffServ model has a scalable architecture because most of the packet classification and conditioning is done at the edge of the network. The core of the network merely forwards the packets from one hop to another.

In the DiffServ model, traffic that enters a network is first classified and then possibly conditioned at the edges of the network. Depending on the result of the packet classification process, each packet is associated with one of the behavior aggregates (BA) supported by the DiffServ domain (DS domain). A BA is a service class within the DiffServ framework. Packets belonging to the same BA are marked with the same DiffServ codepoint (DSCP) and are given the same forwarding treatment by a router. The DSCP is used by the router to select the per-hop behavior (PHB) that a packet experiences at each hop within a DS domain.

2.5.1 Architecture

A DS domain is a contiguous set of routers that operate with common sets of service provisioning policies and PHB group definitions. (Fig. 2.4). A DS domain is typically managed by a single administrative authority that is responsible for ensuring that adequate network resources are available to support the service level specifications (SLS) defined in the service level agreement (SLA) [29].

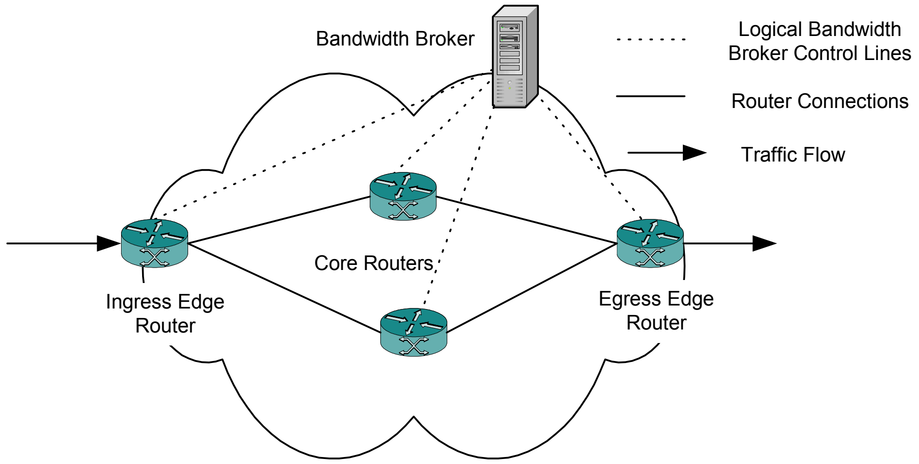

Figure 2.4: Differentiated Services Domain

A DS domain consists of DS boundary nodes (edge nodes) and DS interior nodes (core nodes). The routers within the DS domain may be controlled by a single administrative entity known usually as a bandwidth broker (BB) [30].

2.5.1.1 Edge Nodes

Edge nodes function as both ingress and egress nodes for different directions of traffic flows. When functioning as an ingress node, an edge node is responsible for the classification, marking and possibly conditioning of ingress traffic. It classifies each packet based on the header with a DSCP. When functioning as an egress node, the edge node may be required to perform traffic conditioning on traffic forwarded to a directly connected peering domain.

Packet classification of packets is done based on the content of fields in the packet header. There are two types of classifiers:

- A behavior aggregate (BA) classifier selects packets based on the value of the DCSP only.
- A multifield (MF) classifier selects packets based on a combination of the values of one or more header fields. These fields can include the source address, destination address, DS field, protocol ID, source port, destination port, or other information, such as the incoming interface. The result of the classification is written to the DS field to simplify the packet classification task for the core nodes.

Traffic conditioning may consist of traffic metering, marking, shaping and dropping, but not necessarily all. The following describes the elements:

- A meter is used to measure a traffic stream to determine whether a particular packet belonging to the stream should be considered in-profile or out-of-profile. The meter passes the information to the other elements in the conditioner.
- A marker is used to write the DSCP into the DS field of the packet header, that has been decided by the packet classifier and the information passed from the meter.
- A shaper is used to delay some or all packets in a traffic stream to bring the stream into conformance with its traffic profile.
- A dropper is used to discard some or all packets in a traffic stream to bring the stream into conformance with its traffic profile.

The two functions described above are shown in Fig. 2.5.

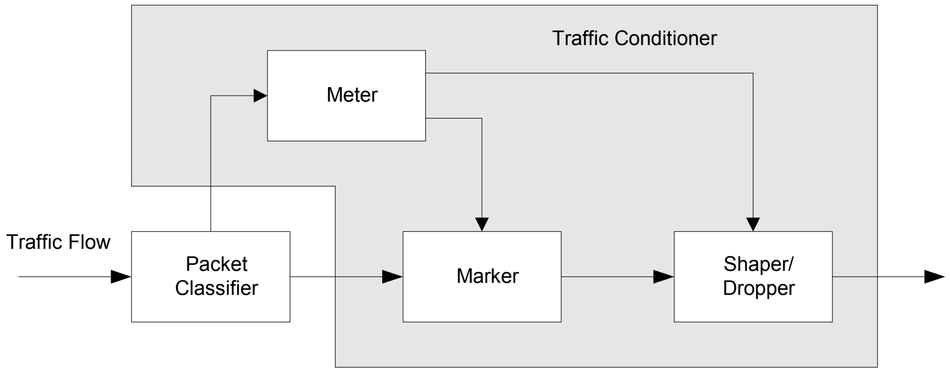

Figure 2.5: Packet Classifier and Traffic Conditioner

2.5.1.2 Core Nodes

Core nodes are high-speed routers that forward packets according to the PHB assigned based on the DSCP in the packet header. Core nodes map the DSCP to one of the PHB groups supported by all the core nodes within the DS domain. Core nodes connect only to another core node or edge node within the same domain.

2.5.1.3 Per-Hop Behaviors (PHB)

When a packet is marked with a DSCP, routers within the DS domain treat the packet according to a PHB associated with the DSCP marking. This brings about differentiation between the various classes. The three common PHBs defined by IETF are the Expedited Forwarding (EF) PHB [31], the Assured Forwarding (AF) PHB group [32] and the Best Effort (BE) default PHB.

The EF PHB is the highest class of service and is reserved for low loss, low latency, low jitter, assured bandwidth edge-to-edge service. There can be only one instant of EF in a DS domain, unlike the AF PHB group. The AF PHB group provides four independently forwarded AF classes. A packet in each of these AF groups can also be assigned to one of three drop precedences. Therefore, theoretically 12 instances of AF with can exist in a DS domain. The AF PHB group is mainly used for services with looser QoS requirements. The tiered forwarding treatment gives different bandwidth priority to each class of service. This is useful for services with different latency requirements. The drop precedences provide differential treatment to loss performance. This can be used to segregate low loss services like transaction-based services from loss tolerant services like video streaming. The BE PHB is for traffic that does not have any particular QoS requirement, like email. The bulk of Internet traffic would fall under this category. Therefore, it is important that service providers still provide as much leftover bandwidth as possible to maintain good relations with the majority.

PHBs can be implemented through the scheduler and the dropper in each DiffServ router. For the EF PHB, the scheduler should provide more than enough bandwidth so that low delay and low loss can be achieved. The dropper should be set to drop EF packets once a queue of EF packets starts forming. For the AF PHB group, the scheduler should be set to provide a rate assured by the SLS for each class of traffic. The dropper should also be set to drop packets from classes with higher packet loss tolerance and lower delay tolerance first. To achieve independent bandwidth provisioning, a form of weighted fair scheduler [33-37] is recommended.

2.5.1.4 Bandwidth Broker (BB)

In the DiffServ framework, the DS domain management entity has not been defined. The bandwidth broker (BB) model [30], as defined by the Internet2 Qbone BB Advisory Council [39], is a good implementation of a DS domain management entity for DiffServ. The BB is used in this work to manage resources in the DS domain and to configure the DS nodes dynamically through the use of network management protocols such as SNMP (Simple Network Management Protocol) [40].

A BB manages the QoS resources within the DS domain based on the SLS agreed upon. The SLS is a translation of an SLA into the appropriate information necessary for provisioning and allocating QoS resources within the network devices. The BB may gather and monitor the state of QoS within the DS domain through traffic measurements deployed around the domain to ensure that QoS is maintained. This measurement-based approach is an alternative to rigid reservation-based approaches that require complex signaling between BBs of various DS domains and DS nodes within the domain. The measurement-based approach is also more bandwidth efficient as it does not require bandwidth to be reserved. However, traffic measurements may be costly as probes and counters that eat up processing cycles are required to be deployed at each link. We foresee that efficient standalone gear may be developed in the near future to implement network measurements efficiently [41,42]. The measurements will then be fed back to the BB and the BB can then make decisions on how to provision QoS efficiently. The results of the decisions are translated into DiffServ router configurations that may change the traffic conditioner settings, the buffer management configuration and the scheduler settings. The measurement-based management of QoS in a DS domain can thus be seen as a closed-loop control problem.

2.5.2 Mapping of UMTS Service Classes to DiffServ Classes

There have already been proposals to map the UMTS service classes to the DiffServ classes [11,37] so that the DiffServ model can be used in the UMTS core network. However, in this thesis, a simpler mapping is preferred. This is done to simplify the presentation of the algorithms proposed and to reduce the complexity of the implementation for simulation purposes. The algorithms can be expanded to greater dimensions if needed depending on the number of service levels a service provider offers in actual implementation. For example, if five service classes are to be offered – VoIP and Video conferencing conversational classes, a video streaming class, an interactive class and a default class, then the algorithm would provision for five bandwidth proportions.

The thesis presents a solution for a three-class DiffServ implementation – EF, AF and BE. The UMTS classes are mapped as given in the table below.

UMTS Class	DiffServ Class	QoS Requirements
Conversational	EF	Low delay Low loss
Streaming	AF	Medium delay Medium loss
Interactive	EF	Low delay Low loss
Background	BE	Reasonable delay Reasonable loss

Table 2.2: Mapping of UMTS Service Classes to DiffServ Classes

CHAPTER 3

BANDWIDTH PROVISIONING

3.1 INTRODUCTION

Bandwidth provisioning is required for the UMTS core network to support a diversity of services. This is because traffic flows with various QoS requirements share the same network and may contend for the limited bandwidth resources. As traffic patterns are not constant, there will be intervals of time when the total amount of data being transmitted far exceeds the capacity of the links. This phenomenon is known as traffic burstiness. Its occurrence is due to the random nature of user usage and the way IP packets are packetized for transmission. Therefore, though the average amount of traffic over a long period of time may be constant, there may be congestion over shorter periods.

If however, the congestion periods are long enough to cause delays and packet drops (due to buffer space being exhausted), the amount of bandwidth provisioned for delay and loss intolerant traffic should be increased at the expense of other more tolerant traffic flows. This is known as service differentiation. Service differentiation allows certain classes of traffic to have better "service", in terms of priority to network resources, than others. This is usually required for applications that need a high level of QoS. For example, video conferencing applications need data to be sent with little delay. Otherwise, there will be irritating synchronization problems between the two parties trying to converse with each other.

A service differentiation model called DiffServ has been described in the previous chapter and will be used in this work. Bandwidth provisioning in DiffServ networks involves the determination of the amount of bandwidth to allocate for each PHB across each network link in the DS domain. This is usually done at the router's outgoing ports through packet scheduling. Through bandwidth provisioning, behavior aggregates (BA) share the bandwidth in a certain proportion as they contend for the use of a link to transmit data packets from one node to another. By allocating different proportions of bandwidth, the service levels of each BA can be differentiated.

3.2 QOS THROUGH BANDWIDTH PROVISIONING

The original intention of DiffServ was to provide premium or better-than-best-effort service. Therefore, the concept of QoS was a qualitative one. This means that services requiring a higher level of QoS would be allocated a PHB that gives preferential treatment in terms of scheduling and buffering. However, the absolute level of service is not defined. This may be sufficient as a starting point, but as users demand better service quality and become less tolerant to poor service, the need for quantitative QoS arises. In the new definition of the expedited forwarding PHB [31], absolute values of delay and loss requirements can be specified. This can be seen as a trend towards having more concrete SLS defined in SLAs [29,43]. Unfortunately, as demands grow the provisioning of resources to meet strict QoS requirements becomes much more complicated. This is especially so when the traffic mix in the network is so diverse and QoS mechanisms of a large network have to inter-work with each other [44].

It is important to note that service differentiation only becomes enforced when there is congestion. It has already been argued in the previous chapter that having larger capacity links to combat congestion is inefficient, as data traffic has a high level of burstiness. Therefore, the best way of handling congestion is to manage resources in the network. In the following analysis, it is shown that, given a comfortable level of buffering, dynamic bandwidth provisioning can be used to determine the QoS achieved. This means that a solution that effectively and intelligently provisions bandwidth adaptively is good enough to maintain a reasonably strict level of QoS. The analysis is based on tail-drop FIFO queues and Weighted Fair Queuing (WFQ) schedulers [33]. Fig. 3.1 shows the WFQ model that is to be used. For each service class i, λ_i is the arrival rate of the packets, σ_i is the buffer size, q_i is the average queue length and ψ_i is the weight assigned. r is the link capacity.

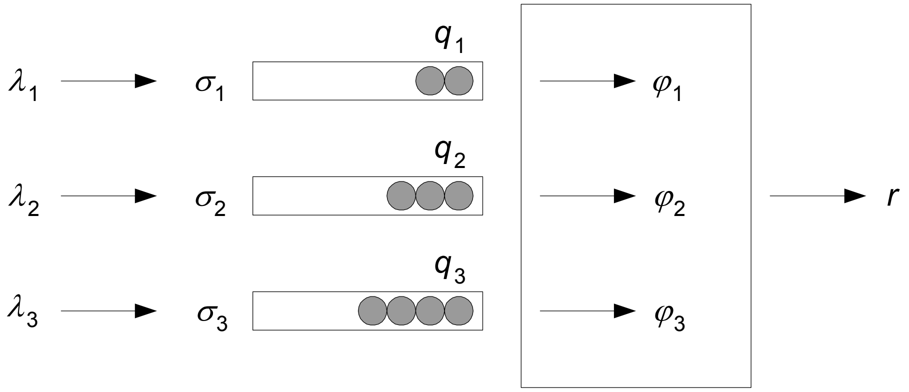

Figure 3.1: WFQ Scheduler

3.2.1 Throughput Analysis

It is common to use a weighted fair queuing (WFQ) scheduler to provision bandwidth on DiffServ links. A WFQ scheduler is bandwidth efficient in the sense that it fairly re-allocates any excess bandwidth not used by a flow to other flows that require more than their allocated bandwidth. This is especially useful when there is congestion and a particular aggregate flow does not need its full allocation. Other aggregate flows can then use the excess bandwidth. The amount of bandwidth provisioned by WFQ scheduler across a link for a particular aggregate flow i follows the equation

$$B_i = \frac{\xi_i}{\sum_{j=1}^{n} \xi_j} r \tag{3.1}$$

where ξ_i is the weight given to flow i, n is the number of flows utilizing the link and r is the link capacity. This is however the amount of bandwidth guaranteed to an aggregate flow and does not guarantee anything for individual flows within the aggregate. It is only possible to assure an average throughput over longer timescales for individual flows. The proportion of bandwidth allocated to the flow is given by the proportion $\psi_i = \frac{\xi_i}{\sum_{j=1}^{n} \xi_j}$.

From equation (3.1), is can be seen that the throughput requirements of the service class can be met by provisioning ψ_i accordingly.

3.2.2 Queuing Delay Analysis

Queuing delay is mainly caused by packets building up in queues. This happens when there is insufficient bandwidth to service the flow. For a non-preemptive scheduler like the WFQ scheduler, slight delays caused by burstiness cannot be avoided. Thus, it is assumed that some delay for a small percentage of packets is tolerable. Rather, the aim should be to maintain the percentage of packets that face excessive delay below the tolerable level.

The delay bound for a WFQ scheduler is given by

$$D_i = \frac{\sigma_i}{\psi_i r} + \frac{p_{\max}}{r} \tag{3.2}$$

where p refers to packet size.

The average latency is given by

$$d_i = \frac{q_i p_i}{\psi_i r} + \frac{p_{ave}}{r} \tag{3.3}$$

$$\Rightarrow d_i \propto {}^{q_i}\!/_{\psi_i} \tag{3.4}$$

To maintain a certain average latency, the average queue length q_i has to be kept small. This is done by ensuring that $\lambda_i < \psi_i r$. We clearly see that by increasing the bandwidth provision ψ_i, q_i will not only be kept small, but d_i will also be reduced due to both a small q_i and a large ψ_i (given in equation (3.4)).

Note that, $\max(\psi_i) = 1$

$$\Rightarrow \min(D_i) = \frac{\sigma_i + p_{\max}}{r} \tag{3.5}$$

and $$\min(d_i) = \frac{q_i p_i + p_{ave}}{r} \tag{3.6}$$

Therefore, if q_i is reduced due to a large ψ_i ($\psi_i \to 1$), then from (3.6) we see that $d_i \to {}^{p_{\max}}\!/_{r}$ (a constant value). However, we cannot indiscriminately set the provision of one class to be large, as this will cause a trade-off in a lower provision for another class. We can see from the above analysis that the average latency of a service class can be controlled by strictly by the provisioning of ψ_i. Due to the trade-off in the provisioning of one class with another, it is important to dynamically and adaptively control the bandwidth provisions of each class with respect to one another.

3.2.3 Packet Drop Analysis

Packet losses due to full queue buffers can occur in 3 ways:

- When the burst rate (the rate at which packets enter the queue over a short period) $\gamma_i > \psi_i r$. This occurs over short timescales, when the burst time (the period of time over which multiple packets are entering the queue) $t_b > \frac{\sigma_i}{\gamma_i - \psi_i r}$. Such packet losses can only be prevented by adequately allocating σ_i, as setting ψ_i large enough to absorb bursts is similar to over-provisioning, which is bandwidth inefficient.
- When $\lambda_i > \psi_i r$. This can occur over long timescales and is due to insufficient bandwidth to meet sustained traffic demands. This can be solved through increasing ψ_i.

- When there are large IP packets being served in other queues and preemption is not an option. This problem seldom occurs in large capacity backbone links and is more commonly seen in low bandwidth links, where the transmission time for a large IP packet can take up considerable time. Packets start to get dropped when $p_{ave}/r > \sigma_i/\lambda_i$. To prevent packet drop, $\sigma_i > \frac{p_{\max}\lambda_i}{r}$.

3.3 METHODS OF PROVISIONING

A survey of the methods of provisioning is done here. The reasons why priority queuing and bandwidth partitioning methods are insufficient for provisioning multi-service UMTS core networks are given.

3.3.1 Priority Queuing

The use of priorities as a way of providing better service is common not only in networking [45-49] but also in any service industry. Those who want better service pay more and are served first. There are a few assumptions made here. Firstly, there is a desire for a premium service that is not possible to get otherwise. This implies that normally when there is no service differentiation, the service is not good enough for some. Secondly, customers who are given priority are not too concerned about the actual service level. They just want a better service. If the better service is still not good enough, they are willing to pay even more for even better service. This brings about the concept of multi-priority. Lastly, in a priority-based system, there is no way to guarantee a specific level of service, as the service level may decline when more customers join the service class.

The points mentioned above are similar in multi-service networks that employ priority queuing. If priority queuing is used in DiffServ networks to differentiate, for example, EF, AF and BE classes, it would be difficult to guarantee any strict QoS. The QoS received by traffic in the AF class is dependent on volume of EF traffic. When EF traffic volume is high, the QoS of AF traffic degrades. This is known as starvation, as the needs of a higher priority customer are met first regardless of the needy low priority customer. Traffic in the EF class is not protected from degradation either. Since the priority is applied to the whole aggregate flow, individual flows cannot be guaranteed any deterministic QoS [47,49]. Therefore, the implementation of priority queuing in DiffServ networks is at best for qualitative QoS and cannot be used for quantitative QoS.

Methods have been proposed to introduce absolute QoS to DiffServ networks based on priority queuing. These include using admission control and dynamic priority assignment [45], aggregation control and buffer management [47,48], and traffic conditioning and measurement-based admission control [49]. However, these methods are mainly used to provision for EF class traffic. AF traffic is not considered. It is fair to say that if AF traffic requires similar delay bounds (albeit looser ones), some form of extension to the priority queuing that have been proposed can be used, but this has not been researched so far. Therefore, it may not be feasible to use an entirely priority-based scheme to implement DiffServ in UMTS multi-service networks.

3.3.2 Bandwidth Partitioning

Bandwidth partitioning is by far the most popular way of provisioning service. It is simple and effective. Bandwidth partitioning is done by allocating a fixed amount of bandwidth to the customer. This means that there is no sharing of bandwidth. It is as if there are separate pipes from point to point. Therefore, each customer is guaranteed a certain bandwidth, alleviating the problem of starvation associated with priority queuing. However, there is a trade-off. Bandwidth partitioning methods usually employ over-provisioning. This is bandwidth inefficient as mentioned previously. Either users end up paying for more than they need or service providers end up with fewer customers than they could have dimensioned for.

In a DiffServ network, bandwidth partitioning is used to allocate and reserve bandwidth for each class. In the strictest sense, once bandwidth has been fully allocated, an increase in bandwidth allocation of one class can only result from a re-allocation from other classes. The optimal allocation of bandwidth to each class on each link is known as the bandwidth provisioning/allocation/dimensioning optimization problem. Links have to be provisioned such that each class has sufficient bandwidth to meet traffic demands and QoS requirements. The problem is formulated in section 3.4 of the thesis.

Usually, the problem of estimating traffic patterns and meeting Qos is separated from the allocation problem. (We have chosen to define and solve the problem as a whole, as this would lead to better bandwidth allocation efficiency). The problem of estimating the required bandwidth is not a simple one. We discuss some methods that have been proposed later in section 5.2. Most providers resort to over-provisioning based on the average volume of traffic observed, the burstiness of the traffic and how strict the QoS requirements of the services provided are.

Methods have been proposed to reduce the rigidity of the bandwidth allocation. Link-sharing strategies [50] have been proposed to have shared portions of bandwidth in a hierarchical manner. Over-provisioning is reduced as over-provisioned portions can be shared. Virtual partitioning [51] employs a penalty concept, where bandwidth is initially nominally provisioned. When excess bandwidth is required, a penalty is imposed. The aim is to minimize the penalty and thereby increase the bandwidth allocation efficiency. Optimal allocation methods are described in references [52-55]. They include re-allocation methods, greedy methods, preemption methods and measurement-based methods [52-55].

All the bandwidth partitioning methods described in the literature are meant for use in more stable (in terms of predictability of traffic pattern) wireline networks over long timescales. Traffic patterns are very dynamic in UMTS core networks. This is because users are expected to move about in the network, causing traffic patterns to change rapidly. Such methods would become too complex to employ in shorter timescales.

3.3.3 Weighted Fair Queuing

Weighted Fair Queuing (WFQ) [33] is a fair and bandwidth efficient way of provisioning bandwidth. In WFQ, bandwidth is shared in proportions according to a ratio of weights. Any excess bandwidth not used from any proportion is fairly re-distributed according to the ratio of the other proportions. In this way, there is absolutely no wastage of bandwidth. WFQ can theoretically achieve full utilization of link capacity when the incoming traffic rate is greater than the link capacity. Another advantage is that bandwidth is inherently re-allocated without the need for extra mechanisms, unlike bandwidth partitioning methods.

WFQ is capable of assuring bandwidth for each class, making it a good substitute to bandwidth partitioning. A minimum bandwidth according to the proportion allocated and the capacity of the link is assured. The proportion allocated is determined by the WFQ weight assigned in relation to the other classes. WFQ weights can be assigned to each class in the DiffServ framework to provide service differentiation [56]. The actual bandwidth provisioned to any class can be greater than or equal to the minimum assured as excess bandwidth unused by other classes may be re-allocated.

In section 3.2, it was shown that the adjustment of the WFQ weights can change the QoS experienced. The only drawback to using WFQ is the inability to strictly guarantee QoS. This means that delay and loss bounds have to be loose. However, this is not so much of a problem for most applications, other than mission-critical applications. A loose, or correctly a probabilistic, bound may not be able to strictly guarantee QoS, but it can give an assurance on the percentage of traffic meeting the QoS bounds. Most users would not mind an occasional delay or loss, as long as most of the time they are getting good service. If mission-critical applications, such as real-time mobile medical equipment carried by paramedics, need to be provisioned for, then a combination of priority queuing and WFQ can be used [57].

3.4 FORMULATION OF BANDWIDTH PROVISIONING OPTIMIZATION PROBLEM

Based on the method of provisioning using weighted fair queuing, the bandwidth provisioning problem can be broadly formulated as follows:

For all nodes,

Given,

$x_{EF,j}(t)$, $x_{AF,j}(t)$, $x_{BE,j}(t)$ $\in x(t)$, and C_j for all $j \in$ all outgoing links from node

where $x_{EF,j}(t)$, $x_{AF,j}(t)$ and $x_{BE,j}(t)$ are the traffic rates of each DiffServ class (EF, AF and BE) entering the node destined to leave through link j, $x(t)$ is the set of all input traffic rates at every node, and C_j is the capacity of link j.

Select,

$w_{EF,j}(t)$, $w_{AF,j}(t)$, $w_{BE,j}(t)$ $\in y^x(t)$

where $w_{EF,j}(t)$, $w_{AF,j}(t)$ and $w_{BE,j}(t)$ are the weighted fair proportions of bandwidth for each class on link j and $y^x(t)$ is the set of all actions selected at every node based on the input $x(t)$.

Such that,

$$r = \max_{y \in Y}\left(r \middle| x(t), y^x(t)\right)$$

where Y is the set of all possible actions and r is the net revenue (gross revenue minus QoS penalties) earned depending on the pricing plan and SLA contracted, which spell out the revenue earned for provisioning each class of service and the penalties paid out if any of the QoS requirements are breached.

Constrained by,

$$l_{EF} < l_{EF}*,\ l_{AF} < l_{AF}*,\ l_{BE} < l_{BE}*,\ D_{EF} < D_{EF}*,\ D_{AF} < D_{AF}*,$$

where l and D are the actual loss and delay percentages and $l*$ and $D*$ are the loss and delay requirements. The loss requirement is given as the maximum percentage of packet loss tolerable and the delay requirement is stated as a bound on the percentage of packets that are allowed to exceed the end-to-end latency bound.

Depending on the objective function, the solution space of the problem may vary with the tightness of the QoS constraints. Although a solution can be readily found through some approximate method when the objective function is to meet some loose QoS targets, a solution is not so easily found when the QoS constraints are tight and the objective is to maximize revenues from bandwidth usage. There are some constraints that may make the problem non-tractable.

Firstly, the problem above is a continuous time problem. This means that at every time instant, there is a different optimization problem to solve due to the continuously changing traffic rates of each class. As the traffic entering each node is a random variable depending on time, the weights may have to be changed from time to time if any of the constraints are not met. A method of circumventing this is to discretize the time space into time intervals. This would lead to an approximation of the traffic rates and a set of weights to be used in each time interval. Unfortunately, selecting an appropriate time interval is a difficult task, as too long an interval would render the solution ineffective and too short an interval would become too computationally intensive.

Another problem confronting us in solving the bandwidth provisioning problem is the fact that traffic rates are random variables that have distributions that are not easily obtained. From the above formulation, we see that in order to set the weights, for time interval t, we have to know the traffic rates for time interval t at the beginning of the interval. This would require *a priori* information on the traffic that has yet to arrive. This is a well-known problem of bandwidth provisioning, and much work has been done in trying to model, approximate or even predict traffic distributions. While traffic prediction and characterization methods are fairly advanced today, they still lack a way to balance multi-class traffic mix. Due to the lack of feedback from the network and exchange of state information between the flows, traffic predictors concentrate on individual flows, neglecting the need for load balancing between the flows, so as to achieve QoS all round.

Lastly, in order to determine how much bandwidth to provision, the network topology and QoS mechanisms need to be considered. For example, the capacity of the links, the number of nodes and the buffer management schemes and settings, all need to be used in theoretical computations to achieve the desired QoS. This means that the behaviors of network mechanisms need to be well-understood with respect to changing traffic. Although the area of network theory has been well-researched [58,59], results often give loose bounds and are based on assumptions.

From the former discussion, it can be seen that the bandwidth provisioning problem is a “hard” problem that cannot be solved precisely. Many of today’s networks that are deployed make use of a static provisioning method, where a conservative level is used to over-provision the network. Based on past history, the bandwidth is provisioned to balance QoS performance and bandwidth utilization. However, this is a very crude and highly approximate method. Some form of adaptive bandwidth provisioning has to be used if bandwidth is to be used efficiently.

A different approach has to be taken to circumvent the problems mentioned above. Feedback control is a means whereby the network is able to feedback how the current bandwidth provisioning is performing in terms of meeting QoS. By using an iterative method, a good approximate solution can be obtained through experience. Feedback control methods may be model-based or modeless. The modeless methods are appealing as the modeling of network mechanisms can be too cumbersome. Feedback control methods do not need to have any prior knowledge of the traffic, as methods can be made to be reactive or conservatively proactive. The methods proposed in this thesis solve the bandwidth provisioning problem through iterative optimal control using a reinforcement learning framework. We describe how reinforcement learning-based optimal control is able to solve the bandwidth provisioning problem and why it is a favorable method.

CHAPTER 4

REINFORCEMENT LEARNING-BASED PROVISIONING

4.1 INTRODUCTION

The concept of learning is a very integral part of life. Right from the moment a child enters the world, he learns from the environment around him. This interaction with the environment allows him to learn new things and to do things in more efficient ways. For example, the child may first learn to pick up a fork and start to jab at objects on his plate, but gradually, he learns to use his fork for anchoring, twirling and mashing his food. This learning can only be a result of his exploration and cognitive ability to reason what ways are better than others. Each time he explores, the interactive environment gives him a feedback; perhaps by turning the fork he could get better results. The child then registers this as a positive feedback and would like to try something similar again. After repeated successful results from variations of turning the fork, the action of turning the fork to eventually get the spaghetti twirled is reinforced. This is a good example of how learning can help an agent (the learner) achieve optimal results in the long run without requiring any prior knowledge, understanding of the mechanics, or supervision from a learned agent.

All learning examples share similar features. They involve the interaction between an active decision-making agent and its environment in which the agent seeks to achieve a goal despite uncertainty about its environment. The agent’s actions are permitted to affect the future state of the environment (the turning of the fork in the pool of spaghetti gets it around the fork), thereby affecting the options and opportunities available to the agent at later times. At the same time, the effects of actions cannot be fully predicted, and so the agent must frequently monitor its environment and react appropriately. Lastly, all learning agents can use their experience to improve their performance over time. Though prior knowledge of the agent influences what is useful or easy to learn, the interaction with the environment is essential for adjusting behavior to exploit specific features of each task.

4.2 REINFORCEMENT LEARNING THEORY

Reinforcement learning [60] is the learning of a mapping from situations, presented to a learning agent from the environment, to actions, taken to influence the environment, so as to maximize the positive feedback received or achieve an optimal goal. Different from other forms of machine learning, the learner does not need to be directly told which actions to take but must instead discover the actions that yield the most reward by trying them. A reinforcement learning agent must be able to *sense* information pertinent to the state of its environment and must be able to take *actions* that affect the state. The agent must also have a *goal* defined in terms of how the environment behaves over time under the influence of its actions. These three aspects – sense, action and goal – are the fundamental blocks of all reinforcement learning algorithms.

All reinforcement learning algorithms require a particular synergistic combination of search and memory. Search is required to find good actions, and memory is required to remember what actions worked well in what situations in the past. A dilemma that arises from this is that a tradeoff between exploration and exploitation exists. To obtain a lot of reward, a reinforcement learning agent must prefer actions that it has tried in the past and found to be effective in producing reward. But to discover which actions these are, it has to select actions that it has not tried before.

In the following, an exposition of the basic concepts of reinforcement learning theory is given. The concept of reinforcement learning was also developed at the same time by Bertsekas [61] and is known as Neuro-Dynamic Programming (NDP). The advantages of using reinforcement learning are then discussed and examples of how reinforcement learning has been applied effectively to some network control problems are described.

4.2.1 Basic Concepts

The reinforcement learning problem is framed as a problem of learning from interaction to achieve a goal. The learner and decision-maker is known as the *agent*. The entity it interacts with, comprising everything outside the agent, is called the *environment*. The agent and the environment interact continually with the agent selecting actions and the environment responding to those actions and presenting new situations to the agent. The environment also gives feedback about its state in terms of rewards or penalties, which the agent tries to maximize over time.

The agent and environment interact in a sequence of discrete time steps, $t = 0,1,2,3...$ At each time step, t, the agent receives some representation of the environment's *state*, $s_t \in S$, where S is the set of possible states. Based on the state, the agent then selects an *action*, $a_t \in A(s_t)$, where $A(s_t)$ is the set of actions available in state s_t. One time step later, in part as a consequence of its action, the agent receives a numerical *reward*, $r_{t+1} \in \Re$, and finds itself in a new state, s_{t+1}. Fig. 4.1 shows the agent-environment interaction.

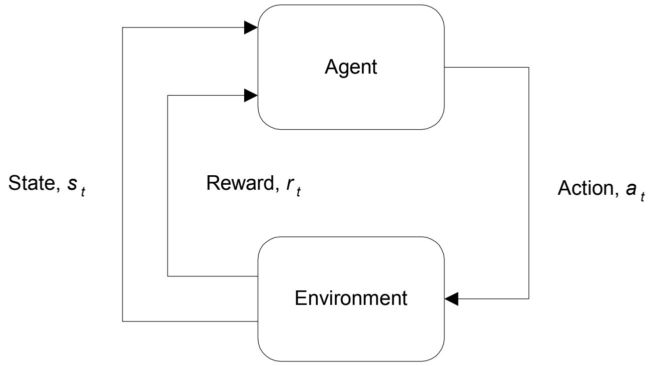

Figure 4.1: Reinforcement Learning Framework

At each time step, the agent maps the state s_t to an action a_t. This mapping is called the agent's *policy* and is denoted as π_t, where $\pi_t(s,a)$ is the mapping function that selects $a_t = a$ if $s_t = s$. Reinforcement learning methods modify the agent's policy through its experience. The purpose or goal of the agent is formalized in terms of the *reward*, which passes from the environment to the agent. The reward is just a single number whose value varies from step to step. Informally, the agent's goal is to maximize the total amount of reward or minimize the total amount of penalty it receives over the long run.

The framework is abstract and very flexible, allowing it be applied to many different problems in many different ways. This follows the inherent nature of reinforcement learning, which can be applied to any problem that fits the above framework. For example, the actions might be low-level controls such as the voltages applied to the motors of a robot arm, or high-level decisions such as whether to go for a movie or have lunch first. Similarly, the states can take a wide variety of forms. They can be completely determined by low-level sensations, such as direct sensor readings, or they can be more high-level and abstract, such as the mood of a person. In general, actions can be any decisions we want to learn how to make, and the state representations can be anything we can know that might be useful in making them. The state and action representations vary greatly from application to application and affect the choice of algorithms chosen to solve them. There have been many reinforcement learning algorithms that have been developed to solve all kinds of problems [62]. Fittingly, reinforcement learning algorithms should be categorized by the problems rather than the methods. The simplest of problems are those that have a discrete number of state and actions. There are also those that have continuous state and action spaces. Algorithms that solve problems with continuous state spaces are known as connectionist algorithms.

The use of a *reward* signal to formalize the idea of a goal is one of the most distinctive features of reinforcement learning. Although this way of formulating goals might at first appear limiting, in practice it has proven to be flexible and widely applicable. The best way to see this is to consider an example of how it is used naturally. In the same example of a child learning to use his fork, an evaluation (reward) is given each time the child attempts to do something with his fork. In learning how to twirl the spaghetti, the child may see no reward until he finally succeeds in twirling the spaghetti, when the satisfaction of accomplishing something is a positive reinforcement signal. A more intelligent child might be able to associate intermediate steps like having the spaghetti slip from the fork by turning the fork the wrong way with a negative signal or seeing more spaghetti get twirled as a positive sign to achieving a higher goal.

We see that for an agent to learn something, rewards must be provided to it in such a way that in maximizing them the agent will also achieve its goals. It is critical that the rewards we set up for the agent’s feedback truly indicate what we want accomplished. Therefore, the reward function used for evaluating action responses to a state is the key to proper learning.

4.2.2 Advantages of Using Reinforcement Learning

The main advantage of reinforcement learning is that the framework allows us to relate a wide range of state, action and reward representations in a manner that suits the problem's abstraction and goals. This is advantageous in problems where the action may not directly affect the state in a deterministic manner, or where the goal of the problem may not have a known relation with the state or action. The reinforcement learning framework allows us the freedom to forgo complex analysis of state-action-goal relations and even to do away with any form of modeling if need be. In section 3.4, it was seen that the bandwidth provisioning problem is a hard problem involving continuous time, stochastic traffic patterns, and complex interactions of QoS mechanisms. This is the key motivation for selecting reinforcement learning algorithms to solve the bandwidth provisioning problem.

The freedom of abstraction allows partial relations between state and actions. In other model-based methods, there needs to be complete state representation relating to the actions. The freedom of abstraction also allows optimization of objectives such as revenue, which is on a different management plane from control mechanisms. Finding closed-form relations for such abstractions is often difficult. Without the need for modeling, changes in objectives can easily be implemented in the reinforcement learning framework.

Furthermore, reinforcement learning using associative (input-output mapping) and connectionist techniques, combines the advantage of pattern recognition through Neural Networks [63] and the ability to relate patterns to actions. The use of neural networks also offers some degree of prediction, allowing actions to be taken for future states based on current patterns. Another advantage of connectionist techniques is that they are able to generalize learnt state-action policies. These advantages are ideal for the continuous nature of the bandwidth provisioning problem.

4.2.3 Application of Reinforcement Learning in Network Control

Reinforcement Learning (RL) is not new to the networking community. There have been much success in applying RL to network control problems like admission control, routing, flow control, channel allocation, adaptive marking and path selection.

Marbach [64,65] first worked on the application of RL to call admission control and routing Integrated Services networks. The objective was to maximize revenue by controlling the number of calls admitted per unit time. Since the problem is too complex to allow for an exact solution, RL was naturally used. A feature-based state representation was used where the inputs were the number of calls of each class admitted on each link. The action was to decide whether to admit the call and on which route the call should traverse. Tong [66-67] followed up the work by including QoS constraints in a multimedia network. This placed constraints on the solution space. Q-Learning [68] was used to solve the problem.

In the area of flow control, Atlasis [69] used RL to modify the leaky-bucket. Due to the statistical nature of burstiness, policing sources can be very complex. The LB-SELA algorithm proposed makes use of RL to learn the behavior of the source, so that the leaky-bucket can be optimally tuned to result in better statistical gain and QoS guarantees.

RL has also been used to solve channel allocation problems in cellular networks. In Singh's work [70], NDP was used to re-allocate channels such that the reused channels are sufficiently far apart. The aim was to minimize the number of blocked calls while having a high channel reuse. The problem had a very large state space and therefore was infeasible to be solved by dynamic programming methods. Previously, only heuristic methods that ignored the optimal control framework could be used. However, it was shown that the problem could be optimally solved using an NDP framework. Further works on dynamic channel allocation have since been done by Nie [71] and Senouci [72]. Senouci recently solved the problem under QoS constraints for call-blocking in multi-class cellular networks. He included call admission control in order to maintain QoS and changed the objective to maximizing revenues.

In work that our group has done, RL successfully been applied to adaptive marking in DiffServ networks [73]. Q-Learning was used to re-mark the packets entering a DiffServ (DS) domain. The objective was to minimize transmission costs by re-marking a packet as a lower class. This however has to be done whilst QoS constraints were still met. RL was also applied to path selection in MPLS networks [74]. Paths were selected such that new call blocking due to insufficient bandwidth was minimized. It was shown that RL is able to allocate paths better than the commonly-used widest-shortest-path (WSP) method.

All of the above works surveyed have applied RL to network control problems that have discrete solution spaces. A few have used function approximation to reduce intractable state spaces. However, none so far have used connectionist RL methods to solve network control problems with continuous state and action spaces. In reality, problems with continuous spaces are usually the problems that are intractable due to the infinite number of states and actions possible. Heuristic methods are usually used to solve such problems, but they are often done without the notion of optimality. In this thesis, work is presented on the use of RL in continuous state-action spaces. It is hoped that this work would serve as a model for other network control problems in continuous space to be solved using continuous space RL.

4.3 CONTINUOUS STATE-ACTION SPACE REINFORCEMENT LEARNING

One of the fundamental requirements in reinforcement learning is that the information learnt can be stored. The problem in reality is that a wide class of control problems has states and actions that must be described using real-valued variables. In this class of problems, an agent must select an action from a continuous range of values after every fixed time interval while basing its decision using the currently perceived state, which is also one of infinitely many possible states. This poses a problem that is not seen in discrete spaces. In discrete spaces, RL algorithms often use a lookup table to store state and action variables [65]. However, this is not feasible in continuous spaces.

Another problem is that every experience that the agent goes through is unique. This means that the agent must be able to generalize what it has learnt in the past and apply it to future situations that it will face. Connectionist RL algorithms should therefore be able to handle real-valued variables as inputs, precisely map the state-action pairs to their assigned values, use memory resources efficiently, support learning without too much computational burden, and generalize the immediate outcome of specific state-action combinations to other regions of the state and action spaces.

A common approach that has been used to generalize continuous spaces it to quantize the state and action spaces into a finite number of cells and aggregate all states and actions within each cell [75]. This is one of the simplest forms of generalization in which all the states and actions within a cell have the same value. Thus, a lookup table can still be used. However, there is a compromise between the efficiency and accuracy of this class of tables that is difficult to resolve at design time. In order to achieve accuracy, the cell size should be small to provide enough resolution. But as the cell size gets smaller, the number of cells required to cover the entire state and action spaces grows exponentially, which causes the efficiency of the learning algorithm to deteriorate because more data is required for the approximation.

Another better approach is to avoid the problems associated with quantizing the state space altogether by using other types of function approximators, such as neural networks, that do not rely on quantization and can be used to generalize across all states [76-78]. The approach however is limited to associating one function approximator to represent all the states and one specific action. This is insufficient when the action space is also continuous, though the action space can be quantized. The next sections introduce two types of algorithms that are able to generalize over both state and action spaces. These algorithms are used in the bandwidth provisioning algorithms that will be introduced later in the thesis.

4.3.1 REINFORCE Algorithms

A framework was proposed by Williams [79] in which stochastic learning automata units were used in multi-layer connectionist network. A class of learning algorithms called REINFORCE algorithms were derived, which enable parameter updates to be made in a way that stochastically hill-climbs a performance measure in reinforcement learning problems. He further defined the *stochastic semilinear unit*, whose output y_i is drawn from some given probability distribution whose mass function has a single parameter p_i, which is in turn computed as

$$p_i = f_i(s_i), \tag{4.1}$$

where f_i is a differentiable squashing function i is the iteration step number and

$$s_i = w^{i^T} x^i = \sum_j w_{ij} x_j , \tag{4.2}$$

the inner product of w^i and x^i. x^i and w_{ij} are the inputs and weights on the input lines to this unit. The probability density function for y_i is expressed as

$$g_i\left(\xi, w^i, x^i\right) = \Pr\left\{y_i = \xi \middle| w^i, x^i\right\}, \tag{4.3}$$

emphasizing its dependence on the weights of the unit and its input vector.

In a reinforcement learning problem, the objective of the learning system is to learn to respond to each input pattern $x^i \in X$ with the action $y^x \in Y$, where y^x is such that $E\left\{r \middle| x^i, y^x\right\} = \max_{y \in Y}\left(E\left\{r \middle| x^i, y\right\}\right)$. In the REINFORCE algorithm, the original expected value of the reward $E\left\{r \middle| x^i, y^x\right\}$ is transformed to $E\left\{r \middle| W\right\}$, where W, the weight matrix, gives the associative input-output mapping. Note that the use of expected values here takes into consideration the randomness of the inputs, outputs and reward. The task of the reinforcement learning system is then to search the space of all possible weights matrices W for a point where $E\left\{r \middle| W\right\}$ is maximized.

Williams defines the class of algorithms that are of the following form as *REINFORCE* algorithms. The learning algorithm updates the parameter w_{ij} at the end of each trial after receiving immediate reward r as follows

$$\Delta w_{ij} = \alpha_{ij}\left(r - b_{ij}\right)e_{ij} , \tag{4.4}$$

where α_{ij} is a *learning rate factor*, b_{ij} is a *reinforcement baseline*, and $e_{ij} = \partial \ln g_i / \partial w_{ij}$ is called the *characteristic eligibility* of w_{ij}. The reinforcement baseline b_{ij} is conditionally independent of y_i, given W and x^i, and the rate factor α_{ij} is non-negative and depends at most on w^i and t.

REINFORCE algorithms have the property (proven by Williams) that relates $\nabla_W E\{r|W\}$, the gradient in weight space of the performance measure $E\{r|W\}$, to $E\{\Delta W|W\}$, the average update vector in weight space, in such a way that $E\{\Delta W|W\}$ lies in a direction for which $E\{r|W\}$ is increasing. This means that for each weight w_{ij}, the quantity $(r - b_{ij})\partial \ln g_i / \partial w_{ij}$ represents an unbiased estimate of $\partial E\{r|W\} / \partial w_{ij}$.

An extension of the single-parameter output distribution is to draw the output from a multi-parameter distribution. In particular, when the output is drawn from a Gaussian distribution, with parameters μ and σ individually controllable by separate weights, control over σ is tantamount to control over the exploratory behavior.

For such a Gaussian unit, the real-valued output y has a density function g given by

$$g(y, \mu, \sigma) = \frac{1}{(2\pi)^{1/2}\sigma} e^{-\frac{(y-\mu)^2}{2\sigma^2}} , \qquad (4.5)$$

The characteristic eligibility of μ is then

$$\frac{\partial \ln g}{\partial \mu} = \frac{y - \mu}{\sigma^2} , \qquad (4.6)$$

and the characteristic eligibility of μ is then

$$\frac{\partial \ln g}{\partial \sigma} = \frac{(y-\mu)^2 - \sigma^2}{\sigma^3}, \tag{4.7}$$

A REINFORCE algorithm for this unit thus has the form

$$\Delta\mu = \alpha_\mu (r - b_\mu) \frac{y-\mu}{\sigma^2} \tag{4.8}$$

and

$$\Delta\sigma = \alpha_\sigma (r - b_\sigma) \frac{(y-\mu)^2 - \sigma^2}{\sigma^3}, \tag{4.9}$$

where α_μ, b_μ, α_σ and b_σ are chosen appropriately.

We see that equation (4.8) adjusts the bias of the output y and equation (4.9) adjusts the amount of exploration away from μ. The terms $(r - b_\mu)$ and $(r - b_\sigma)$ indicate how good the last action was with respect to the reinforcement baselines b_μ and b_σ.

Gaussian units are simple to use and are efficient in finding an optimal action. However, they lack the ability to respond to different input states, as μ is not determined by x. Therefore, Gaussian units are only capable of reacting to changing states through the reinforcement signal r. This may be a good thing as a new state may not require much of a change in the action to achieve a high value of r. In fact, by choosing μ to be independent of x, we can reduce the volatility of the system by slowing the response to changing states. This feature is especially useful when we use it to provision for the UMTS core network, where stability is more of an issue. In the next section, another algorithm is described that takes into consideration the input state when deciding the output action to take. This would enable faster response to changing states.

4.3.2 Stochastic Real-Valued Units

The Stochastic Real-Valued (SRV) unit introduced by Gullapalli [60] is a variation of the REINFORCE algorithm. In fact, the SRV unit is actually an extension of the Gaussian unit. The SRV unit introduces a parameter θ that relates the state x to the mean μ. The relation is represented by a simple inner product

$$\mu = {\theta_n}^T x_n, \tag{4.10}$$

where n is the iteration step number.

Another difference between the SRV unit and the Gaussian unit described above is that σ is not a parameter belonging to W. Gullapalli decided that σ should be directly related to the baseline reinforcement $\hat{r}$ through a squashing function

$$\sigma_n = s(\hat{r}_n), \tag{4.11}$$

and that the baseline reinforcement should be related to the state x by some parameter ϕ through a simple inner product

$$\hat{r}_n = {\phi_n}^T x_n, \tag{4.12}$$

However, this parameter is not updated according to the REINFORCE algorithm. Instead, the parameter ϕ is updated using the LMS rule of Widrow and Hoff [81] given by

$$\Delta\phi_n = \rho\left(r(y_n, x_n) - \hat{r}_n\right)x_n, \tag{4.13}$$

where $\rho > 0$ is a learning rate parameter.

The update of the parameter θ is similar to equation (4.8) and is given by

$$\Delta\theta_n = \sigma_n(r - \hat{r}_n)(y_n - \mu_n)x_n. \tag{4.14}$$

Notice that the state x_n is now included in the update and that $\alpha_\theta = \sigma^3$ is used.

SRV units improve over Gaussian units by relating the current state observed and the bias μ of the action taken. Furthermore, the amount of exploration is directly controlled by the baseline reinforcement and is not a parameter that needs to be learnt. This implies that as the baseline reinforcement becomes more favorable, the amount of exploration is proportionally limited. In Gullapalli's algorithm, this is effective because reinforcements range between 0 and 1; 1 being the optimal reinforcement. However, in many cases the maximum reinforcement is unknown. Although the reinforcement baseline is learned using the LMS rule, it is difficult to gauge the actual goodness level. In the bandwidth provisioning algorithms that are to be presented later in the thesis, some portion of William's REINFORCE algorithm is used to improve the SRV algorithm.

4.4 REINFORCEMENT LEARNING FORMULATION OF BANDWIDTH PROVISIONING OPTIMIZATION PROBLEM

In section 3.4, the bandwidth provisioning problem was formulated as an optimization problem. It was seen that the problem was hard to solve due to the continuous and random nature of its variables. The reinforcement learning framework allows us to "learn" the solution to hard problems through iterative improvements. The bandwidth provisioning problem is re-formulated in a reinforcement learning framework as follows:

At each node,

Given,

$x_{EF,j}{}^{i}$, $x_{AF,j}{}^{i}$, $x_{BE,j}{}^{i}$ $\in x^{i}$, for all $j \in$ all outgoing links from node

where $x_{EF,j}{}^{i}$, $x_{AF,j}{}^{i}$ and $x_{BE,j}{}^{i}$ are the traffic rates of each class entering the node destined to leave through link j, and x^{i} is the set of all input traffic rates at every node for the i^{th} iteration.

Select,

$w_{EF,j}{}^{i}$, $w_{AF,j}{}^{i}$, $w_{BE,j}{}^{i}$ $\in y^{x}$

where $w_{EF,j}{}^{i}$, $w_{AF,j}{}^{i}$ and $w_{BE,j}{}^{i}$ are the weighted fair proportions of bandwidth for each class on link j and y^{x} is the set of all actions selected at every node based on the input x^{i}.

Such that,

$$E\{r|x^{i}, y^{x}\} = \max_{y \in Y}\left(E\{r|x^{i}, y\}\right) \tag{4.15}$$

where r is the net revenue (gross revenue minus QoS penalties) earned depending on the pricing plan and SLA contracted, which spell out the revenue earned for provisioning each class of service and the penalties paid out if any of the QoS requirements are breached.

The main difference between the RL-based formulation and the linear programming formulation is the approach taken. Whilst the linear programming formulation requires an optimal solution to be derived continuously at every time instance, the RL-based formulation (which is in fact a dynamic programming [61] formulation) is an iterative formulation that seeks *a* solution (policy) that achieves a long term goal. The problem with the linear programming method is that it is intractable to determine the optimal solutions at every time instance. The advantage of the RL-based formulation is that a policy (one that achieves objective (4.15)) can be developed through reinforcement learning algorithms that would, in the long run, be close to optimal. This would circumvent the problems of continuous time, traffic predictability and complex analysis of network topology and QoS mechanisms, which render the original formulation intractable.

CHAPTER 5

REINFORCEMENT LEARNING-BASED PROVISIONING FOR CORE BACKBONE NETWORK

5.1 INTRODUCTION

In this chapter, we discuss the provisioning of UMTS core backbone networks. The UMTS core backbone network has characteristics that are different from fixed backbone networks. Firstly, UMTS core backbone networks are managed by service providers, usually mobile telecommunications providers. Whereas, fixed backbone networks are managed by carriers. The main function of carrier networks is to transport data at high speeds point-to-point, either between two geographical locations or between two service provider networks. Service providers on the other hand operate a distributed network providing numerous multi-point heterogeneous connections to all their customers within their topologically large network. Therefore, the provisioning of high-speed fixed backbone networks has very different conditions when compared to the provisioning of UMTS core networks.

Secondly, QoS management is critical in UMTS core networks due to multiple levels of services provided over a large and diverse topology, whereas, QoS management in fixed backbone networks is relatively straightforward, requiring guarantees only from point-to-point. Thirdly, traffic patterns in UMTS core networks are more diverse than in fixed backbone networks, which transport traffic that is highly aggregated and has longer timescales. Lastly, due to the transport of UMTS data and voice services over the network, traffic in UMTS core networks would have very different traffic patterns and requirements from that of current core networks of data service providers and mobile network operators. Such existing core networks either carry only data traffic or only voice and circuit-switched data traffic. The provisioning of these types of networks is much less complex as compared to future converged networks.

In the following section, current methods of adaptive provisioning are reviewed and reasons why they are inadequate for UMTS core networks are given. A solution that is capable of use in UMTS core networks is then proposed.

5.2 CURRENT METHODS OF ADAPTIVE BANDWIDTH PROVISIONING

There have been few bandwidth provisioning proposals in the literature specifically for UMTS core networks. However, the area of bandwidth provisioning for fixed networks has been long researched. Bandwidth provisioning methods include measurement-based admission control methods, adaptive control methods, methods that use traffic predictors and methods that solve the bandwidth resource problem through the use of pricing strategies. Some of the methods proposed are simply for estimating or predicting bandwidth requirements. Others are capable of provisioning for multiple classes in Integrated Services (IntServ) [82] networks or in DiffServ networks. The methods surveyed also have different QoS criteria. Some methods make use of cell-loss ratio (CLR) and blocking probability in the Asynchronous Transfer Mode (ATM) [83] and IntServ context, while other methods use latency and packet loss in the DiffServ framework.

5.2.1 Measurement-based Admission Control Methods

Admission control is a means to control network congestion and maintain QoS. The decision to admit a flow is based on the effective bandwidth already utilized by admitted traffic. Due to multiplexing, the effective bandwidth utilization of an aggregated flow can be much lower when there is greater aggregation. The determination of the effective bandwidth is however a difficult task and has been studied by many researchers. The task is made all the more difficult when there are multiple classes of traffic and each class of traffic has QoS requirements that must be maintained. Measurement-based admission control methods have been popular due to its model-free approach. Jamin [84], Kelly [85] and Knightly [86] have all contributed significantly to the development of various methods for estimating the effective bandwidth. The algorithms have mainly been focused on IntServ networks.

Jamin [87] has commented that measurement-based admission control methods, while being effective in improving bandwidth utilization while maintaining a reasonable service level, cannot guarantee any quantitative QoS level. Therefore, the use of such methods for provisioning bandwidth can only be used for loose QoS requirements. However, UMTS core networks are required to carry conversational and streaming classes of traffic, which have strict QoS demands.

Lately, measurement-based admission control methods have also been developed for DiffServ networks and VoIP networks [88,89]. Oottamakorn [88] showed that by using effective envelopes and service curves, the delay bound can be guaranteed for an aggregate class. Although the use of service curves can provide guarantees, this is usually at the expense of some loss in utilization. Mase [89], on the other hand makes use of adaptive control to adapt a parameter that determines the level of admission control. If a stricter level of QoS is required, the parameter can be made to be more conservative. Adaptive control is a much better means of achieving QoS.

In general, the use of admission control methods to provision bandwidth leads to less than full utilization. This is because the effective bandwidth computation is always conservative, especially when strict QoS is required. Furthermore, traffic is assumed to be multiplexed. Without any class-based scheduling, bandwidth cannot be guaranteed.

5.2.2 Adaptive Control Methods

The use of adaptive control methods to adjust weighted fair proportions is a straightforward and effective way to maintain QoS. By using indicators such as the average queue length, traffic intensity and QoS feedback like cell loss ratio, delay and jitter, adaptive controllers can adjust weighted fair proportions to maintain these indicators within a required range, so that QoS requirements can be met.

Various types of adaptive controllers can be used to different effects. Chandramathi [90] and Siripongwutikorn [91] have proposed the use of fuzzy controllers using QoS feedback and average queue length as inputs respectively. Fuzzy control makes use of some expert heuristics to control bandwidth provisioned. However, without pre-knowledge or experience of the network dynamics, it is difficult to establish the right utility functions to use. This is evidenced by the need to have various sets of functions for different networks as well as for different QoS requirements. Chou [92] presented an adaptive controller that makes use of a Genetic Algorithm (GA) to judge which bandwidth provisioning settings have been effective in achieving high utilization given a certain traffic load. A neural network was used to adjust the parameters of the GA agent. The controller was used to minimize bandwidth changes while keeping utilization high and not for meeting QoS though.

There are adaptive controllers that make use of more direct ways to control bandwidth provisions. Wang [93] in his paper describes a way controlling the level of provisioning using a similar principle to that of the well-known Random Early Detection [94] algorithm. The average queue length is used to determine whether thresholds have been exceeded. If they have, the bandwidth provisions are adjusted accordingly. The method was shown to be effective in improving QoS, but did not have the ability to maintain a target QoS level. In another work by Liao [95], an optimal control method was proposed that can achieve a targeted level of QoS, both in terms of delay and loss. Bandwidth provisions are adjusted based on the target traffic intensity as a target control value and the measured traffic intensity as a feedback signal. Whenever there is an overloaded or underloaded traffic condition, the weighted fair bandwidth proportions are adjusted. The delay requirement is met by limiting the queue length and the loss requirement is met by setting the target traffic intensity as a function of the loss requirement. Though this method was shown to be highly effective, bandwidth provisioning through control feedback methods is at best reactive in nature. If the convergence time is longer than the traffic fluctuation cycles, the controller may not be able to reach steady state each time. A method that is capable of proactively (rather than reactively) provisioning bandwidth is required.

5.2.3 Traffic Prediction Methods

A way of proactively provisioning bandwidth is to predict the traffic and decide how best to provision for future traffic such that QoS can be met and utilization can be maximized. We review some methods in the literature that are capable of predicting multimedia traffic. However, it should be noted that even if traffic can be predicted, the task of provisioning traffic to maintain certain levels of QoS for all classes is a difficult one. This is due to the lack of good methods to load balance the different traffic classes during congested periods. (Load balancing, or sometimes known as traffic engineering, usually involves the task of re-routing portions of traffic to other routes). The need to provision the right "mix" (the proportion of bandwidth allocated to each class) of traffic on each link is a vital part of maintaining QoS levels in converged networks, such as UMTS core networks. Nonetheless, traffic prediction methods have been popular as it gives good foresight to network managers.

Sahinoglu [96] has proposed the use of a novel wavelet-decomposed signal energy approach to characterize traffic arrival rates in order to predict near future behavior. The algorithm presented by the author however aims to improve average queue length and bandwidth utilization for a homogeneous link. Hence, the algorithm may not be effective on heterogeneous links. Moreover, the algorithm has no notion of QoS. Another prediction method based on FARIMA models was demonstrated by Ilow [97]. It was shown that the method was capable of predicting both short term and long term traffic patterns.

A prediction method proposed by Gallardo [98] takes into account QoS. A modified leaky bucket is used together with their linear prediction algorithm to control the service rate of each class. In his paper, Gallardo challenged the effectiveness of effective bandwidth methods in maintaining QoS. He demonstrated his method in DiffServ networks and showed that his scheme was able to optimize network resources and thus minimize the probability of violating QoS contracts. Although such a method is effective, the application of rate control to provision bandwidth, although better than admission control in terms of bandwidth utilization, is still not as effective as provisioning bandwidth through a weighted fair queuing scheduler.

5.2.4 Pricing Methods

Another way of provisioning bandwidth that is popular lately is through pricing. This method is a macro-level method, unlike micro-level methods that solve QoS provisioning solely at the network layer. For micro-level methods, the SLA is first negotiated at a business level, without much consultation at a network level. Once an agreement as been penned, the service provider then provisions his network to meet this SLA. Therefore, the SLA influences the provisioning, but not the other way around. What pricing methods do is to introduce the influence of provisioning on the SLA. This two-way method would involve major changes in business frameworks as market competition (through bidding) becomes a two-way affair. Service providers would be able to offer alternative levels of service on-the-fly depending on the network condition. This greatly departs from long term contractual agreements commonly seen today. Nonetheless, it is a probable and possibly viable way of provisioning bandwidth in the future.

Semret's work on pricing, provisioning and peering [99] is the pioneer in this field. Semret proposed the use of a decentralized auction-based approach to allocate bandwidth at the edges of DiffServ networks. To optimize the allocation of bandwidth and service levels, a game-theoretic method was used. In recent work, Malinowski [100] proposed a pricing method for flow control and Garg [101] introduced a three-tier pricing model with penalties that allows customers to dynamically change their demands and encourage better utilization of resources.

Pricing models can be tiered according to different levels of service offered (SLA), because customers have different requirements are thus willing to pay differently. The use of multi-tiered pricing models to influence the provisioning of bandwidth for different classes is a good objective-based approach. By provisioning with relation to not only the QoS requirements contracted but also with relation to the pricing of services, a bandwidth provisioning policy that satisfies both customers and service providers can be achieved. This idea is used in the following solution that we propose.

5.3 REINFORCEMENT LEARNING ADAPTIVE PROVISIONING BASED ON REVENUE MAXIMIZATION

When a customer requires a mobile service to be provided to him, an SLA (service level agreement) is contracted between the customer and the mobile service provider. This contract binds various aspects of the level of service that the mobile service provider has to provide for the customer. The SLA would include details about the QoS specifications. These are found in the service level specifications (SLS) [29] portion of the SLA. The SLS specifies requirements that are pertinent to the service provided. Usually, the service provider offers several standard SLS for the mobile services they provide.

In these SLS, delay requirements are specified in terms of an upper bound, where each packet is to have end-to-end delay less than the value specified. Throughput requirements are specified in terms of a lower bound, where the rate of traffic delivered measured at regular intervals is to be greater than the value specified. Packet loss requirements are specified in terms of the packet loss ratio. All these QoS requirements are what customers expect from the service provider. However, for most applications that mobile customers would be using, like video streaming and web-browsing, occasional service lapses can be tolerated to a certain level. Only mission-critical applications like emergency retrieval of medical records require absolute guarantees. These services form a small percentage of traffic volume and can be provisioned separately using some priority-based scheme. For most of the other applications, customers would not mind paying lower prices for less-than-guaranteed services. This is especially attractive to the customer if he has a higher level of tolerance.

5.3.1 Multi-tiered Pricing Strategy

Since QoS is closely tied to the customer's demands and a customer's demands may be influenced by the price he is willing to pay, it is important to develop a pricing strategy that benefits both the customer and the service provider. There are a few ways of pricing a service, and the method of pricing is linked very much to the method of provisioning used. Often, users are charged a fixed price per month for unlimited time usage. This is sometimes backed by a bandwidth guarantee, as service providers may treat the service like a leased-line service. Service providers use bandwidth partitioning to provision for such services. This type of charging benefits users who have a high usage pattern, but for the majority, this unlimited level of service is not needed. For most of the time, the bandwidth is left idle. This also does not benefit the provider as the unused bandwidth can be used to generate more revenue. However, for such service provisioning, QoS is easily guaranteed as bandwidth is partitioned in a way that would usually be more than user's need (partly due to the denominations that bandwidth is offered in).

Another way of charging users is by time usage. This is a good way of charging for circuit-switched services. In current 2nd Generation mobile networks and ATM-based data networks, service providers can multiplex users based on erlang computations; the users are assumed to fully utilize the bandwidth for the time they are connected. Again, many customers do not require the full bandwidth when they are connected. Often the data traffic pattern is bursty and service providers make use of effective bandwidth methods to statistically multiplex a greater number of users. For such services, QoS may not be so easily guaranteed, but often it can be assured to a high level, if the multiplexing is done in a conservative way. However, being conservative would mean less bandwidth utilization.

The best way of charging users (in an economic sense) is to do it based on data usage. Since packet-switched services came into the market, some service providers have offered such services. An amount is charged for each kilobyte of data transmitted. In this way, users are charged exactly for the amount of bandwidth used. This could mean great savings for users. Although high-usage customers could benefit from unlimited bandwidth services due to a flat fee, usage-based pricing may be competitively implemented in such a way that gives attractive rebates to high-usage customers. Thus, such usage-based pricing benefits both low and high usage customers. It also benefits service providers, because bandwidth utilization can be increased with greater aggregation. The drawback to this is that QoS is difficult to achieve.

To solve this issue, a penalty-based approach is proposed. Penalty refunds are not new to service providers, who offer refunds for any QoS breeched in the SLA. Extending this concept, a service provider may offer to refund based on the amount of data transmitted that breeched the QoS contracted. The penalty may be valued at double the price charged for transmitting the data. For example, if the price charged is $0.001 per kilobyte transmitted; then the penalty could be $0.002 per kilobyte of data that had QoS breeched. That means if during a transmission session, 10 packets of 10 kilobytes out of 100 packets were dropped or delayed during transmission, then the amount charged would be $0.80. This approach acts as a disincentive for service providers to breech the service level contracted, while at the same time providing some room for occasional lapses, which the user may not mind as long as he is adequately compensated.

The provisioning of multiple services on the same converged network requires a multi-tiered pricing model. As the level of QoS demanded increases, users should be made to pay higher rates. This would also be equally complimented by higher penalty rates, since a breech of such high value services would usually be less tolerable. For example, conversational and interactive classes of traffic that use the EF DiffServ class may be charged at a higher price, while streaming class traffic that use the AF class may be charged at a lower price. This has implications to the bandwidth provisioning strategy. An intelligent bandwidth provisioning scheme would take into account the pricing plan and SLS contracted, in order to maximize the revenue earned by the service provider. This may mean provisioning more for the AF class if it is more profitable than the EF class for example. The next section proposes such an intelligent bandwidth provisioning algorithm that is able to learn a policy that maximizes revenue.

5.3.2 RLAP Algorithm

The concept behind the proposed Reinforcement Learning-based Adaptive Provisioning (RLAP) scheme is that an SLS tied to the proposed multi-tier pricing model described above can be used as a feedback mechanism to determine a policy that maximizes revenue. Reinforcement Learning (RL) can be used to adaptively adjust bandwidth provisioning for each DiffServ class of traffic. The reward function used to compute the reinforcement feedback is constructed as the amount of net revenue earned based on the pricing plan contracted. The RL agent adaptively adjusts weighted fair proportions in each DiffServ router at a regular interval based on the feedback of how much revenue was generated in the last interval. Since the provisioning based on current traffic conditions affects the QoS experienced and the revenue earned, the action-reward forms a closed-loop as illustrated in Fig. 5.1.

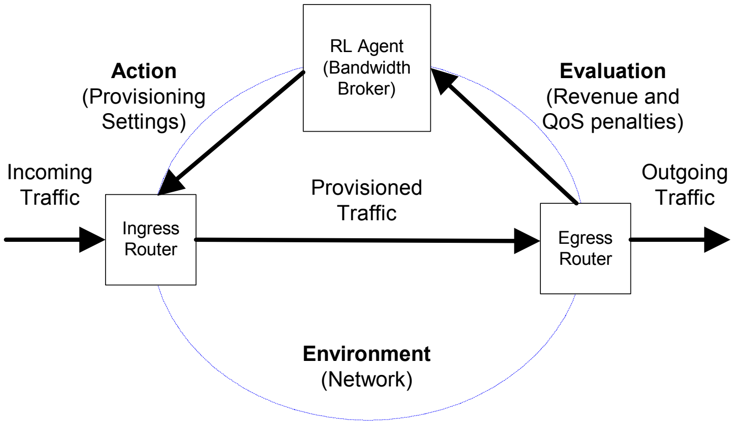

Figure 5.1: Reinforcement Learning Loop in RLAP

When RLAP is applied to the UMTS core backbone network, it can be used in the following scenarios to improve provisioning:

1) Changing traffic conditions – due to changing traffic intensities of EF and AF traffic, EF and AF weights need to be adjusted accordingly. Under-provisioning may lead to high delays and low throughput. On the other hand, over-provisioning may not be necessary when traffic intensity is low.

2) Strictness of SLS – depending on the delay and throughput bounds and the level of congestion in the network, weights can be changed to reflect how critical these bounds are. The lower the bound and the higher the traffic intensity, the greater should be the weight.

3) Different pricing plans – by changing the reward function, weights can be changed in favor of the traffic that generates more revenue. If penalties are not severe, weights can be lowered to take advantage of revenue from other traffic, allowing the occasional penalties.

To describe how RLAP can be implemented in a DiffServ network, a one-domain topology in Fig. 5.2 is used here as an example, and is made as simple as possible without loss of generality. Sources S_0 to S_7 have destinations D_0 to D_7 respectively. R_1 and R_2 are ingress edge routers and R_5 and R_6 are egress edge routers. These routers could represent either GGSNs or SGSNs depending on the direction of traffic. R_3 and R_4 represent core routers in the UMTS core backbone network. While only a single direction is considered here, the scheme works similarly for bi-directional traffic. In general, traffic flows may enter from any edge router and exit through any other.

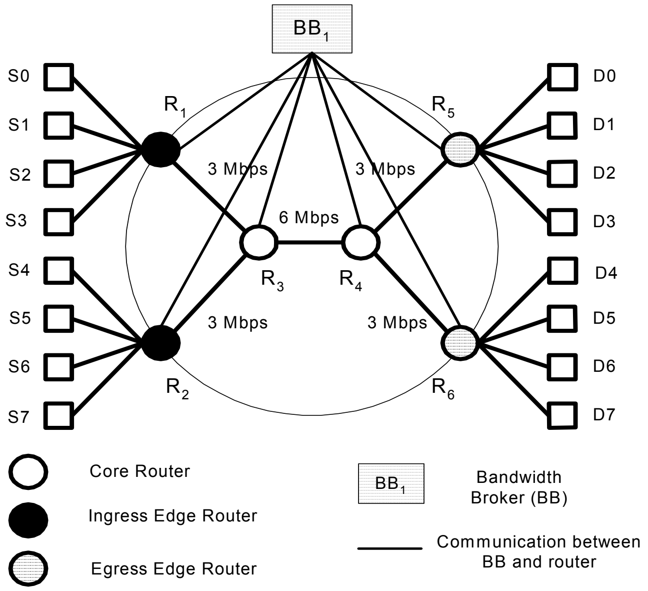

Figure 5.2: DiffServ Network Topology

A bandwidth broker BB_1 is used as the centralized collection and decision-making point. A similar framework to the Clearing House Architecture proposed by Chuah [102] is used. RLAP is used instead for determining bandwidth provisioning. RL agents can either be placed in each router, or housed in the bandwidth broker with separate logical agents for each router. We chose the latter, to be in line with the framework. This design also requires almost no modification to existing routers in the framework to implement RLAP.

At regular intervals, BB_1 collects traffic measurements (in terms of number of bits) from all routers in the domain for computation of revenue. Concurrently, destination nodes report, via a system of accounting, any QoS violations in terms of amount of traffic delayed and number of intervals throughput was not met, to the bandwidth brokers along the paths of the flows (in a multi-domain case); BB_1 in our topology. BB_1 then makes decisions through the RL agents and sends the WFQ weight configurations to the respective routers. SLAs and pricing plans are stored in the database for computing the charges and penalties used as feedback to the RL agents. In a multi-domain scenario, bandwidth brokers may shift some of the functions up the hierarchy of bandwidth brokers. In return, decisions made are passed downwards towards the routers.

In implementing REINFORCE Gaussian units in the RLAP scheme; we have set the output y of the RL agent to be the WFQ weight settings that determine provisioning. The time step n is the n^{th} time interval since the RL agent started; each time interval being of duration T. At the end of each interval, the cumulated reward r, based on the evaluation of the weight settings for traffic conditions over the interval, would be used as feedback to adaptively adjust μ and σ of the RL agent, which are used to determine the next weight settings. For RL agent RL_i, y_i, μ_i and σ_i are vectors with elements $y_{i,j}$, $\mu_{i,j}$ and $\sigma_{i,j}$ for each PHB aggregate j.

For each agent RL_i, the RLAP algorithm is as follows (omitting the subscript i for clarity):

```
Initialize $\mu_0$ and $\sigma_0$.
At the end of every interval $n$,                    $n \geq 1$
update      $\hat{r}_n = \gamma r_n + (1-\gamma)\hat{r}_{n-1}$
            $\mu_{n+1} = \mu_n + \alpha_\mu (r_n - \hat{r}_n)(y_n - \mu_n)$
and         $\sigma_{n+1} = \sigma_n + \alpha_\sigma (r_n - \hat{r}_n) \frac{(y_n - \mu_n)^2 - {\sigma_n}^2}{\sigma_n}$
Select      $y_n \sim N(\mu_n, \sigma_n)$
```

The choice of initial values for μ_0 and σ_0 is an important issue. The effect of the choice of different initial values for μ_0 is simulated and discussed in the next section. For all real practical cases, we suggest that the choice of μ_0 be based on any limited *a priori* information available; for example, traffic specifications given by the user and past traffic measurements may provide average and peak throughput information, which can be used as the initial provisioning values.

The choice of σ_0 should be made such that a trade-off is struck between the smoothness of the exploration and the rate of convergence. Larger σ_0 values tend to provide better convergence to the global optimal solution; however, fluctuations in μ values may tend to be undesirably large. Smoothness of exploration is important during the real operation of the network. A possible way to solve this concern is to have a trial phase, where a large σ_0 may be used to train the RL agents, before actual (chargeable) network traffic is carried. We have not done this in our experiments and σ_0 was set to a rather low value. This was done intentionally to demonstrate the capability of RLAP under situations where such trial periods are not possible and large fluctuations in network settings are undesirable.

In any proposal that implements RL agents, convergence is always an important issue. In continuous RL methods, the assumption that the reward function has no discontinuity in the practical range and also be monotonic towards the global optimum must hold [77]. If not, such algorithms may be stuck in a local maximum. In our case, the pricing plan must have a monotonically increasing form, which is true for most cases. A related issue is the rate of convergence. In RLAP, this is controlled by the constants α_μ, α_σ and γ. By adjusting their values, a trade-off is made between the rate of convergence and the fluctuation in action values. A smaller constant would mean a slower rate of convergence, but a larger constant would increase the fluctuation in the WFQ settings.

Another parameter to consider is time interval T. It should be set up to balance the trade-off between the adaptability of the scheme and the frequency of disruption caused by network re-configurations. We prefer a larger interval as it causes less disruption to the network and also allows RLAP to be less susceptible to traffic irregularities, due to the averaging of traffic and reward/penalty measurements over longer periods.

5.3.3 Reward Function

We have emphasized in section 4.2.1 the importance of setting up a reward function that accurately reflects our goals. The goal of RLAP is to maximize the revenue earned by the service provider. This is achieved by provisioning bandwidth for each class of users such that revenue earned from carrying traffic is maximized without incurring too much penalty from QoS violations.

We make use of the proposed multi-tier pricing model as the reward function. Users are charge based on the amount of traffic carried on the network. As such, users pay only for what they use. A different price is charged for each service class and a penalty is imposed for each QoS criteria, contracted in the SLA, not met.

The reward function computes the reward r_i for RL agent RL_i in the following way:

$$r_i = \sum_j \begin{pmatrix} c_j \times t_{i,j} - p_{loss,j} \times l_{i,j} - p_{dly,j} \times d_{i,j} \\ - p_{thr,j} \times th_{i,j} \end{pmatrix} \tag{5.1}$$

For each PHB aggregate j, c_j is the charge per bit of traffic carried, $t_{i,j}$ is the amount of traffic forwarded by router i, $l_{i,j}$ is the number of packets loss at router i, $d_{i,j}$ is the amount of traffic that passed through router i not meeting delay QoS and $th_{i,j}$ is the number of intervals not meeting throughput QoS for traffic that passed through router i. $p_{loss,j}$ is the penalty per packet loss, $p_{dly,j}$ is the penalty per bit of PHB aggregate j not meeting delay QoS and $p_{thr,j}$ is the penalty per interval of PHB aggregate j not meeting throughput QoS.

5.3.4 Simulation and Results

5.3.4.1 Simulation Setup

Using *ns-2* [8] DiffServ extensions, the topology in Fig. 5.2 was set up to compare RLAP with static provisioning. We show that RLAP is able to improve even over commonly used over-provisioning methods (this requires the assumption that users are able to describe their usage, which is not always possible). Droptail queues were used for each class instead of the usual RIO (RED with In and Out packets) buffer management to separate the effects of RLAP from RIO. The buffer length for AF and BE traffic were set to 100, while the buffer length for EF traffic was set to 2 for the static provisioning case (as is the common practice to keep end-to-end delay low) and 10 for our scheme. We show that RLAP is able to accommodate more packets in the buffer, and yet not hamper end-to-end delay by much.

5.3.4.2 Traffic Characteristics

In our simulations, traffic from all 3 PHB aggregates were generated on all links in the DiffServ domain. The links were made to carry close to full capacity most of the time. Sources S_0 to S_3 have similar characteristics to sources S_4 to S_7 respectively. Table 5.1 summarizes the characteristics of the traffic sources.

TABLE 5.1: Characteristics of Traffic Sources

Source	Traffic Type	Connection Inter-arrival Time (s)	Connection Holding Time (s)	ON Rate (kbps)
S_0, S_4	EF	1.875	30	64
S_1, S_5	BE	7.5	60	128
S_2, S_6	AF	4.5	30	300
S_3, S_7	BE	1.775	30	128

S_0 is an EF source that represents delay bounded traffic like VoIP traffic. S_2 is an AF source that represents loosely delay bounded, high throughput requirement, traffic like video traffic. Both are modeled as exponential ON-OFF sources with same ON (500ms) and OFF (500ms) times. S_3 is a BE source that represents non-bounded aggregated web traffic. It is modeled as a pareto ON-OFF source with the same ON and OFF times as the EF and AF traffic. These 3 sources run over UDP. The last source S_1 runs over TCP, and is a BE source that represents non-bounded high throughput traffic like FTP traffic. It is a CBR source that consumes any unused capacity on the link. This enables the link to be fully utilized most of the time. Sources S_0 to S_7 have destinations D_0 to D_7 respectively. The average amount of generated traffic towards each ingress router for EF and AF are 500 kbps and 1 Mbps respectively. BE traffic utilizes the remaining amount of capacity; about 1.5 Mbps.

5.3.4.3 Experimental Details

In the static provisioning case, we set up the WFQ weights to be static throughout each experiment lasting 20,000s. For the RLAP case, we set up the experiment to initially begin with the same weights as in the static case. We also set μ_0 to be equal to the static setting. The RLAP scheme then kicks in after 1,500s, adjusting the WFQ weights in the routers adaptively and improving the policy in the RL agents through time. All simulations involving RLAP were run with 5 different random seeds. The mean and range is plotted wherever possible. For both cases, we measure performance only after an initial period of 5,000s for an additional 15,000s. The time-step interval T chosen is 500s. The choice of α_μ, α_σ and γ used are 0.00005, 0.00001 and 0.2 respectively.

Table 5.2 shows the pricing plan used. The charge for EF and AF traffic is 10 and 4 times the charge for BE traffic respectively. The penalties for packet losses and delayed EF traffic are double the charge. The penalties for AF traffic not meeting the delay or the throughput bound is equal to its charge, such that if AF traffic does not meet both requirements, it will be penalized double its charge. The delay requirements for EF and AF are 15 ms and 35 ms respectively, and the throughput requirement for AF is 200 kbps.

TABLE 5.2: Pricing Plan

c_{EF}	0.0001	$p_{loss,EF}$	0.2	$p_{dly,EF}$	0.2
c_{AF}	0.00004	$p_{loss,AF}$	0.08	$p_{dly,AF}$	0.04
c_{BE}	0.00001	$p_{loss,BE}$	0.02	$p_{thr,AF}$	10

TABLE 5.3: WFQ Weight Settings for Various Provisioning Strategies

Provisioning Strategy	**WFQ Weight Settings (EF:AF:BE)**
Under-provisioning	1:2:3
50% over-provisioning	3:4:5
Over-provisioning	1:1:1

5.3.4.4 Comparison under Different Initial Provisioning

In our first experiment, we compare how RLAP improves over various initial WFQ weight settings. Since RLAP only kicks in after 1,500s, the weights remain static for the initial period. After which, RLAP adjusts the weights from these initial values. We benchmarked RLAP against static provisioning, which maintains the WFQ weights throughout the experiment. For static provisioning, the different WFQ weight settings mean different provisioning strategies. Three strategies were tested. The under-provisioning strategy provisioned EF at the expected average traffic rate, i.e., 0.5 Mbps of bandwidth. The 50% over-provisioning strategy allocated 50% above the average EF traffic rate, i.e., 0.75 Mbps of bandwidth. The most commonly-used strategy, the over-provisioning strategy, which allocates EF with more than sufficient bandwidth to handle bursts, provisions EF traffic at twice the bandwidth. For all 3 strategies, AF traffic was provisioned at the expected average AF traffic rate, i.e. 1.0 Mbps, while BE was allocated the remaining of the capacity. It is to be noted that static provisioning settings can only be determined when the average or peak rate information is accurately available *a priori*. On the other hand, RLAP requires only an estimate as an initial point. For our experiment, the average rate of the traffic generated by simulation is assumed to be known in order to have a comparison. The initial weight settings are given in Table 5.3.

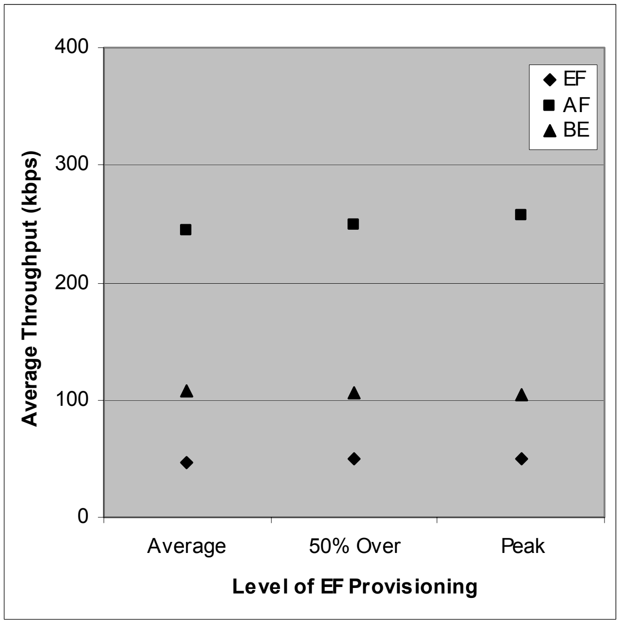

Figure 5.3: Average Throughput per Flow under Different Initial Provisioning for Static Provisioning

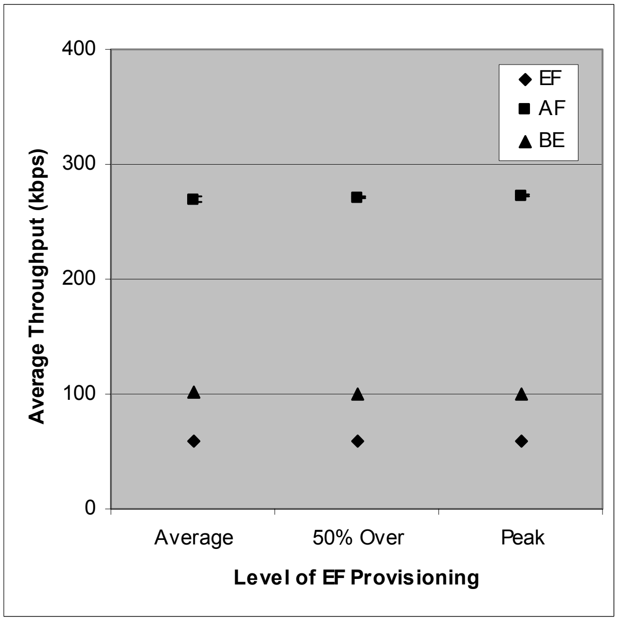

Figure 5.4: Average Throughput per Flow under Different Initial Provisioning for RLAP

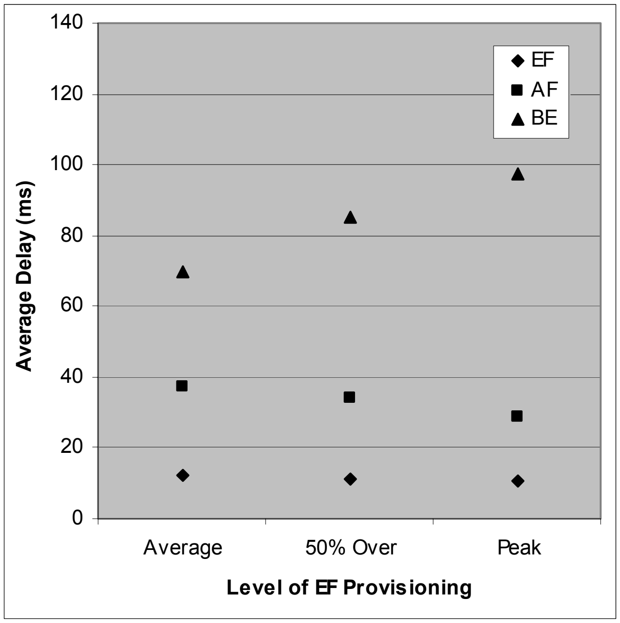

Figure 5.5: Average Delay under Different Initial Provisioning for Static Provisioning

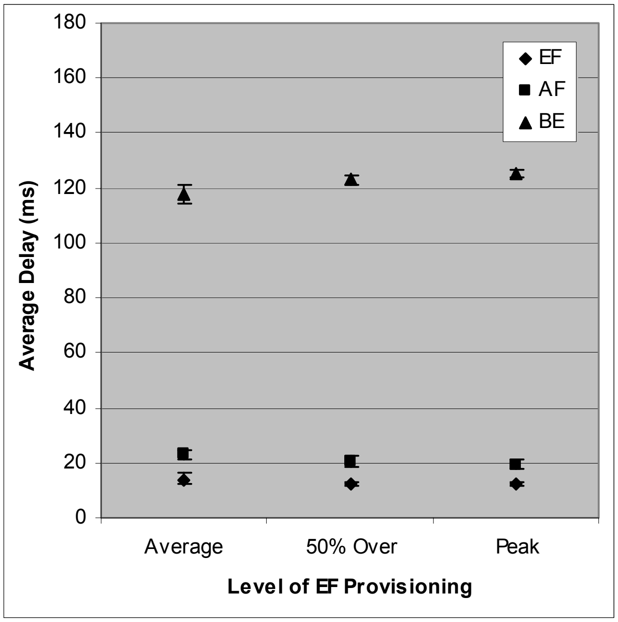

Figure 5.6: Average Delay under Different Initial Provisioning for RLAP

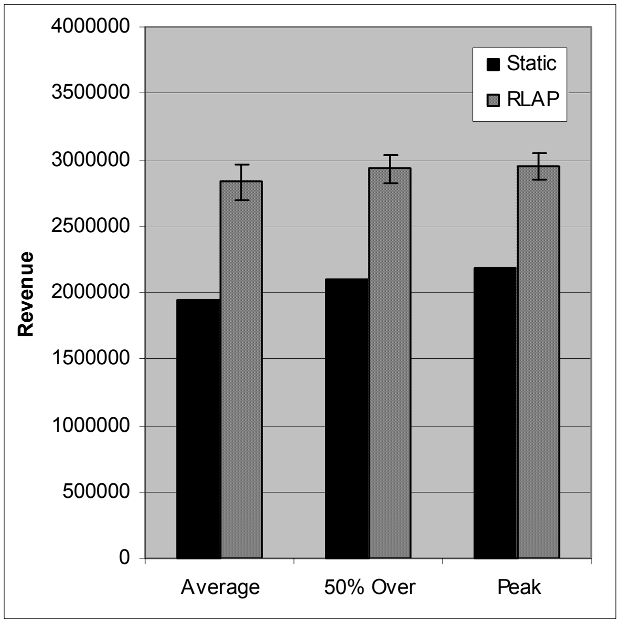

Figure 5.7: Revenue Comparison under Different Initial Provisioning

Fig. 5.3 and 5.4 show the average throughput comparisons and Fig. 5.5 and 5.6 show the average delay comparisons across flows in each PHB for the various strategies. It can be seen that RLAP is able to find a policy that improves QoS for AF traffic and maintains QoS for EF traffic at the expense of QoS for BE traffic. This is despite having increased throughput for EF and AF traffic. We also observe that RLAP is able to adjust to this policy regardless of the initial weight settings. For the case of static provisioning, we see that as the initial level of EF provisioning increases, the average throughput and delay of each PHB aggregate changes. This shows the level of bias given to provisioning EF. The performance for RLAP however, is almost consistent. This clearly demonstrates the ability of the algorithm to find an optimum strategy regardless of the initial values used.

Fig. 5.7 shows a chart of the improvement in revenue that RLAP makes over static provisioning. This is a key feature of RLAP; the ability to adapt provisioning based on a pricing plan and QoS requirements, such that long-term revenue is maximized.

The slight difference in levels of revenue for the RLAP case shows that though RLAP improves over static provisioning, the convergence to the optimal policy is slightly slower for the average and 50% over-provisioning cases. As with all adaptive algorithms, this is due to the need to converge from an initial point further away from the optimal point. Nonetheless, convergence is still achieved.

5.3.4.5 Comparison under Changing Traffic Conditions

In the second experiment, we set out to compare how RLAP improves over static provisioning over varying traffic conditions. EF traffic from S_0 and S_4 was halved after 10,000s and 15,000s respectively to cause the change. We used the peak-rate over-provisioning strategy for this experiment and the rest that follow since it is the common practice. Tables 5.4 and 5.5 summarize the results obtained and Fig. 5.8 shows the gain in revenue of RLAP over static provisioning across time. We see that RLAP always performs better than the static provisioning case and that it is able to adapt well to the changes in traffic pattern at time 10,000s and 15,000s, evidenced by a drop followed by an increasing trend in gain in revenue. Increase in gain is observed despite more favorable conditions for the static provisioning (since there is relatively lighter priority traffic). We also see that the QoS performance is similarly good, as in the previous experiment. By plotting out all 5 runs in Fig. 5.8, we see a trend that regardless of the random seed used, the algorithm would still converge. During the initial period, there is high fluctuation due to learning and exploration. But after 10,000s, all 5 runs converge. This trend is seen in other experiments as well.

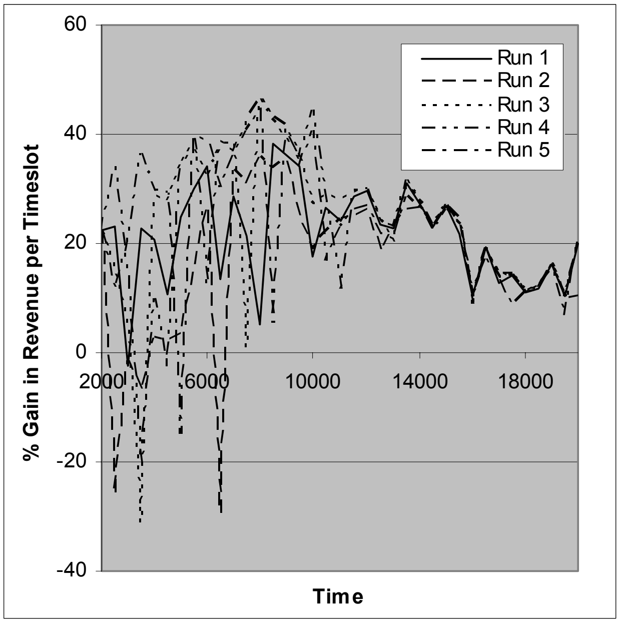

Figure 5.8: Percentage Gain in Revenue of RLAP over Static Provisioning across Time

TABLE 5.4: QoS and Revenue Achieved for Static Provisioning

Traffic Type	Throughput (bps)	Delay (ms)	Revenue
EF	51,679	10.5	804,743
AF	261,006	26.3	838,167
BE	105,262	87.6	464,828

TABLE 5.5: QoS and Revenue Achieved for RLAP

Traffic Type	Throughput (bps)	Delay (ms)	Revenue
EF	59,171	12.2	1,147,587
AF	273,094	18.7	1,159,049
BE	103,048	106.3	439,558

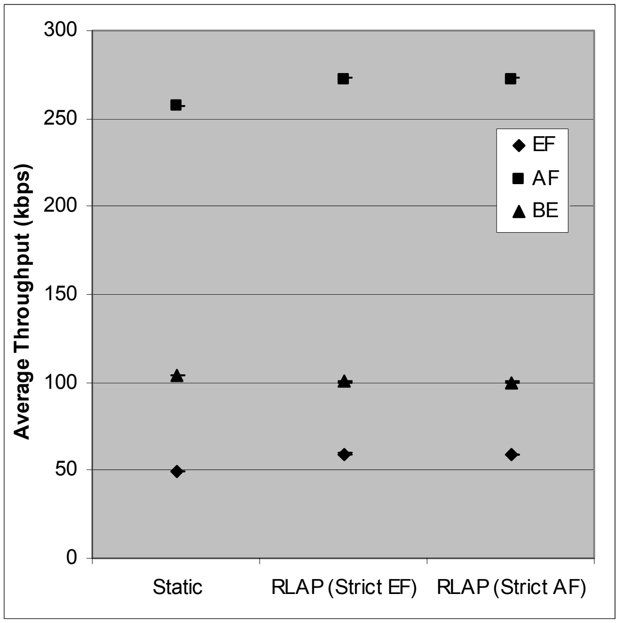

Figure 5.9: Average Throughput Comparison under Different QoS Requirements

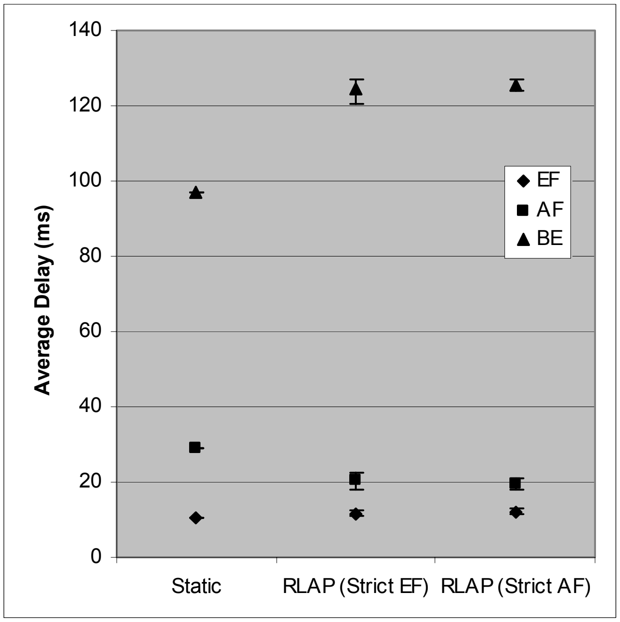

Figure 5.10: Average Delay Comparison under Different QoS Requirements

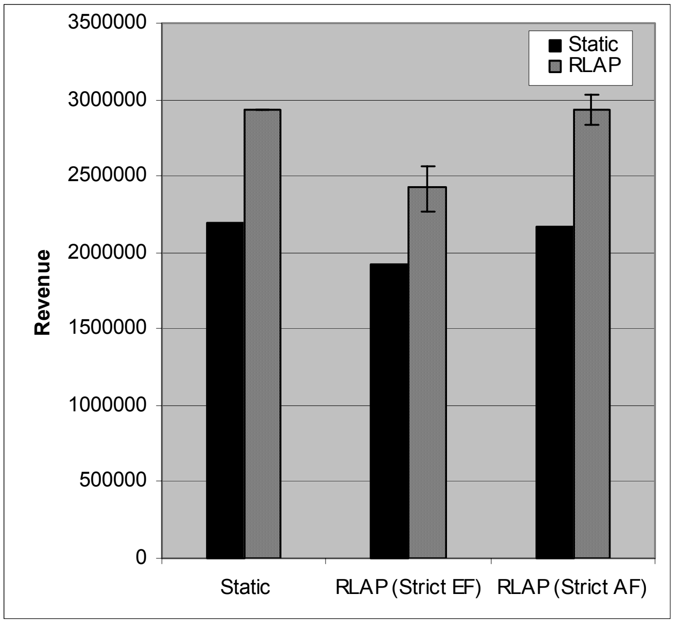

Figure 5.11: Revenue Comparison under Different QoS Requirements

5.3.4.6 Comparison under Different QoS Requirements

In our third experiment, we seek to demonstrate how RLAP performs when QoS requirements are changed. To alter the QoS requirements, we reduced the EF delay bound to 12ms in one case and the AF delay bound to 30ms in the other case to represent stricter EF and AF requirements respectively. Fig. 5.9 and 5.10 show a comparison of average throughput per flow and average delay across flows for each PHB between static provisioning and RLAP for both QoS cases. As the results do not change with QoS settings for the static provisioning case, the values for static provisioning in the figure are the same for the stricter EF and the stricter AF cases. Hence, we classify the static provisioning cases together under “static”. Note that the range markers may not be clear in each of the figures. This is attributed to the small variation in performance between the 5 runs.

We see that RLAP learns a policy that improves the performance of AF QoS as AF QoS requirements becomes stricter. We recall that AF is provisioned at the average rate initially. This is clearly not sufficient in this case, and under-provisioned AF traffic gets penalized heavily. As such, the effect of QoS requirements must be taken into account. It is also noted that in the case of stricter EF QoS requirements, RLAP does not see the need to improve the performance of EF QoS, due to the already good QoS performance. This means that not all adjustments to QoS requirements need to be taken into account equally. RLAP demonstrates here its capability to act accordingly. Fig. 5.11 shows the consequent improvements in revenue earned as a result of RLAP.

5.3.4.7 Comparison under Different Pricing Plans

In the final experiment, we seek to confirm that RLAP performs according to the pricing plan used; unlike most provisioning schemes proposed that do not consider this. To vary the pricing plan, we doubled EF revenue and penalties for one case and doubled AF revenue and penalties in the other case. Fig. 5.12 shows the improvement in revenue made between static provisioning and RLAP.

We see that RLAP makes more significant improvements in net revenue earned than static provisioning when the pricing of EF and AF traffic carried is doubled. This shows clearly that RLAP adjusts weights according to the pricing plan as well. A larger gain in profits will encourage the RLAP scheme to increase provision of that component which is causing the good result.

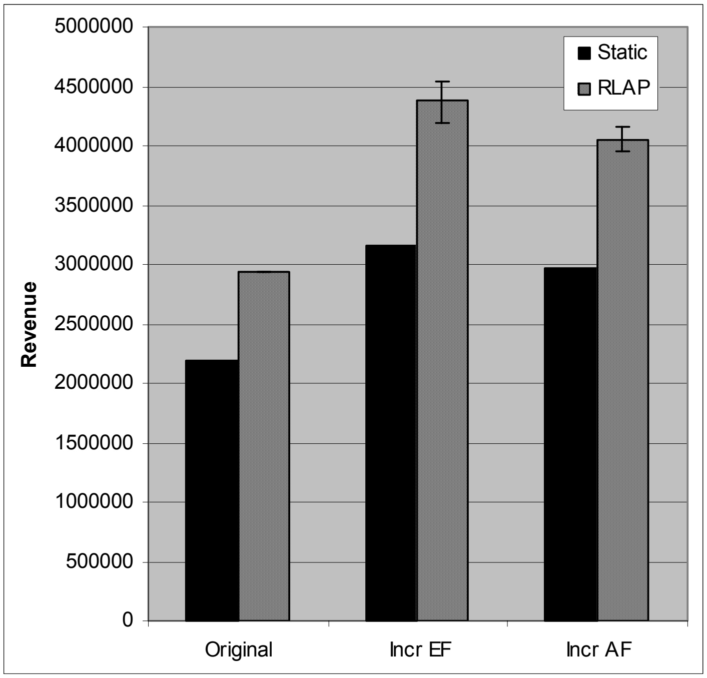

Figure 5.12: Revenue Comparison under Different Pricing Plans

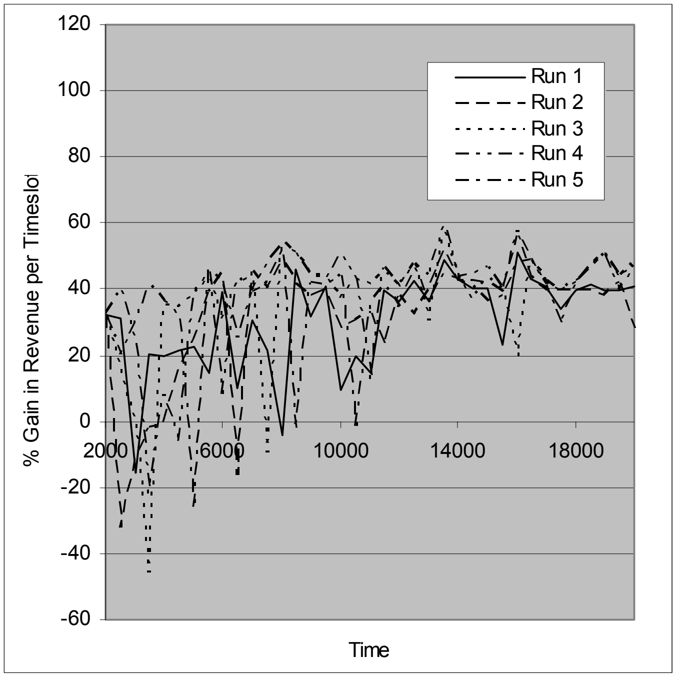

Figure 5.13: Percentage Gain in Revenue across Time for Increased EF Pricing Plan

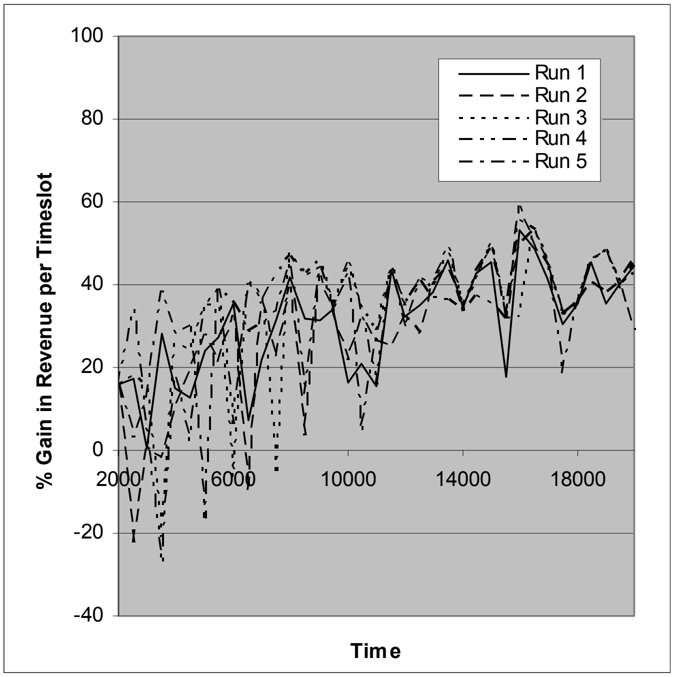

Figure 5.14: Percentage Gain in Revenue across Time for Increased AF Pricing Plan

Fig. 5.13 and 5.14 show the percentage gain in revenue across time of RLAP in comparison to static provisioning. We see that RLAP initially requires a learning phase where at times it fairs worse than the static provisioning. However, after about 10,000s, RLAP begins to make significant improvements. In both figures, we see that there is a convergence. The fluctuations observed towards the latter part of the simulation are not due to non-convergence, as confirmed by decreasing σ values. Rather, they are due to fluctuating traffic conditions.

From all the results of the above experiments, one might be tempted to think that if we set out to provision EF and AF traffic at higher rates initially and left them static, we could achieve the same results as RLAP. Unfortunately, we could not have known how much to provision a priori, and if traffic conditions were to change or pricing plans and QoS requirements altered, we would not be able to determine the corresponding provisioning without complex analysis, which may require certain assumptions. This is where RLAP provides a simple solution that is able to learn and adapt without supervision or expert analysis.

5.4 REINFORCEMENT LEARNING DYNAMIC PROVISIONING BASED ON QUALITY OF SERVICE REQUIREMENTS

While the RLAP scheme is economically efficient, many of the service providers would not like to charge by the traffic usage. There are many reasons for this. Firstly, the billing infrastructure to implement such a charging scheme can be quite complex and costly. There must also be external auditing done to ensure that the charges are correctly accrued, which is a difficult auditing task. Secondly, service providers prefer to offer fixed plans; for example, plans that charge an amount that includes pre-paid usage. This gives service providers a minimum level of commitment from customers. This is crucial in an environment where customer loyalty is hard to come by.

An alternative scheme to RLAP is presented that is based on guaranteeing an assured level of service to customers. The scheme is compatible with current-day pricing plans and service level agreements [43].

5.4.1 Service Level Agreements

In many of today's standard SLA contracts, QoS requirements are backed by an assurance level. For example, a 99.9% assurance level for delay means that at least 99.9% of the packets should experience end-to-end delay of less than the delay bound specified. In this way, the service provider can have some leeway for occasional service lapses that may be entirely unforeseen. The lower the assurance level, the easier it would be for the service provider to meet the requirements. This could also mean lower prices for the service. This is especially attractive to the customer if he has a higher level of tolerance, for example, if he is using a non-mission-critical service like a video streaming service.

However, such assurance levels are to be strictly adhered to. Customers understand that there may be occasional lapses, but if the frequency of QoS violations goes beyond the tolerance level that they have specified, the functioning of customers' services would be severely hampered. Sometimes, this may lead to huge business losses if the customers are corporate clients. Therefore, in such arrangements, the service provider is obliged to keep the service level well within the limits, while still maintaining a high level of utilization (profitability).

5.4.2 RLDP Algorithm

The concept behind the proposed Reinforcement Learning-based Dynamic Provisioning (RLDP) scheme is quite different from the RLAP scheme. Here, the SLS is used to judge whether the frequency of QoS violations is beyond the assured level. The degree to which the assurance level is violated is used as the feedback mechanism to determine a policy that maintains the frequency of QoS violations below the assured level. The amount of penalty (not revenue) received at the end of each time interval determines how the RL agent adaptively adjusts weighted fair proportions in each DiffServ router at each interval.

The same framework as that described for RLAP in section 5.3.2 is used (refer to Fig. 5.2). However, in the RLDP scheme, SRV units are used in place of Gaussian units. This is necessary because the objective here is to enforce QoS levels, which is different from maximizing revenue. The requirements of the objective are relatively stricter and necessitate a faster, more direct responding algorithm. The SRV algorithm directly relates the context experienced and the action to take. This means that bandwidth provisions can directly respond to changing traffic conditions, without the need to respond only as a result of the reinforcement signal (as for the Gaussian unit). The SRV algorithm works by modifying parameters in a neural network (specifically a multi-layer perceptron network) that has traffic rates as the input (context) and WFQ weight settings as the output (action). Another difference is the bandwidth broker is used to compute penalties based on the SLS. A pricing plan is not needed for the computations.

For each agent RL_i, the RLAP algorithm is as follows (omitting the subscript i for clarity):

```
Initialize $\theta_0$ and $\sigma_0$ and set $\mu_0 = 1 + \frac{100}{1 + exp\left(\theta_0{}^T x_0\right)}$
Select    $y_0 \sim N(\mu_0, \sigma_0)$
At the end of every interval $n$,    $n \geq 1$
Update    $\theta_n = \theta_{n-1} - \alpha_\theta \frac{(r_{n-1} - \hat{r}_{n-1})}{\hat{r}_{n-1}} (y_{n-1} - \mu_{n-1}) x_{n-1}$
          $\sigma_n = \sigma_{n-1} - \alpha_\sigma \frac{(r_{n-1} - \hat{r}_{n-1})}{\hat{r}_{n-1}} \frac{(y_{n-1} - \mu_{n-1})^2 - \sigma_{n-1}{}^2}{\sigma_{n-1}}$
          $\mu_n = 1 + \frac{100}{1 + exp\left(\theta_n{}^T x_n\right)}$
          $\hat{r}_n = \gamma r_{n-1} + (1 - \gamma)\hat{r}_{n-1}$
Select    $y_n \sim N(\mu_n, \sigma_n)$
```

where, x_n is the vector of the traffic state observed at $(n-1)^{th}$ time interval and y_n is the vector of WFQ weights to be set for the n^{th} time interval. μ_n is the mean used to set y_n. σ_n is the variance used to set y_n and θ_n is the NN parameter relating x_n and

μ_n. r_n is the penalty received for QoS violations in the n^{th} time interval and $\hat{r}_n$ is the cumulative penalty for QoS violations beyond assured levels. α_θ is the step size for θ_n and α_σ is the step size for σ_n.

The idea behind the RLDP algorithm is similar to the RLAP algorithm. Perturbation is added using a Gaussian unit. If the perturbation has caused the unit to receive a penalty that is less than the cumulative penalty, it would be desirable for the RL agent to shift μ_n closer to y_n. To do that the NN parameters θ_n are adjusted in a negative gradient direction. This means that the adjustment should be in the opposite direction to $(r_n - \hat{r}_n)$ and toward $(y_n - \mu_n)$. The x_n factor is used to scale the gradient according to the traffic state. Note that y_n and μ_n are three-dimensional vectors representing the three traffic classes (EF, AF and BE). x_n is a six-dimensional state, comprising of the average traffic proportions and the average buffer occupancy of the three classes. Thus, the NN parameter θ_n would be a 3×6 matrix linking x_n to μ_n. A measurement-based framework is used, where the average traffic proportions are computed as the proportion of traffic for a class with respect to the total traffic measured across each interval and are weighted averaged by a factor of 0.8. Similarly, the average buffer occupancy is measured across each interval. A simple Multi-layer Perceptron (MLP) network with 2 layers is used to represent the relation between the state and the action. The MLP network outputs a mean μ_n that lies in a range $1 \leq \mu_n \leq 101$. This is to simplify the implementation of y_n, which we set to be an integer between 1 and 100.

5.4.3 Penalty Function

Since the objective of RLDP is to ensure that QoS assurance is maintained, the penalty function has been set up to penalize any QoS violation beyond the level assured (agreed upon in the SLA).

The penalty function used in RLDP is given by

$$r = \sum_{j} [exp(c_{loss,j} \times l_{i,j}) + exp(c_{dly,j} \times d_{i,j})] \qquad (5.2)$$

For each traffic class j, $l_{i,j}$ is the percentage of packets loss exceeding the packet loss requirement at router i and all routers downstream from i, and $d_{i,j}$ is the proportion of traffic above the tolerance level for delay that passed through router i not meeting the delay bound. $c_{loss,j}$ is the weight given to $l_{i,j}$, and $c_{dly,j}$ is the weight given to $d_{i,j}$.

The penalty function was designed with the following features. Firstly, the exponential function has an inherent momentum feature, in that the gradient gets steeper as QoS is violated by a greater extent. This would promote faster convergence toward better QoS. The gradient becomes less steep as it nears the QoS level required to promote more accurate convergence. When QoS is better than required, the gradient is negligible. This concavity stabilizes the algorithm within the solution space. Secondly, we included weights for each QoS component. These weights not only control the steepness of the function's gradient, but also give relative importance to the QoS components. Therefore, if it is more crucial for EF class QoS requirements to be met, then the respective weights may be set higher with respect to the others. The service provider could set this based on the relative value of the service contracts.

With feedback based on the extent that QoS is being met, RLDP is focused on maintaining SLA agreements. This is superior to methods that focus on achieving relatively better QoS, but are unable to achieve specific QoS targets. RLDP also has the ability to load-balance when traffic is heavy, such that all classes co-operatively meet QoS requirements. This is because, if one class meets QoS at the expense of another, there would be a higher penalty, using the exponential penalty function.

5.4.4 Simulation and Results

5.4.4.1 Simulation Setup

The topology in Fig. 5.15 was set up to compare bandwidth provisioning with and without RLDP. The set up is similar to Fig. 5.2 for the RLAP simulations. The only difference is that the link capacities have been changed to make every link a bottleneck link. This would require stringent bandwidth provisioning on all links in order to maintain QoS.

5.4.4.2 Traffic Characteristics

In the simulations, traffic from all 3 PHB aggregates were generated on all links in the DiffServ domain. The links were made to carry close to full capacity most of the time. The characteristics of sources S_0 to S_7 are given in Table 5.6. All sources continuously generate traffic flows that have exponential inter-arrival and holding times with mean values as given in the table.

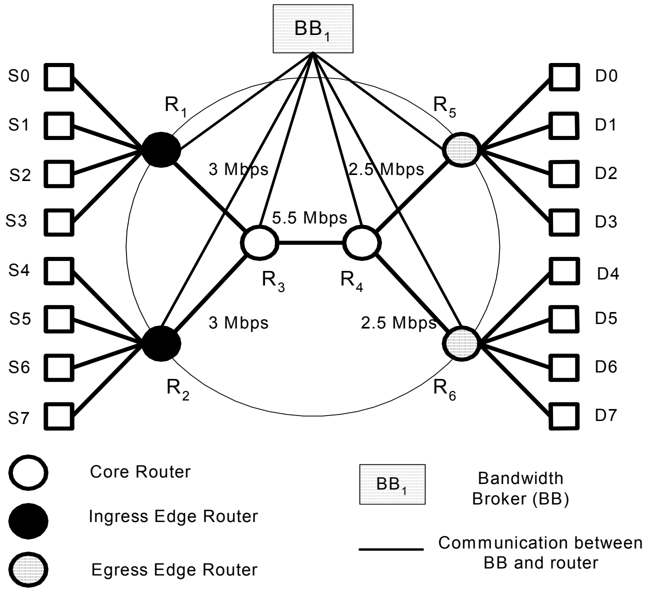

Figure 5.15: DiffServ Network Topology for RLDP simulations

TABLE 5.6: Characteristics of Traffic Sources

Source	Traffic Type	Connection Inter-arrival Time (s)	Connection Holding Time (s)	ON Rate (kbps)
S_0, S_4	Exponential ON/OFF	3.75	60	32
S_1, S_5	CBR	12.0	60	128
S_2, S_6	Exponential ON/OFF	10.0	60	150
S_3, S_7	Pareto ON/OFF	3.0	30	128

S_0 and S_4 are EF sources that represent delay bounded traffic like VoIP traffic. S_2 and S_6 are AF sources that represent loosely delay bounded, high throughput requirement, traffic like video traffic. S_1 and S_5 and S_3 and S_7 are BE sources that represent FTP traffic and non-bounded aggregated web traffic respectively. The average amount of generated traffic towards each ingress router for EF, AF and BE are 0.5 Mbps, 1 Mbps and 1.5 Mbps respectively.

5.4.4.3 Experimental Details

In the experiments that follow, a comparison is made between provisioning with and without RLDP. Without RLDP, provisioning is essentially left static throughout each experiment lasting 20,000s. The long experiment time demonstrates how provisioning works on medium time scales. Another reason for the long simulation time is for various traffic conditions to be encountered during the course of simulation. For the experiment with RLDP, we set μ_0 to be equal to the static setting. The RLDP scheme only kicks in after 1,500s (the initial lag is to allow the traffic to build up in the network), adjusting the WFQ weights in the routers adaptively and improving the policy in the RL agents through time. The initial service weights for the 3 classes of traffic were set to be equal. Though this initial setup overprovisions for EF and AF classes, it is shown that the amount of overprovisioning cannot be easily determined and should be dynamically adjusted. Note that a minimum service weight can be set if a throughput bound is required by the SLA. We however do not simulate this, as a minimum throughput requirement should be pre-provisioned and left static, while the remainder of the link capacity left to be dynamically provisioned. Droptail queues were used for each class to separate the effects of RLDP from buffer management. The buffer length for EF, AF and BE traffic were set to 5, 30 and 30 respectively. The time-step interval T chosen was 500s; a reasonably long interval. This interval was used for each change in bandwidth provisions as well as for traffic and QoS measurements to be taken. The initial values of θ_0 and σ_0 were set to 0 and 40 respectively. By choosing θ_0 as zero, μ_0 is initialized to the value 51. The value of σ_0 was chosen quite large to encourage greater exploration at the beginning. The constants α_θ, α_σ and γ were set to 0.1, 1.0 and 0.8 respectively.

TABLE 5.7: QoS Achieved for Static Provisioning

Traffic Type	Packet Loss (%)	Delay (%)
EF	0.29	0.003
AF	9.0	10.7
BE	1.2	-

TABLE 5.8: QoS Achieved for RLDP

Traffic Type	Packet Loss (%)	Delay (%)
EF	0.78	0.37
AF	2.1	0.6
BE	2.0	-

5.4.4.4 Comparison between Static Provisioning and RLDP

In the first experiment, we set out to compare provisioning with and without RLDP. The QoS requirements for EF and AF delay bounds were set to 20ms and 60ms respectively. The tolerance on the percentage of packets delayed was set to 1% for EF and 5% for AF. The maximum packet loss for EF, AF and BE were set to be 1%, 5% and 20% respectively. We set the weights in the penalty function in equation (5.2) to 5.0 for all constraints to give equal weighting. Tables 5.7 and 5.8 summarize the QoS achieved for the last 10,000s of the simulation. We see that for the case without RLDP, the AF traffic did not meet its QoS requirements for packet loss and delay. Clearly, this shows that the AF requirements are strict, and relatively more bandwidth share has to be given to AF to ensure that the requirements are met. We see that with RLDP, all 3 classes meet their requirements. This was achieved by trading off the QoS of EF and BE, but not to the extent of violating their requirements.

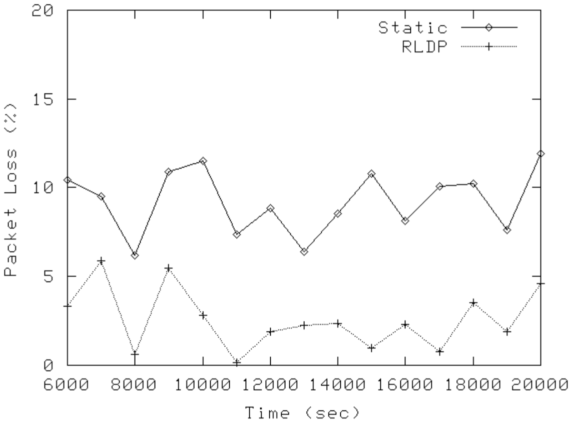

Figure 5.16: Percentage of AF Packet Loss at 1000s time intervals

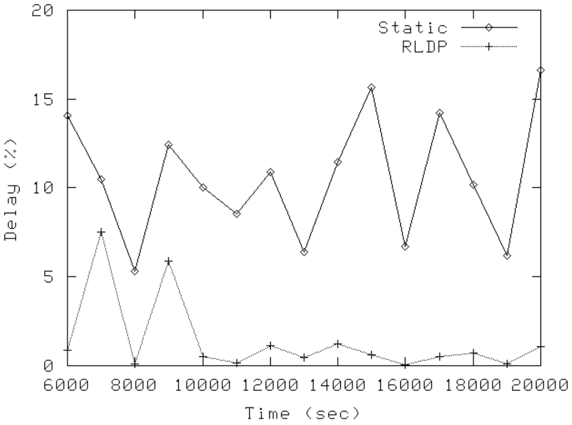

Figure 5.17: Percentage of AF Packets Delayed at 1000s time intervals

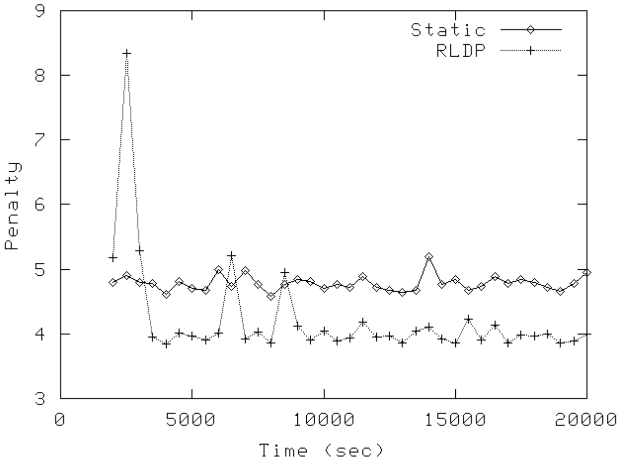

Figure 5.18: Penalty per time interval for R_5

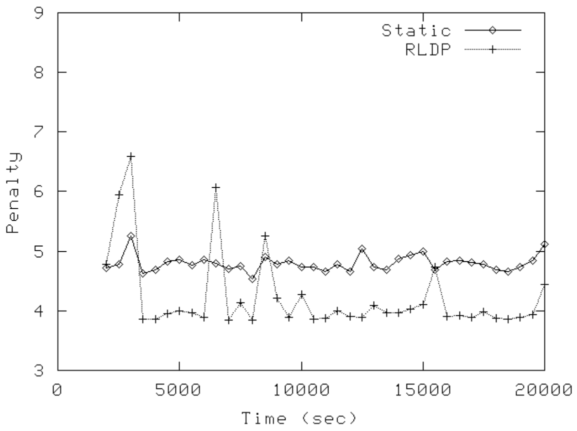

Figure 5.19: Penalty per time interval for R_6

Fig. 5.16 and 5.17 show the percentage of AF packets loss and delayed in 1000s time intervals. We see that the RLDP algorithm has learned a strategy that dynamically provisions such that the AF traffic is consistently able to meet its QoS requirements across time. This was done by increasing the bandwidth proportion of AF whenever AF traffic and average buffer occupancy was relatively high. Note that in the RLDP algorithm, the mean weight is dependent on the NN output that considers the traffic proportions and buffer occupancies of all 3 classes. The argument is that if the state of only one class is used, there would be no relation between classes. For example, without this relation, a state that has high AF and BE traffic would be no different from a state that has high AF but low BE; when in fact, the former case would require higher AF weight relative to BE. Fig. 5.18 and 5.19 show the improvement in penalty feedback across time for routers R_5 and R_6. This gives us an insight to how well the two sets of aggregate flows are meeting QoS as they exit the domain. We see that RLDP is able to minimize the penalty compared to static provisioning. We also see the improvement of the penalty across time.

TABLE 5.9: QoS Achieved for Static Provisioning

Traffic Type	Packet Loss (%)	Delay (%)
EF	0.29	1.3
AF	9.0	2.0
BE	1.2	-

TABLE 5.10: QoS Achieved for RLDP

Traffic Type	Packet Loss (%)	Delay (%)
EF	0.11	0.9
AF	5.3	0.6
BE	1.6	-

5.4.4.5 Comparison under Strict EF Requirements

In our second experiment, we seek to demonstrate how RLDP performs when QoS requirements are changed. To alter the QoS requirements, we reduced the EF delay bound to 10ms and increased the AF delay bound to 90ms. The AF delay tolerance and packet loss requirements were increased to 10% and the EF packet loss requirement was reduced to 0.5%. This would relax the AF QoS requirements and make the EF QoS constraints active. This makes the solution space to the problem much smaller. To give more significance to the EF constraints, we also increased the EF weights in the penalty function to 10.0 and decreased the other weights to 1.0. Tables 5.9 and 5.10 summarize the end-to-end QoS achieved. We see that now, for the case without RLDP, the EF traffic is unable to meet its QoS requirements for delay, confirming that the EF requirements are the new constraint. However, with RLDP QoS is able to be met for all classes.

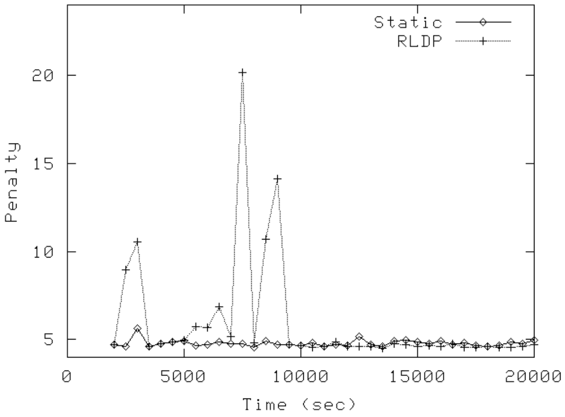

Figure 5.20: Penalty per time interval for R_6

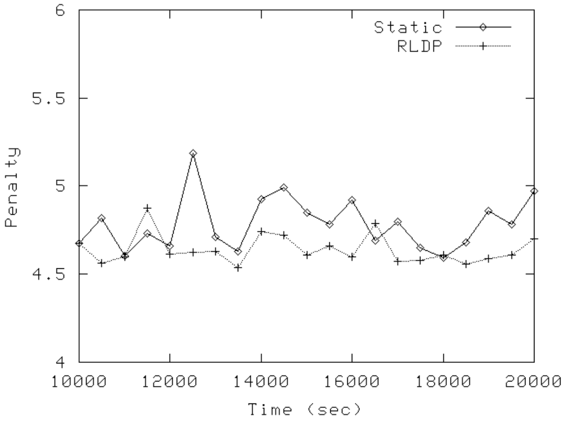

Figure 5.21: Penalty per time interval for R_6 from 10,000s

Fig. 5.20 and 5.21 show the improvement in penalty across time for router R_6. We see in Fig. 5.20 that the QoS is very poor for RLDP before 10,000s. This can be attributed to exploration. Since the solution space is small, the learning process takes a longer time to converge. Though RLDP gets penalized heavily during its adaptation phase, these "bad" experiences are used to improve the policy. In fact, RLDP begins to perform better than the static provisioning after 10,000s, as seen in Fig. 5.21. If high penalties cannot be tolerated, then a smaller initial variance σ_0 has to be used, incurring a trade-off in convergence speed if the initial policy is far from the optimal one.

From all the results of the above experiments, one might be tempted to think that if we set out to provision EF and AF traffic at higher rates initially and left them static, we could achieve the same results as RLDP. However, whether to provision more to EF or AF traffic, and by how much more, is still an issue. Furthermore, we could not have known how much to provision a priori, and if traffic conditions were to change or QoS requirements were to be altered, we would not be able to determine the corresponding provisioning without further analysis. This is where RLDP provides an intelligent solution that is able to learn and adapt without supervision or expert analysis.

CHAPTER 6

REINFORCEMENT LEARNING-BASED PROVISIONING FOR RADIO ACCESS NETWORK

6.1 INTRODUCTION

In this chapter, we discuss the provisioning of UMTS radio access networks. The UMTS radio access network, known as the UMTS terrestrial radio access network (UTRAN), has characteristics that are different from the UMTS core backbone network discussed in the previous chapter. Firstly, radio access networks generally have a hierarchical topology and are much smaller in size. A SGSN typically serves a few radio access networks, which are individually controlled by a radio network controller (RNC) in each radio network subsystem (RNS). A RNS is a collection of a RNC and the base stations (node B) administered under it. A figure of the topology can be seen in chapter 2 (Fig. 2.1).

Secondly, handoff (or mobility) patterns greatly influence the traffic pattern and mix. A handoff occurs when a mobile user moves from one base station’s radio coverage, called a cell, to another neighboring cell. Traffic from the user and traffic destined to the user stops being transmitted through the old base station, and is instead transmitted through the new base station. This can occur in an abrupt manner (hard handoff) or in a smooth and continuous manner (soft handoff), which is supported by UMTS networks. The transfer of “responsibility” from one base station to another is a complex process known as mobility management, and is achieved through the Mobile IP protocol [104,105]. Sometimes when a handoff occurs, it could be between base stations administered by different RNCs. When this happens, an inter-RNC handoff occurs at the same time as an inter-node B handoff takes place. This hierarchical model of mobility management extends to inter-SGSN handoff as well when a user possibly moves between two areas served by different SGSNs. We consider traffic pattern changes due to inter-SGSN handoffs as less dynamic. Methods for provisioning UMTS core backbone networks, where inter-SGSN handoffs are handled, have already been discussed in the previous chapter. In this chapter, we consider the more dynamic nature of UTRANs.

Lastly, at the access portion of the network, QoS management is more critical. In the upstream direction, smaller flows are aggregated into larger flows. The way traffic is aggregated, i.e. how much of the bandwidth proportion should be allocated to different classes of traffic, affects how QoS can be met. In the downstream direction, large capacity links are feeding into smaller capacity links and aggregated traffic flows are being distributed. Routers upstream should be aware of the congestion levels occurring downstream and appropriately regulate the traffic proportions of the various classes. Otherwise, QoS cannot be maintained no matter how routers downstream try to handle the traffic passed from negligent upstream routers. QoS management have been incorporated into the Mobile IP protocol by Das [106].

We see therefore that the provisioning methods for UMTS core networks cannot be applied directly to UTRANs, since the characteristics are quite different. Most of the methods in the literature treat bandwidth allocation in RANs in a different way from core networks. In fact, most of the methods borrow heavily from cellular network theory, with modifications to suit next-generation multimedia wireless networks. In the following section, a survey is done, covering the more prominent work. (There has been a lot of work done in this area). The reasons why such methods might be inadequate for UTRANs are given. A solution that handles bandwidth provisioning in a QoS-objective way is then proposed.

6.2 CURRENT METHODS OF QOS PROVISIONING WITH MOBILITY FACTORED

The use of Mobile IP would enable users to move freely from one place to another and have seamless connection. Users should expect that the QoS level be maintained even as they roam around. Each time a user moves between cells, traffic paths migrate due to handoffs as shown in Fig. 6.1. If there is insufficient bandwidth along the new path to support the traffic being handed over, the user's connection could either be blocked or degraded. If the action taken is to block the user's connection, a more appropriate measure of QoS would then be the probability that his call is blocked; since QoS of all other existing calls is assured at the expense of call blocking.

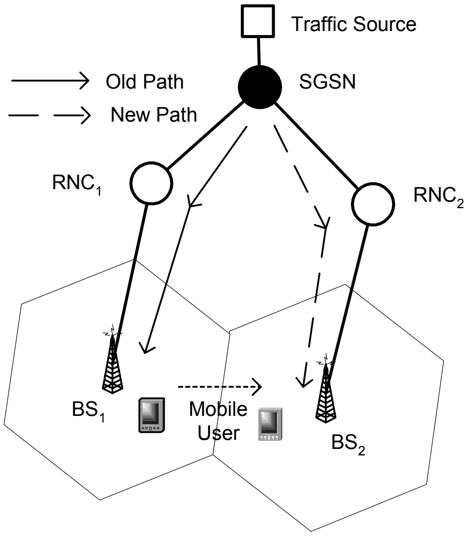

Figure 6.1: Mobile Handoff

Most methods for QoS provisioning in cellular networks make use of call blocking to maintain QoS. QoS requirements such as delay, packet loss and throughput bounds are assumed to be maintained through appropriate reservation. For example, in Fig. 6.1, when the mobile user is in the former cell, bandwidth in the neighboring cells like the latter cell may be reserved in advanced before the handoff is made. The task then is to ensure that the cell that the mobile user is moving to has sufficient bandwidth; otherwise the call would be blocked. This is a difficult task as there would be a need to balance between over-reserving, causing too much bandwidth to be wasted, and under-reserving, resulting in too high a call-blocking probability.

Mitchell [107] in his paper has discussed the effects of mobility on bandwidth allocation strategies in multi-class cellular networks. He pointed out that bandwidth sharing between classes (as opposed to bandwidth partitioning) results in better utilization. Conversely, if bandwidth is reserved strictly on a per-flow basis so as to guarantee QoS, bandwidth utilization would be unnecessarily low. However, Mitchell also noted that the mobility rate of one class can cause QoS in other classes to be adversely affected. That means that some efficient way of bandwidth sharing that takes into consideration the traffic patterns should be used. But little work has been done to model UMTS traffic [108], possibly due to the lack of real-implementation experience. Thus, most methods use some form of ad-hoc method or predictive method to allocate bandwidth efficiently.

6.2.1 Call Admission Control and Reservation-based Methods

Call admission control is closely tied to the method of bandwidth allocation or reservation. There are two types of call admission control in a cellular network – new call admission and handoff admission. A call is admitted only if there is enough spare capacity. Sometimes, there may be unused capacity, but it may be reserved for other users, causing the call to be dropped. Depending on the scheme used, handoff calls may be treated with higher priority as compared to new calls. Studies have shown that people tend to tolerate a dropped call in progress less than a new call being rejected.

Therefore, most methods proposed are more conservative in that they consider if the call can be sustained before admitting the call. The complex decision has to be based on a large number of factors such as the network congestion level, the mobility and the usage pattern of the user and other users, and the QoS required of the user.

Choi [109] surveys and compares a number of bandwidth reservation and admission control schemes. One class of schemes like his, Naghshineh's [110] and Wu's [111] aim to keep the handoff dropping probability below a target level by estimating handoff probabilities. Only admission control is used and no bandwidth is explicitly reserved. These schemes determine how admitting a call would affect handoff blocking by using historical mobility data. There are some other admission control schemes that make use of prediction methods to determine the handoff blocking probability [112-114].

Another class of schemes, like the ones proposed by Talukdar [115], Mahmoodian [116] and Yoon [117], make use of bandwidth reservation to reserve bandwidth in advanced, rather than just relying on admission control. The first two schemes however, reserve bandwidth on a per-connection basis. Per-connection reservation may guarantee better QoS but are not bandwidth efficient and are computational intensive. The third scheme makes reservations based on DiffServ classes. Oliviera [118] improves per-connection reservations by introducing a scheme that reserves bandwidth dynamically. Bandwidth reservations are adjusted reflecting network conditions, which helps conserve bandwidth when possible.

There are schemes that make use of shared bandwidth reservation that have much higher utilization due to statistical multiplexing and are less computationally intensive as they may be time-driven rather than event-driven. The reservation may be done on a per-class basis. Misic [119] describes a dynamic way of computing the amount of bandwidth to reserve by estimating the bandwidth reservation rate, which is dependent on the handoff arrival rate. Li [120] and Kim [121] also propose schemes that make use of local information to estimate the amount of bandwidth to reserve. Zhang [122] also proposes a scheme that adjusts bandwidth reservations using local information. His scheme however is based on a predicting the instantaneous handoff traffic demand, which is different from other schemes that model factors that impact the demand like handoff rates and mobility patterns.

A third class of schemes combine admission control with dynamic reservation. These schemes aim to first adjust reservations to accommodate handoff calls; failing which the degradation of other calls may be considered to allow the handoff call in. If the degradation falls below a threshold level, call admission control can be used to block calls to prevent further degradation. Das [123] describes such a scheme. It is not enough to take into consideration the bandwidth utilization and call blocking probability, degradation performance as well as frequency of bandwidth reallocation are also important considerations. As such, Chou [124] proposed a scheme that combines admission control and adaptive bandwidth reservation that takes these factors into consideration. Though these are good performance measures, mobile users are more interested in concrete QoS measures like call blocking, latency and packet loss. To be able to quantify and optimize performance in these terms are important.

6.2.2 Problems with Call Admission Control and Reservation-based Methods

The previous section describes schemes that have certain similar characteristics. Firstly, they are concerned with bandwidth provisioning on the wireless portion of the network. There is not much concern about reservation along the new path after handoff, which does not consist only of the wireless portion. As mentioned, when a handoff occurs, many links in the RAN are also affected, depending on the type of handoff. Only schemes described in references [106,107,115,116] consider handoffs in the Mobile IP context. Most of the schemes are based on cellular networks, and therefore cannot be applied to the rest of the RAN.

Secondly, most of the schemes are based on per-flow reservation or call admission, including all of those that consider path provisioning. This is perhaps due to the evolution from circuit-switched networks. However, UMTS networks are to be based on packet-switching. This means that calls are no longer individually routed, but are multiplexed together into aggregate streams. Traffic is also no longer constant bit rate as in voice networks, but is variable bit rate due to higher percentage of traffic consisting of data in the future. Some of the schemes consider this and make use of multi-class aggregate reservation. While this is a more efficient way to guarantee QoS, it is plagued by similar problems as bandwidth partitioning methods described in earlier chapters of this thesis. Aggregate reservation is essentially partitioning a certain proportion of the link capacity for handoff calls. In fact, the bandwidth wastage is greater since there is provision for possible handoff traffic. Of course some schemes, like the one proposed by Lee [125], try to overcome such bandwidth wastage by letting lower priority traffic occupy the bandwidth first and allowing higher priority traffic to displace them later. However, the overheads of controlling the re-allocations are quite high.

Thirdly, with DiffServ becoming the more dominant service model in UMTS than IntServ, many of the schemes would not be suitable as they do not consider DiffServ implementation. There have been works on DiffServ implementations of 3G in the wireless network. These include DiffServ implementations for enhanced GPRS [126], CDMA [127] and Wireless LAN [128-130]. Sivalingam [131,132] has also done work on designing a framework for DiffServ to be used in mobile access networks. However, they have not addressed the issue of DiffServ in the RAN. Venken's work [133] is one of the few that can be found that describes how DiffServ can be implemented in the UTRAN. His work does an analysis of how DiffServ IP-based UTRAN would perform as compared to an ATM-based UTRAN [134]. The conclusion is that an IP-based UTRAN can perform just as well. This gives the impetus for using DiffServ-IP, since IP is more widely used in the data networking domain.

Lastly, the bandwidth reservation and call admission control schemes proposed in the literature do not consider pertinent QoS requirements that are observed in data networks, like latency, packet loss and throughput bounds. Instead, they focus on handoff blocking probability as the QoS requirement. This could be due to differences in paradigms used by researchers in the cellular network field and in the data networking field. While call QoS requirements are met by allocating a certain fixed channel bandwidth, data QoS requirements are quite different owing to the variable bit rate nature of data traffic. A fixed bandwidth allocation is not practical for data traffic, as discussed in earlier chapters, especially in the RAN, as there would usually be low bandwidth utilization and no gain from statistical multiplexing common in data networks.

Therefore, in order to get the best of both worlds, multiplexing gains from flow aggregation (seen in data networks) should be considered together with handoff traffic provisioning (seen in cellular networks). A framework is now presented that provisions bandwidth in a class-based manner and dynamically adapts the amount provisioned for each class on each link in the RAN based on QoS requirements contracted in the SLA and traffic conditions on the link as well as neighboring links.

6.2.3 Aggregated Provisioning-based Method as a Solution

In a DiffServ framework, users are assured a level of service based on the class of service they have contracted. The service model is such that users within the same class (usually having similar application requirements), are given the same treatment in the network. The service provider has to ensure that the treatment given to each class performs up to the standard contracted in the SLA.

When DiffServ is to be used in the UMTS context, mobile users would expect the same level of service extended to them. However, due to the mobility of users, service providers would require more effort in ensuring that the performance still performs up to standard regardless of where the user roams around the wireless network. While an in-advance reservation-based method (reserving enough bandwidth in advance for the user to roam without loss of bandwidth) can be used to seamlessly provide the same experience to the user, it is both bandwidth-inefficient and complicated to implement. Although current mobile networks use the reservation-based system, to provision for calls, next-generation networks would need to provision for a wide range of services that have diverse requirements.

We propose to use an extension of a provisioning method that most current-day fixed data networks implement. Bandwidth is provisioned for each class based on weighted fair proportions and all calls ("connections" in data network terminology) within the class completely share the bandwidth allocated. This is in line with IP-based networks, where packet-switching aggregates packets from different connections. There is no partitioning of bandwidth, not even between classes.

Bandwidth between classes can be shared according to the weighted fair proportion. Since bandwidth provisioning in the IP-based UTRAN has no notion of individual flows, traffic is treated as an aggregated flow that is continuous and has variable bit rate.

When a new connection or handoff is made, no admission control or reservation is required along any part of the path. The connection is always admitted, thus bringing about a very low blocking probability. There is no distinction between new and handoff connections; as there is no distinction of individual flows in the IP-based network. The new or handed-off connection consequently shares the bandwidth with the other connections in the same class. This may cause degradation in QoS, but the degradation would only be an issue if the QoS is below that which has been contracted. This can be ensured by the service provider by adjusting the weighted fair proportions such that the amount of bandwidth allocated is sufficient to support the aggregated traffic rate. The objective of the provisioning problem would then change from maintaining low blocking probability to maintaining high QoS, which are the performance measures directly stated in SLAs.

The control of weighted fair proportions on every link is done through the implementation of WFQ in each router. Unlike reservation-based methods, the aggregate bandwidth proportions can be adjusted on a medium timescale (hundreds of seconds), rather than on a short timescale (seconds). Even in dynamic multi-class reservation-based methods, the bandwidth reserved for each class has to be adjusted on a short timescale. This is because a long interval will cause a trade-off in temporal poor bandwidth allocation. This problem was discussed by Chou [124].

There are many advantages to this framework is summarized as follows,

1) Zero blocking probability – since there is no need for admission control and bandwidth reservation, connections are by default always accepted. The new or handed-off connection shares the bandwidth with the other connections in the same class. There is no bandwidth guarantee in this case. If throughput guarantee is required, then the proportion provisioned has to be in absolute terms.
2) Full utilization during congestion – in a reservation-based scheme, there is always bandwidth left unutilized. This is not the case with our framework as all bandwidth is shared using weighted fair queuing.
3) Simple implementation – because there is no need for reservation or admission control decisions, there is no need for such signaling protocols. The only signaling needed by our framework is the network condition monitoring and the control of WFQ weight settings in the routers. This is done on a time-interval basis, and a medium timescale (hundreds of seconds) is usually sufficient.

We present now a method to adjust the bandwidth proportions for each class such that QoS can be met for all classes despite high user mobility.

6.3 REINFORCEMENT LEARNING BANDWIDTH PROVISIONING BASED ON QUALITY OF SERVICE REQUIREMENTS

In section 5.4, a reinforcement learning-based bandwidth provisioning scheme based on QoS requirements for UMTS core backbone networks was presented. The scheme intelligently adjusted WFQ weight settings based on the traffic intensities of each class of traffic. Each link was provisioned by RL agents implemented in the DS domain's bandwidth broker. The RL agents make use of a neural network to map the measured traffic intensities to the WFQ settings at each interval. The neural network parameters were trained using a reinforcement learning algorithm that considers the feedback on QoS.

The Reinforcement Learning-based Dynamic Provisioning (RLDP) algorithm can be used in a similar way to provision links in the UMTS radio access network. The only problem is that in the access network, traffic patterns are more dynamic due to the continuously changing routes caused by mobile users moving from one location to another. Depending on the type of handoff – inter-node B or inter-RNC, traffic flows will migrate from one or more links to the neighboring links. To the bandwidth broker, only the net change in aggregate traffic is going to affect the QoS. For example, if 2 flows of 256 kbps moves from cell A to cell B, and another flow of 1 Mbps moves from cell B to cell A, the net increase in traffic in cell A is 512 kbps and the net decrease in traffic in cell B is 512 kbps. Thus the bandwidth provisioning should not be based so much on the actual mobility pattern of users, but more on the nett handoff traffic rate, considering both incoming and outgoing users.

By including the traffic rates of neighboring links into the context of the RL agent, the agent would be able to learn the relationship between the traffic intensity in the region (inclusive of own link and neighboring links) and the bandwidth to provision. Intuitively, there should be a positive correlation between the traffic intensity in the region and the amount of bandwidth provisioned. The relationship is in fact related to the probability distribution handoff traffic. Therefore, the RL agent is learning some function of the joint probability distribution of handoff traffic from the region.

6.3.1 RLBP Algorithm

We now introduce the Reinforcement Learning-based Bandwidth Provisioning (RLBP) algorithm, which is the extension to the RLDP algorithm. To describe how RLBP can be implemented in a DiffServ UTRAN network, a one-domain topology in Fig. 6.2 is used here as an example, and is made as simple as possible without loss of generality. At the top of the hierarchical topology, the SGSN connects the UMTS core backbone network to the UTRAN and is an ingress edge router to the UTRAN DiffServ domain in the downstream direction. Radio network controllers RNC_1 and RNC_2 are core routers and base stations BS_1 to BS_4 are egress edge routers that distribute data streams to the mobile nodes within their wireless coverage. In the upstream direction, BS_1 to BS_4 are ingress edge routers for traffic originating from mobile nodes and the SGSN serves as an egress router connecting the UTRAN to the UMTS core backbone network. Traffic can also flow from one mobile node to another. In this case, the base stations act as both ingress and egress routers for those flows.

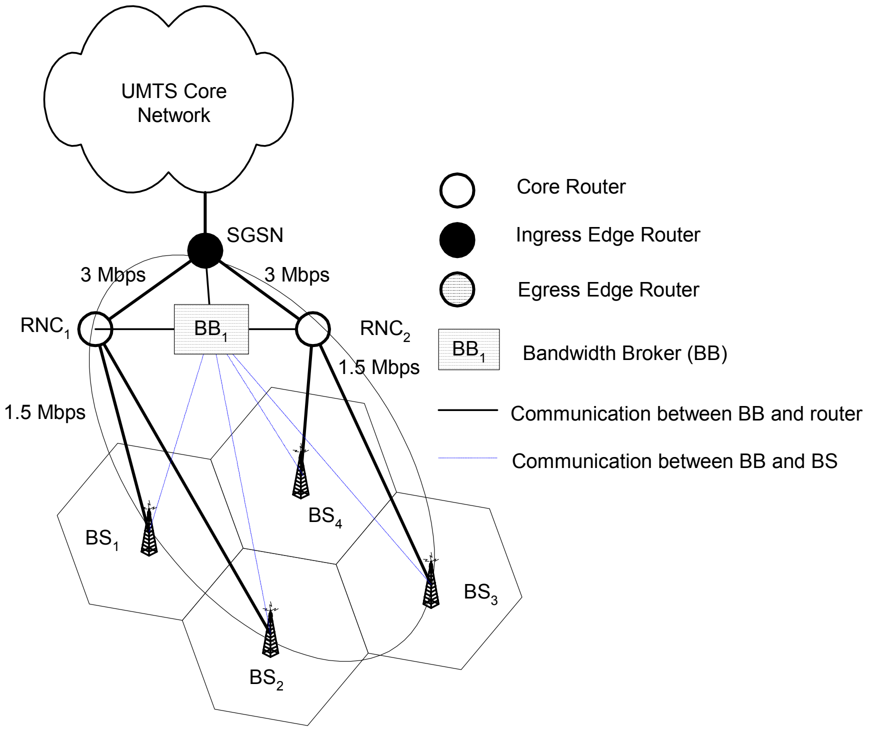

Figure 6.2: UTRAN DiffServ Network Topology (in downstream direction)

A bandwidth broker BB_1 is used as the centralized collection and decision-making point. The function is similar to the one in the RLDP framework described in section 5.4.2. At regular intervals, BB_1 collects traffic measurements (in terms of number of bits) from all routers in the domain as context information. Concurrently, destination nodes and all routers report the last interval's QoS, in terms of amount of traffic delayed and number of packets dropped, to the bandwidth brokers along the paths of the flows. BB_1 then makes decisions through the RL agents and sends the WFQ weight configurations to the respective routers. SLAs are stored in the database for computing the QoS penalties, which are used as feedback to the RL agents.

The RLBP algorithm is essentially the same as the RLDP algorithm. The only difference that modifies it for use in the UTRAN is that the context x used has been expanded to include not only the traffic rates for incoming flows of the router, but also the traffic rates of all incoming flows into neighboring links. For example, in Fig.

6.2, the link between RNC_1 and BS_1 has a 9-dimensional state space that includes the incoming traffic rates of each class of traffic (EF, AF and BE) from link SGSN-RNC_1, link BS_2-RNC_1 and neighboring link SGSN-RNC_2. This additional context information would enable the RL agent to consider the traffic that may be handed off from the neighboring links. The RL agent then learns implicitly the handoff traffic rate from each neighboring link for each class. This is the key feature of the RLBP algorithm.

Another feature that was added is action replay. This is a feature of Reinforcement Learning that feeds historical states, actions and rewards to the RL agent. The RL agent then learns based on these data that have been stored in a table, as if it faced the same state and chose the same action as in the past. Action replay makes better use of past experiences. This is crucial in an environment where experience is costly. In 'live' networks, bad experiences where QoS is severely violated can incur heavy penalties for the service provider, both in terms of costs as well as reputation. There needs to be a careful trade-off between faster learning and costly experiences. An RL agent learns best when it has a wide range of experiences, including bad ones. But a service provider may opt not to bear so much cost and restrict the learning range. This would inevitably lead to slower and perhaps less optimal learning, which may be suffice for the service provider.

For each agent RL_i, the RLBP algorithm is as follows (omitting the subscript i for clarity):

```
Initialize $\theta_0$ and $\sigma_0$ and set $\mu_0 = 1 + \frac{100}{1 + exp\left(\theta_0{}^T x_0\right)}$
Select    $y_0 \sim N(\mu_0, \sigma_0)$
At the end of every interval $n$,    $n \geq 1$
    Update    $\theta_n = \theta_{n-1} - \alpha_\theta \frac{(r_{n-1} - \hat{r}_{n-1})}{\hat{r}_{n-1}} (y_{n-1} - \mu_{n-1}) x_{n-1}$
              $\sigma_n = \sigma_{n-1} - \alpha_\sigma \frac{(r_{n-1} - \hat{r}_{n-1})}{\hat{r}_{n-1}} \frac{(y_{n-1} - \mu_{n-1})^2 - \sigma_{n-1}{}^2}{\sigma_{n-1}}$
              $\mu_n = 1 + \frac{100}{1 + exp\left(\theta_n{}^T x_n\right)}$
              $\hat{r}_n = \gamma r_{n-1} + (1 - \gamma)\hat{r}_{n-1}$
    Select    $y_n \sim N(\mu_n, \sigma_n)$
    Action replay:    $(n > 10)$
    For the last 10 historical sets $x_m$, $y_m$, $r_m$,    $m = n - 10$ to $n - 1$
    and       $\mu_m = 1 + \frac{100}{1 + exp\left(\theta_n{}^T x_m\right)}$
        Update    $\hat{r}_n = \gamma r_{m-1} + (1 - \gamma)\hat{r}_n$
                  $\theta_n = \theta_n - \alpha_\theta \frac{(r_{m-1} - \hat{r}_n)}{\hat{r}_n} (y_{m-1} - \mu_{m-1}) x_{m-1}$
```

where, x_n is the vector of the traffic state observed at $(n-1)^{th}$ time interval and y_n is the vector of WFQ weights to be set for the n^{th} time interval. μ_n is the mean used to set y_n. σ_n is the variance used to set y_n and θ_n is a Neural Network (NN) [63] parameter relating x_n and μ_n. r_n is the penalty received for QoS violations in the n^{th}

time interval and $\hat{r}_n$ is the cumulative penalty for QoS violations beyond assured levels. α_θ is the step size for θ_n and α_σ is the step size for σ_n.

x_n is a 9-dimensional state, comprising of the average traffic proportions of the three classes of traffic from the 2 incoming links as well as the neighboring link. Thus, the NN parameter θ_n would be a 3×9 matrix linking x_n to 3-dimensional μ_n. A measurement-based framework is used, where the average traffic proportions are computed as the proportion of traffic for a class with respect to the total traffic measured across each interval and are weighted averaged by a factor of 0.8. The Neural Network used to relate the state x_n and the action μ_n is a simple Multi-layer Perceptron (MLP) network [63] with 2 layers. The MLP network outputs a mean μ_n that lies in a range $1 \le \mu_n \le 101$. This is to simplify the implementation of y_n, which we set to be an integer between 1 and 100.

6.3.2 Penalty Function

The RLBP penalty function is similar to the one presented for RLDP in section 5.4.3 is given by

$$r = \sum_j [exp(c_{loss,j} \times l_{i,j}) + exp(c_{dly,j} \times d_{i,j})] \qquad (6.1)$$

For each traffic class j, $l_{i,j}$ is the percentage of packets loss exceeding the packet loss requirement at router i and all routers downstream from i, and $d_{i,j}$ is the proportion of traffic above the tolerance level for delay that passed through router i not meeting the delay bound. $c_{loss,j}$ is the weight given to $l_{i,j}$, and $c_{dly,j}$ is the weight given to $d_{i,j}$.

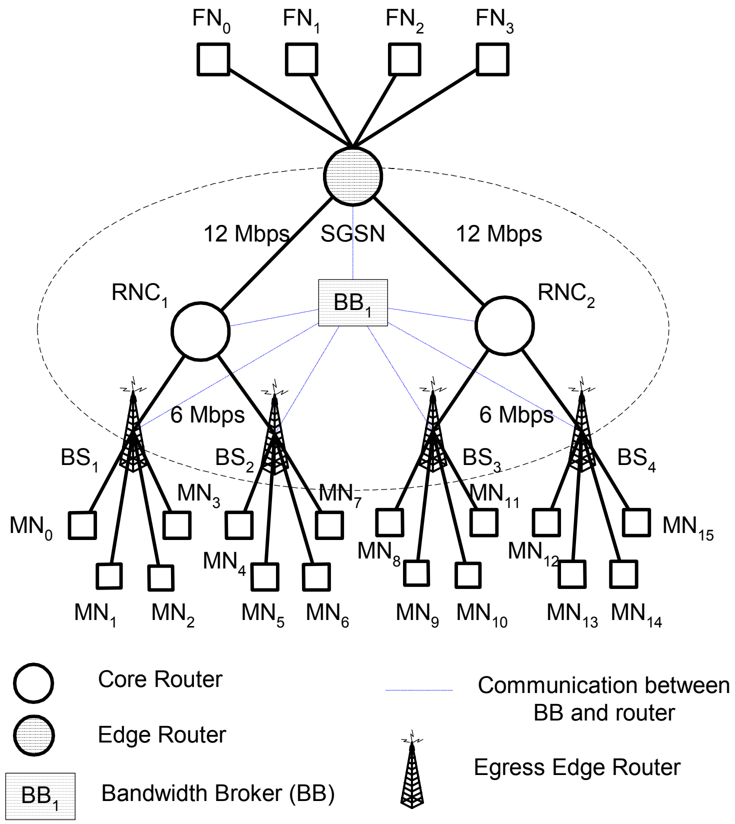

Figure 6.3: UTRAN Simulation Network Topology

6.3.3 Simulation and Results

6.3.3.1 Simulation Setup

ns-2 [8] DiffServ extensions were used for the simulation. The topology in Fig. 6.3, which is an expanded version of the topology in Fig. 6.2, was set up. Through our simulations, we show that RLBP is able to improve on static provisioning and a measurement-based provisioning method. The setup involves 20 nodes, of which 16 are mobile nodes MN. The fixed nodes FN act as sources only and the mobile nodes act as both sources and destinations. Sources from fixed nodes FN_0 to FN_3 have destinations MN_0 to MN_3 (attached to BS_0), MN_4 to MN_7 (attached to BS_1), MN_8

to MN_{11} (attached to BS_2) and MN_{12} to MN_{15} (attached to BS_3) respectively. Mobile nodes similarly send traffic to other mobile nodes attached to other base stations. The nodes attached to the SGSN are meant to represent an aggregate of traffic sources located beyond the UTRAN. The mobile nodes act as an aggregated source and destination as well; each node simulating all traffic of a certain class served by the base station. Thus, one mobile node may represent 20 EF class mobile users for example. The buffer lengths used for all routers are 5, 30 and 50 packets for EF, AF and BE respectively. These buffer lengths are chosen to be a reasonable trade-off between queuing latency and buffer overflow.

To simulate mobility, we do not actually move the mobile nodes. Since we are only concerned with how handoff traffic on new paths affect the provisioning, the mobility is simulated instead by shifting portions of traffic in the aggregated flow to neighboring cells with varying probabilities. This will cause a change in the route taken by that flow. For example, if a mobile source from MN_0 were to initially send traffic to destination MN_4, after a handoff, it could be sending traffic to destination MN_8 or MN_{12}, which are located in neighboring cells.

6.3.3.2 Traffic Characteristics

In the following simulations, traffic from all 3 PHB aggregates was generated on all links in the DiffServ domain in both the downstream and upstream directions. The traffic load was chosen such that different links were congested at different time intervals. The intervals were made long enough to simulate sustained congestion due to mobile nodes congregating at particular cells. This is well known as the *hotspot* problem in cellular networks. Hotspots can be caused by everyday and yet unexpected situations that are hard to predict, such as mobile users stuck with nowhere else to go during a sudden downpour.

TABLE 6.1: Characteristics of Traffic Sources

Source	Connection Inter-arrival Time (s)	Connection Holding Time (s)	ON Rate (kbps)
FN_0	3.0	15	64
FN_1	2.5	15	128
FN_2	5.0	15	300
FN_3	1.25	15	128
MN_0, MN_4, MN_8, MN_{12}	1.5	15	64
MN_1, MN_5, MN_9, MN_{13}	2.5	15	128
MN_2, MN_6, MN_{10}, MN_{14}	3.75	15	300
MN_3, MN_7, MN_{11}, MN_{15}	1.25	15	128

Table 6.1 gives the characteristics of the traffic sources used. Sources from MN_0 to MN_3 have identical characteristics to sources from MN_4 to MN_7, MN_8 to MN_{11} and MN_{12} to MN_{15} respectively. FN_0 and MN_0 are EF sources that represent delay bounded traffic that are in the UMTS conversational and interactive classes. FN_2 and MN_2 are AF sources that represent loosely delay bounded, high throughput traffic that are in the UMTS streaming class. Both EF and AF classes are modeled as exponential ON-OFF sources with same ON (500ms) and OFF (500ms) times. FN_3 and MN_3 are BE sources that represent non-bounded aggregated web traffic in the UMTS background class. They are modeled as pareto ON-OFF sources with the same ON and OFF times as the EF and AF traffic. These 3 source types run over UDP. The last source type FN_1 and MN_1 run over TCP, and are BE sources that represent non-bounded high throughput traffic like FTP traffic that are also under the UTMS

background class. They are CBR sources that consume any unused capacity on the link.

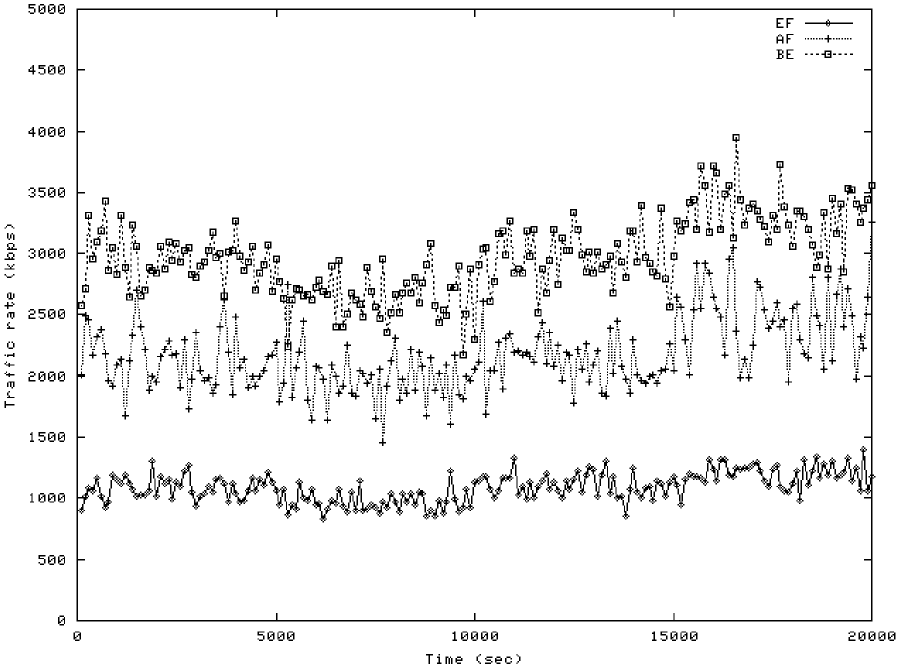

Figure 6.4: Traffic Entering Link RNC_1 - BS_1

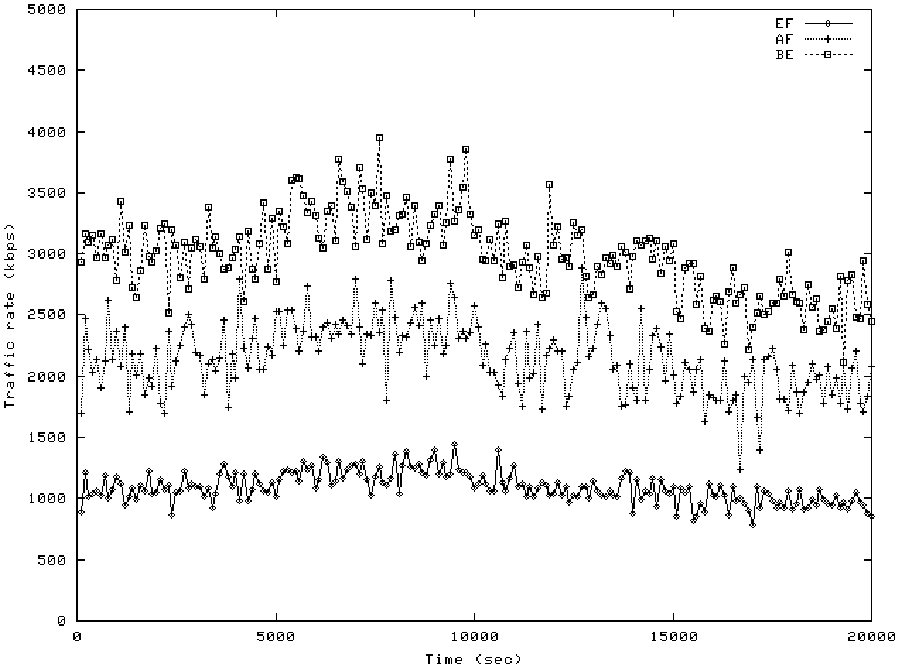

Figure 6.5: Traffic Entering Link RNC_2 - BS_3

The average amount of traffic generated from the UMTS core backbone network for EF, AF and BE are 640 kbps, 1.8 Mbps and 6 Mbps respectively. The average amount of traffic generated from each cell by the mobile nodes for EF, AF and BE are 960 kbps, 1.8 Mbps and 4.5 Mbps respectively. Fig. 6.4 and 6.5 shows the traffic patterns on links RNC_1-BS_1 and RNC_2-BS_3. The two traffic patterns are different due to the mobility pattern that was used for the experiments, which we describe in the next section.

6.3.3.3 Mobility Model

As mobile users move from one cell to another, the traffic originating from or destined to the mobile user that once flowed through a former path, now flows through a latter path as determined by the Mobile IP protocol. From an aggregate point of view, the aggregate flow of traffic on the former path is decreased and the aggregate flow of traffic on the latter path is increased. Since in our topology we have used a single mobile node to represent the aggregate of all mobile users of a class, the movement of the mobile user can be simulated by a change in source node to the respective mobile node attached to the latter cell's base station. Similarly, a change in destination node is needed for all flows that were destined for the former mobile node. For example, when a mobile user makes a handoff from BS_1 to BS_2, the flow with source-destination pair $\{MN_0, MN_8\}$ is changed to source-destination pair $\{MN_4, MN_8\}$, and the flow with source-destination pair $\{MN_8, MN_0\}$ is changed to source-destination pair $\{MN_8, MN_4\}$.

The rate of handoffs and the direction of handoff (simulating the direction of movement of the mobile user) can be controlled with the use of probabilities. A low probability of handoff results in a slow handoff rate and a congregation of mobile users is a result of having a higher probability of moving in a particular direction as compared to other directions. We have only simulated the changing of destination nodes and not the source nodes. However, the effects are similar. In the first 5000s, mobile nodes have a 0.5 probability of handing off and an equal probability of moving into any cell. In the second 5000s, mobile nodes congregate around BS_3 and BS_4, causing links RNC_2-BS_3 and RNC_2-BS_4 to be congested. The probability of handoff is doubled and the probability of handoff in the direction of BS_3 and BS_4 is increased to 0.9. In the third 5000s, the mobile nodes disperse equally again, and in the last 5000s, the mobile nodes congregate again, this time around BS_1 and BS_2.

6.3.3.4 Experimental Details

In our experiments, 3 different schemes are compared – the static provisioning scheme, the measurement-based scheme and RLBP. For the static provisioning scheme, we set up the WFQ weights to be static and equal throughout each experiment lasting 20,000s. The equal WFQ weights setting is based on over-provisioning EF and AF traffic by 100% and 50% over the average traffic rate that we used. (Note that this information is not available *a priori* in live networks). Abella [135] has written an argument for the use of over-provisioning over the use of DiffServ. For the measurement-based scheme and RLBP, we set up the experiment to initially begin with the same weights as in the static case. We also set μ_0 to be equal to the static setting (equal weights for all 3 classes). Both schemes then kick in after 1,500s, adjusting the WFQ weights in the routers adaptively based on the incoming traffic rates of the routers' links and the neighboring routers' links that might handoff traffic to it.

The measurement-based scheme was used as a comparison to see how RLBP fared compared to a simple ad-hoc scheme. The scheme takes traffic intensity measurements in the same way as RLBP does, and computes the bandwidth provisions for the next time interval based on the measured interval. Both EF and AF classes of traffic are provisioned by taking the summation of traffic measured at the incoming links to the node as well as the incoming links of the neighboring nodes and multiplying by an over-provisioning factor. EF and AF traffic was over-provisioned by a factor of 2 and 1.5 respectively. BE traffic is provisioned by taking the remainder bandwidth after EF and AF are provisioned. This scheme is expected to be over-conservative.

All simulations involving RLBP were run with 3 different random seeds for the Gaussian unit and the results were averaged. The time-step interval T chosen is 100s. γ was chosen to be 0.2. But unlike the RLAP and RLDP schemes, we experimented with the use of a non-constant α_μ and α_σ for RLBP. α_μ and α_σ were chosen to be $0.01/\hat{r}_n$ and $1/\hat{r}_n$ respectively. The rationale for selecting a function that is inversely proportional to the baseline penalty is to increase the factors α_μ and α_σ as the RL agent improves the provisioning. Thus, the RL agent adapts more quickly when it is more confident that it is on the right track. This improves the rate of learning, while not sacrificing smooth changes in provisioning in the initial learning phase, where the RL agent is still infant and unsettled.

6.3.3.5 Comparison between Static Provisioning, Measurement-based Dynamic Provisioning and RLBP

In the first experiment, the three schemes are compared against each other. The QoS requirements for EF and AF delay bounds were set to 12ms and 20ms respectively. The tolerance for percentage of packet delayed was set to 2% for EF and 5% for AF. The maximum packet loss for EF, AF and BE were set to be 2%, 5% and 20% respectively. The weights in the penalty function in equation (6.1) were set to 5.0 for all constraints to give equal weighting. The following tables summarize the QoS achieved for the last 10,000s of the simulation.

TABLE 6.2: QoS Achieved for Static Provisioning

Traffic Type	**Packet Loss (%)**	**Delay (%)**
EF	0.01	1.1
AF	6.5	8.2
BE	7.8	-

TABLE 6.3: QoS Achieved for Measurement-based Provisioning

Traffic Type	**Packet Loss (%)**	**Delay (%)**
EF	0.35	2.3
AF	0.57	0.9
BE	12.0	-

TABLE 6.4: QoS Achieved for RLBP

Traffic Type	**Packet Loss (%)**	**Delay (%)**
EF	0.48	1.96
AF	2.6	3.5
BE	10.3	-

We see that for static provisioning, the AF traffic did not meet the assured QoS level of 5% for packet loss and delay, while the measurement-based provisioning was able to. Because we have set AF requirements to be tight, relatively more bandwidth share has to be given to AF to ensure that the requirements are met; especially so to ensure that handoff traffic would not degrade the QoS experienced by the aggregate below the assured level. Fig. 6.6 and 6.7 show the packet loss and delay QoS of AF traffic over the last 10,000s.

We also observe that the measurement-based method was unable to meet the EF delay assurance level of 2%. This is despite over-provisioning for EF. In fact, the performance of AF traffic was overwhelmingly good. Therefore, we can conclude that it is not easy to determine the level of over-provisioning. Furthermore, provisioning should not only take into account the measured traffic, but also the relative tightness of the QoS requirements. RLBP has the ability to learn the optimal policy that takes these factors into account through appropriately designing the penalty function. Therefore, RLBP was able to meet the QoS requirements for all 3 classes.

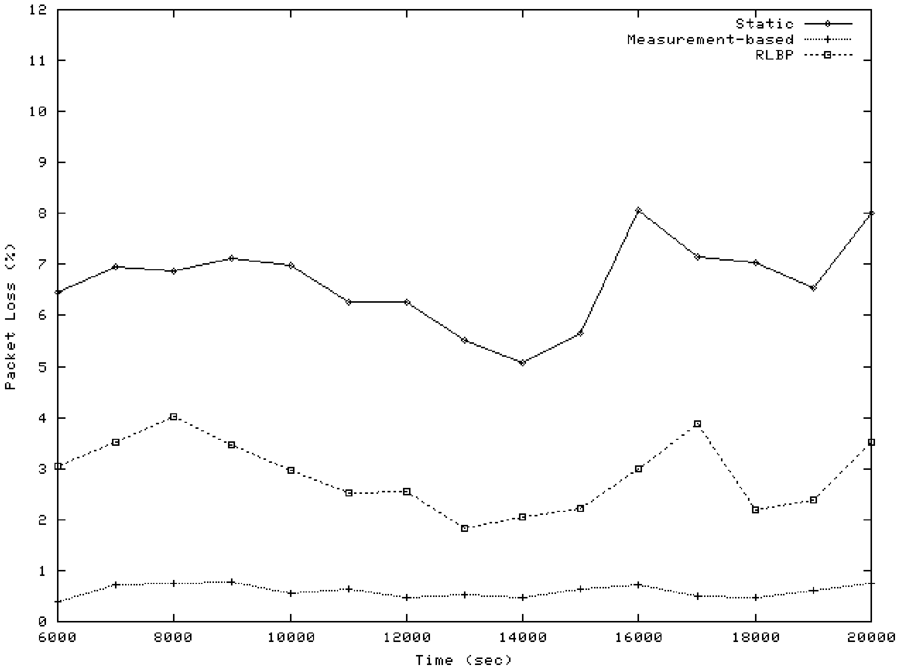

Figure 6.6: Percentage of AF Packet Loss at 1000s Intervals

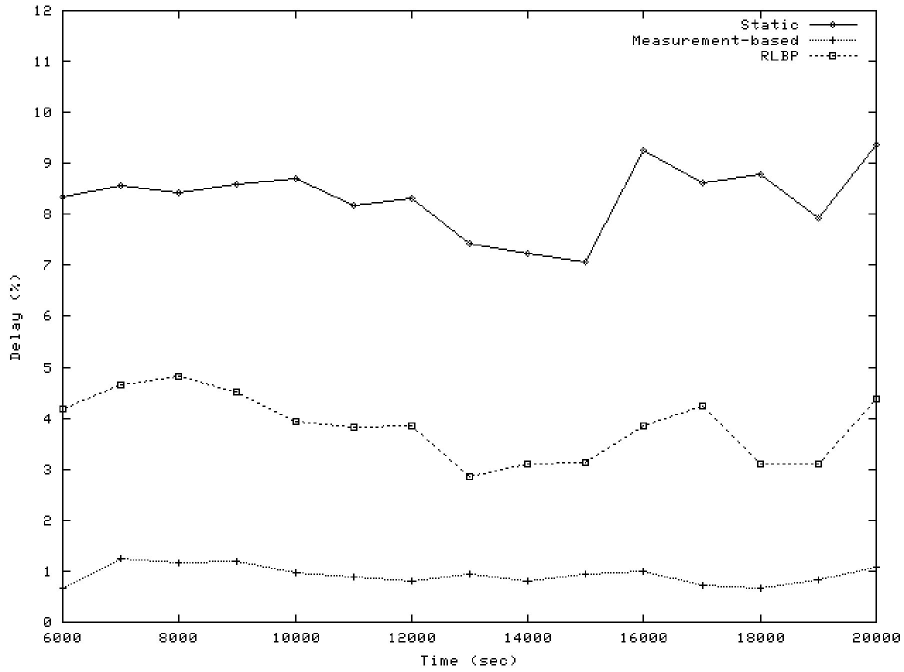

Figure 6.7: Percentage of AF Packets Delayed at 1000s Intervals

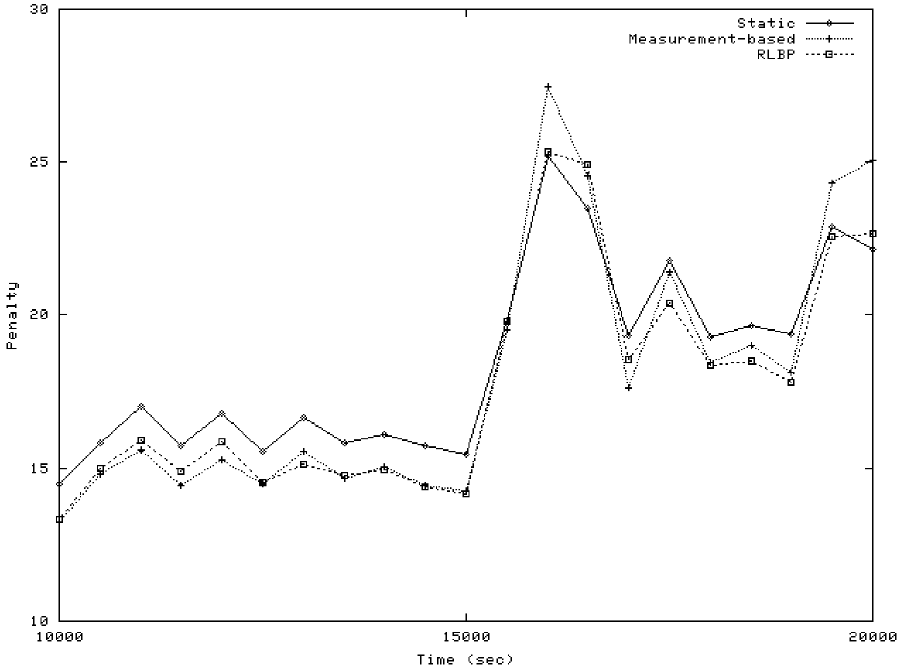

Figure 6.8: Penalty per 500s Interval for link $RNC_1 - BS_1$

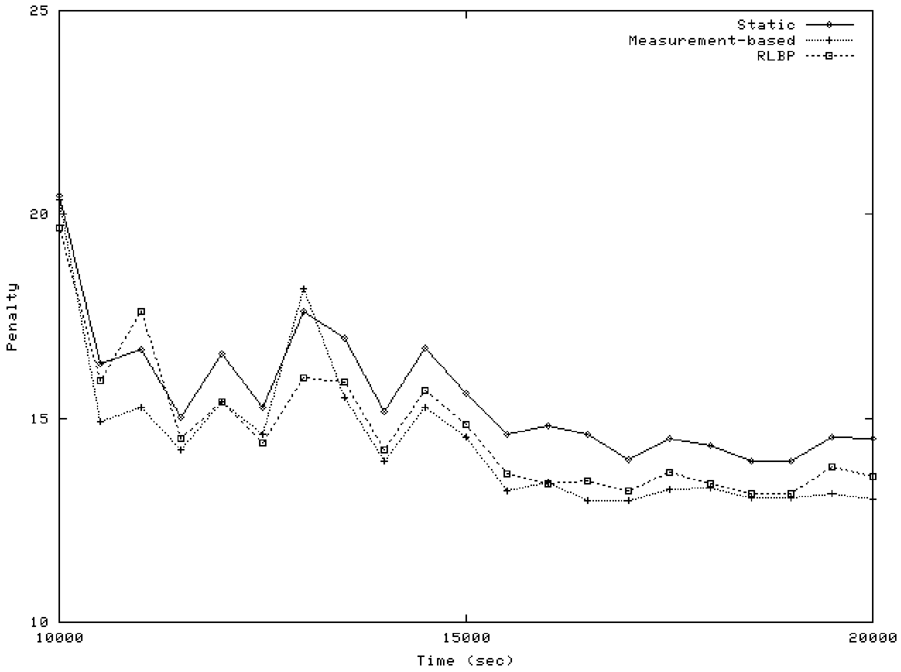

Figure 6.9: Penalty per 500s Interval for link $RNC_2 - BS_3$

Fig. 6.8 and 6.9 show the penalty feedback for RL agents administering links $RNC_1 - BS_1$ and $RNC_2 - BS_3$ over the last 10,000s. The heavy penalty seen in Fig. 6.8 in the last 5,000s is caused by the congregation of traffic during that interval. RLBP was able to perform relatively better during that interval. In Fig. 6.9, we can see evidence that over a relatively longer period of similar traffic, RLBP is able to reduce the penalty to a near-optimal level. Therefore, is a service provider were to implement RLBP over a long period, in the long run, he would have a dynamic and automatic bandwidth provisioning system that requires little human intervention, other than to feed new data into the bandwidth broker's database when new SLAs are signed. From the figures, it can be concluded that RLBP learnt a policy that was in between the 2 extremes of static provisioning and measurement-based provisioning. The policy was able to balance the bandwidth provisions such that no class would suffer QoS violations. The process of tuning the bandwidth provisioning is similar to expert learning. Other methods of provisioning are either limited by prior experience or tight traffic control at the edges, which offer lower bandwidth utilization in exchange for QoS guarantees.

TABLE 6.5: QoS Achieved for Static Provisioning

Traffic Type	**Packet Loss (%)**	**Delay (%)**
EF	0.01	0.01
AF	6.5	4.5
BE	7.8	-

TABLE 6.6: QoS Achieved for Measurement-based Provisioning

Traffic Type	**Packet Loss (%)**	**Delay (%)**
EF	0.35	0.11
AF	0.57	0.23
BE	12.0	-

TABLE 6.7: QoS Achieved for RLBP

Traffic Type	**Packet Loss (%)**	**Delay (%)**
EF	0.92	0.51
AF	9.2	5.9
BE	5.8	-

6.3.3.6 Comparison under Relaxed QoS Requirements

In our second experiment, we seek to demonstrate how RLBP is able to adapt to a different set of QoS requirements through a change in the penalty function. EF and AF requirements are now relaxed. Instead, BE requirements are tightened, possibly to enable background traffic users to enjoy better latency response times. This would help balance customer satisfaction; something that may be overlooked by service providers eager to guarantee good service to high value customers.

The EF delay bound was set to 15ms, up from 12ms, and the AF delay bound was increased to 25ms, up from 20ms. The EF and AF delay tolerance and packet loss requirements were increased to 5% and 10% respectively and the BE packet loss requirement was reduced to 10%, down from 20%. These changes in QoS requirements are reflected in a corresponding change in the penalty function, which governs the learning process of the RL agent. The relative significance of each QoS requirement can be altered by changing the weights attached to each component in the function. The service provider could possibly decide the weights based on which customer market he deems to be more important. This is turn could be based on the pricing of services and the margin of profit. To give more significance to BE customer satisfaction, the BE packet loss weight was changed to 10.0, up from 5.0. The other weights remained unchanged and therefore are relatively less significant.

Tables 6.5, 6.6 and 6.7 summarize the end-to-end QoS achieved for the three schemes. We see that with these new QoS requirements, static provisioning was able to meet all the requirements, while the measurement-based method was unable to meet the 10% assurance level for BE packet loss. This shows that the QoS requirements have been relaxed so much so that static over-provisioning is sufficient, and any degradation in QoS caused by handoffs can be tolerated under the new assurance levels. By being overly conservative in provisioning for handoff traffic, the measurement-based scheme compromised in QoS for BE traffic. Once again, it proves that it is difficult to determine the correct mix of bandwidth provisions. We see that the RLBP scheme is able to adapt and learn a new policy based on the new QoS requirements. Thus, it was able to satisfy all the requirements. Fig. 6.10 gives a closer look at the BE packet loss levels over the last 10,000s, and Fig. 6.11 and 6.12 show the penalty levels for RL agents administering links RNC_1 - BS_1 and RNC_2 - BS_3 over the interval 5,000s to 20,000s.

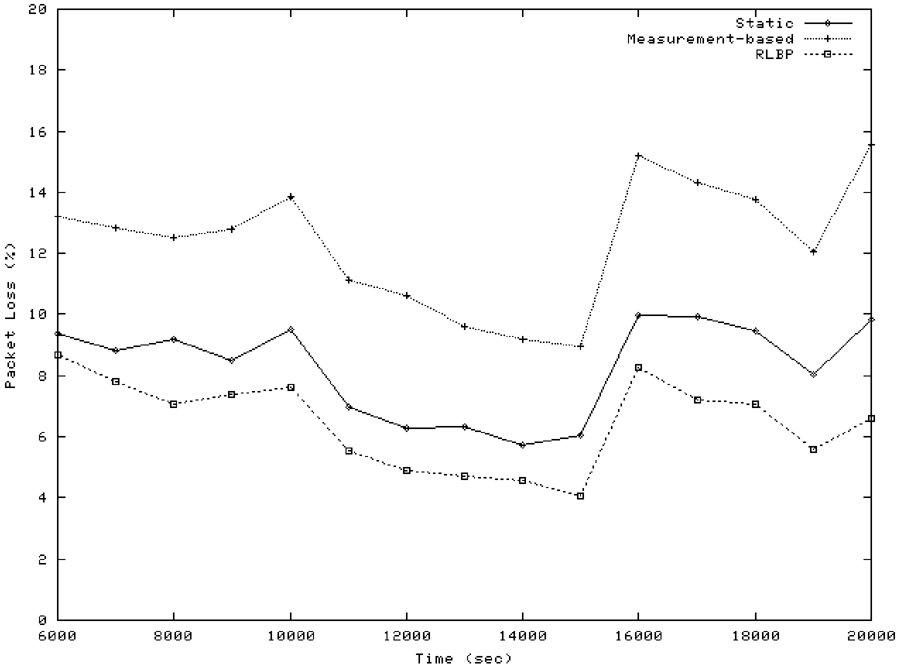

Figure 6.10: Percentage of BE Packet Loss at 1000s Intervals

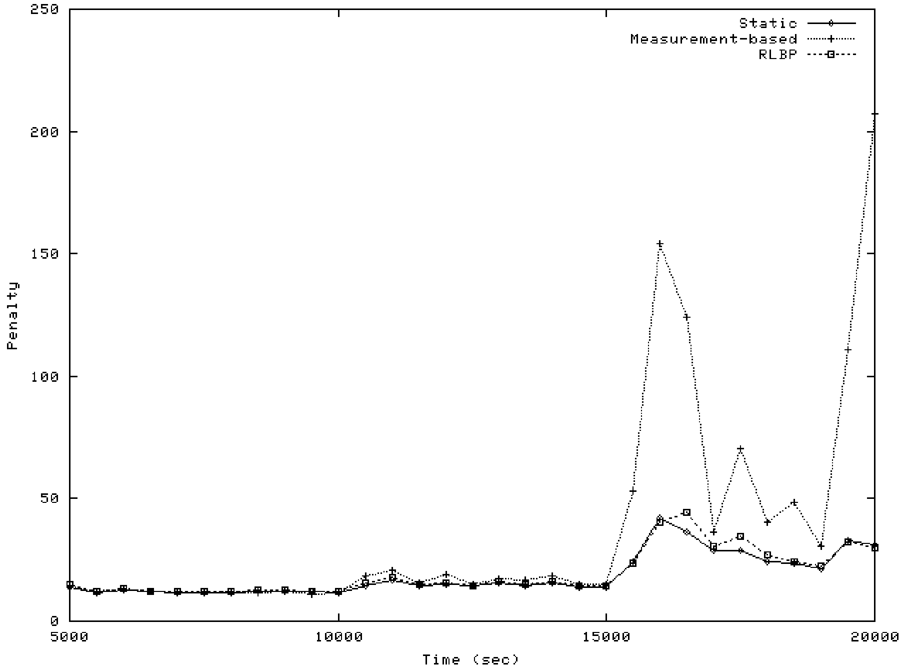

Figure 6.11: Penalty per 500s Interval for link RNC_1 - BS_1

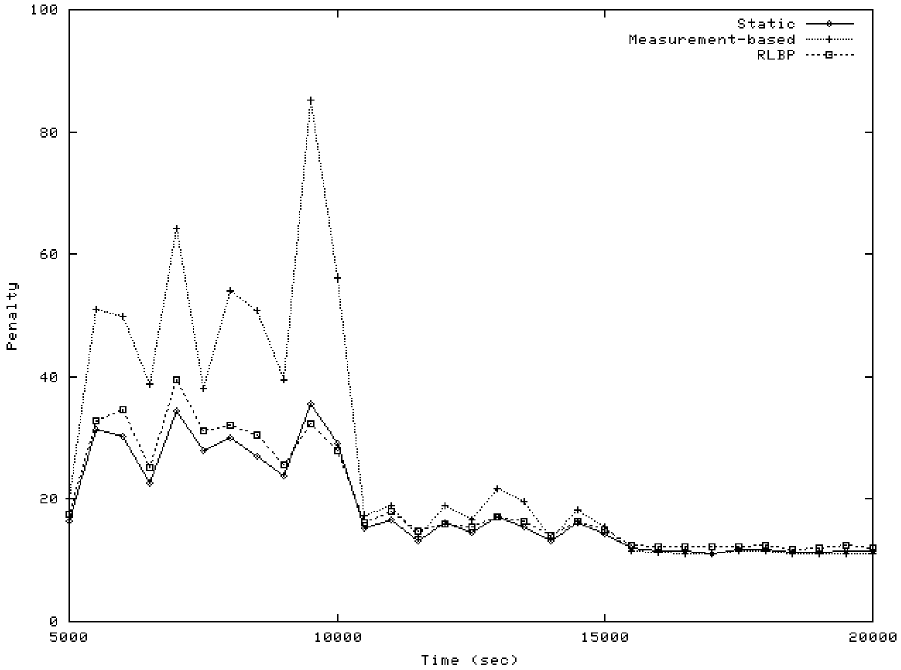

Figure 6.12: Penalty per 500s Interval for link $RNC_2 - BS_3$

Fig. 6.10 shows that RLBP was able to keep BE packet loss constantly under the assured 10% level, while still meeting QoS requirements for EF and AF traffic even in periods of heavy handoffs. Fig. 6.11 and 6.12 show even more evidence that RLBP does intelligent enough not to react in periods of heavy congestion, during the intervals 15,000s to 20,000s and 5,000s and 10,000s respectively, if it is not necessary. The RL agents judge this through the penalty feedback it receives at every interval. Since QoS is not violated, the agent determines that the there is not much adjustment to be made.

From the results of the two contrasting experiments, we can conclude that the RLBP bandwidth provisioning scheme is intelligent enough to balance the amount of bandwidth provisioned to each class such that QoS assurance levels can be maintained in the face of heavy congestion due to handoff traffic. The scheme learns on its own and improves over time. This replaces the need for expert administrators to adjust bandwidth provisioning controls or overly conservative traffic conditioners at the edge to curb traffic. The RLBP scheme is efficient and can be customized to the service provider’s requirements through the use of the penalty function. The construction of the penalty function together with the RL learning algorithm is the key to the success of the scheme.

CHAPTER 7

CONCLUSION

The work presented in this thesis was motivated by the need to develop a bandwidth provisioning scheme for the UMTS core network that is intelligent, efficient and objective at the same time. The provisioning of bandwidth in the UMTS core network is essential for end-to-end quality of service to be achieved in global converged networks. In order for service providers to offer a diversity of services ranging from video-conferencing via mobile phones, to video-streaming on in-vehicle entertainment systems, to m-commerce transactions on personal digital assistants, to data downloads on laptops, different levels of QoS have to be maintained. The different levels of QoS are achieved through the UMTS adoption of the Differentiated Services model. By implementing DiffServ on the IP-based UMTS core network, a diverse range of services can be provisioned in an aggregated manner that is efficient and scalable. Due to the nature of mobile services and the architecture of mobile networks, bandwidth provisioning in UMTS core networks is very different from that done in fixed backbone data networks. Thus, new methods of provisioning are required.

The solution presented in this thesis comprises of two components that are meant to be implemented in the UMTS core backbone network and the UTRAN respectively in a concurrent manner. The need to use two different schemes comes about because of the different topology and traffic dynamism of the two network portions. The emphasis is different in each scheme that we present and provides the service provider flexibility in implementation. The work done in this thesis greatly fills in the gap for a simple, efficient and effective provisioning scheme for UMTS core networks.

7.1 CONTRIBUTION OF THESIS

This thesis presents a scheme for bandwidth provisioning in IP-based DiffServ UMTS core networks. The algorithms used in the scheme are capable of provisioning bandwidth to objectively meet diverse QoS requirements contracted through SLAs between service providers and mobile service subscribers without sacrificing bandwidth efficiency. The scheme intelligently (without the need for expert knowledge) adjusts bandwidth proportions allocated to each service class on a time interval basis making it scalable and easy to implement. The intelligence is achieved through reinforcement learning agents, which are able to develop policies that map traffic conditions to respective bandwidth provisions through reward and penalty feedback.

In chapter 3, adaptive bandwidth provisioning has been shown to be effective in attaining quality of service levels in terms of latency, packet loss and throughput. It was argued that weighted fair queuing provides the most bandwidth efficient method of provisioning bandwidth. Through the formulation of the bandwidth provisioning problem as an optimization problem, it was shown that bandwidth provisioning as a continuous time problem is a *hard* problem. Furthermore, the problem cannot be solved even after discretization due to the unpredictable nature of traffic.

In chapter 4, the bandwidth provisioning problem was re-formulated as a reinforcement learning problem. In this way, the problem can be solved through an iterative method that progressively develops an approximately optimal solution. Due to the continuous nature of traffic parameters and bandwidth provision settings, a continuous state-action space reinforcement learning method was chosen. The use of a continuous state-action space reinforcement learning method is a pioneering work in the field reinforcement learning for network control. This work is important in the sense that it provides guidance for other continuous space network control problems to be solved using reinforcement learning.

In chapter 5, two schemes for bandwidth provisioning in UMTS core backbone networks are presented. The first scheme called Reinforcement Learning-based Adaptive Provisioning (RLAP) introduces a novel way of pricing services. A 3-tier usage-based pricing model is used to promote better utilization of bandwidth. The pricing model is attractive to both users as well as providers; as users pay for only what they use and providers can capitalize on greater traffic multiplexing. In the thesis, 3 tiers are used to differentiate 3 different user requirements. When combined with a penalty refund, providers are able to have some leeway in provisioning services. Users are also kept happy as a high service level is still maintained as service providers have to pay out penalties for breeches in QoS. RLAP makes use of this pricing plan to compute the reward feedback for the RL agents. The algorithm is based on *REINFORCE* Gaussian units and makes use of a gradient ascent iterative method. The aim of the RLAP scheme is to maximize revenue.

The second scheme presented called Reinforcement Learning-based Dynamic Provisioning (RLDP) is different from RLAP as it aims to minimize QoS violations and to assure a level of QoS contracted in the SLA. Stochastic Real-Valued units are used in place of Gaussian units to provide better adaptation to traffic conditions, and average buffer occupancy ratio is included as part of the input to better control QoS. The scheme is independent of the pricing strategy and can be implemented based on SLAs commonly used by service providers. The bandwidth proportions of each class are balanced such that all the classes can meet the assured level of QoS. An exponential penalty function is used to discourage QoS violations and guide the RL agents towards better QoS for all classes. In simulations, both schemes were superior to static over-provisioning schemes, which is not adaptable. The RLAP scheme was shown to be able to adapt to changing traffic conditions and the RLDP scheme was shown to be able to adapt to different sets of QoS requirements. Both RLAP and RLDP are also arguably better than other measurement-based admission control methods, adaptive control methods, traffic prediction methods and pricing methods, since they are more bandwidth efficient, able to balance various QoS classes to meet specific QoS targets, and do not require expert knowledge.

Chapter 6 details a scheme for bandwidth provisioning in UTRANs. The scheme has a vastly different paradigm to commonly-used reservation-based methods. While reservation-based methods have to contend with handoff blocking and poor bandwidth utilization to assure QoS, a weighted fair queuing-based provisioning method offers effectively zero blocking and complete sharing of bandwidth. Bandwidth provisioning methods also focus on maintaining QoS in terms of latency and packet loss bounds, unlike reservation-based schemes, which focus on maintaining low handoff blocking probability. The change in paradigm is a much needed one as mobile networks move away from circuit-switched to packet-switched architectures (as the one in UMTS). The RLDP scheme is modified for use in the UTRAN. The changes are made to accommodate handoff traffic, which is a more pertinent issue in the UTRAN. The scheme called Reinforcement Learning-based Bandwidth Provisioning (RLBP) takes into consideration the traffic in neighboring links as part of the input. RLBP was shown to be able to adapt to different sets of QoS requirements. The static over-provisioning scheme was not able to adapt to the changing amount of handoff traffic. Both schemes also lack sensitivity to different

QoS requirements. It was shown that determining the right amount of over-provisioning was a difficult task, which cannot be done using ad-hoc means.

When the bandwidth provisioning scheme for the UMTS core backbone network is combined with the scheme for the UTRAN, end-to-end provisioning in the UMTS network can be achieved. The complete solution proposed enables service providers to objectively provision bandwidth to meet service level agreements and at the same time maximize bandwidth utilization for greater profitability. The solution is simple to implement through installation in a bandwidth manager and fits in well with DiffServ-capable UMTS networks. The solution also provides flexibility for service providers to modify their pricing and QoS levels to suit customer demands while requiring little to be done at the network level, since the solution proposed has the inherent ability to learn and build new policies on-the-fly.

7.2 RECOMMENDATION FOR FUTURE WORK

The algorithms presented in this thesis are all novel and therefore can be further improved. One area of improvement is through the use of a more integrated learning environment. The reinforcement learning agents in the presented algorithms work in a distributed fashion. By having some form of collaborative learning, faster convergence and a more optimal solution can be achieved.

Although bandwidth provisioning may be sufficient in providing QoS, an efficient buffer management scheme would enhance the control of QoS. Droptail queues have been used in our schemes, but a reinforcement learning-based RED (random early detection) [94] buffer management scheme that adaptively adjusts RED parameters would complete the QoS control problem in a DiffServ network. By controlling the RED parameters adaptively, the queue length can be managed so as to directly control queuing delay and packet drop probabilities. Using such a scheme could effective provide QoS control at a packet level rather than at a flow level.

Another area not explored in this thesis is the application of provisioning to Multi-Protocol Label Switching (MPLS) networks [136]. The use of MPLS in 3G networks have been proposed [137,138] as a way to manage mobility and to provision for QoS. Instead of provisioning for aggregate classes, RL-based provisioning can be used to provision for label-switched paths (LSP). This is particularly useful in provisioning constraint-based routed LSPs (CR-LSP).

As 3rd Generation implementation gets underway, researchers are looking towards designing 4th Generation (4G) mobile networks [139,140]. In 4G networks, various types of wireless access environment will be connected together in a coherent heterogeneous network. These include broadband wireless LAN environments, wireless personal area networks, wireless ad-hoc networks, wireless WAN and satellite networks. This would require very complex QoS management, which is a key component in the 4G framework. Where there are QoS resources to be managed, a reinforcement learning solution can be applied to intelligent provisioning in different kinds of environments. The advantage of using reinforcement learning-based solutions is that they are adaptable and can learn based on any parameters in the environment to achieve a whole range of goals. For example, an extension to this work could be done for provisioning of 4G core networks, where the edges of the core network are attached to various types of radio access networks, with different bandwidths, topology and access technologies.

REFERENCES

[1] Universal Mobile Telecommunications Service (UMTS) Forum, *http://www.umts-forum.org*.

[2] Third Generation Partnership Project (3GPP), *http://www.3gpp.org*.

[3] Third Generation Partnership Project 2 (3GPP2), *http://www.3gpp2.org*.

[4] G. Patel, and S. Dennett, "The 3GPP and 3GPP2 movements toward an all-IP mobile network", *IEEE Personal Communications*, 7(4), pp. 62-64, Aug 2000.

[5] International Mobile Telecommunications– 2000 (IMT-2000), *http://www.itu.int/ home/imt.html*.

[6] 3GPP TS 23.002 v3.6.0, "Network Architecture (Release 1999)", Sep 2002.

[7] 3GPP TS 23.002 v4.6.0, "Network Architecture (Release 4)", Dec 2002.

[8] 3GPP TS 23.002 v5.10.0, "Network Architecture (Release 5)", Mar 2003.

[9] IETF Differentiated Services (DiffServ) Working Group, *http://www.ietf.org/ html.charters/OLD/diffserv-charter.html*.

[10] 3GPP TS 23.107 v5.8.0, "QoS Concept and Architecture (Release 5)", Mar 2003.

[11] F. Agharebparast, and V.C.M. Leung, "QoS Support in the UMTS/GPRS Backbone Network Using DiffServ", Proc. of *IEEE Globecom 2002*, Nov 2002.

[12] H. Chaskar, and R. Koodli, "MPLS and DiffServ for UMTS QoS in GPRS Core Network Architecture". Proc. of *INET 2001*, Jun 2001.

[13] A. Tuoriniemi, G.A.P. Eriksson, N. Karlsson, and A. Mahkonen, "QoS Concepts for IP-based Wireless Systems", Proc. of *Third International Conference on 3G Mobile Communication Technologies 2002*, May 2002.

[14] M. Ricardo, J. Dias, G. Carneiro, and J. Ruela, "Support of IP QoS over UMTS networks", Proc. of *IEEE PIMRC 2002*, Sep 2002.

[15] S.I. Maniatis, E.G. Nikolouzou, and I.S. Venieris, "QoS Issues in the Converged 3G Wireless and Wired Networks", *IEEE Communications Magazine*, 40(8), pp. 44-53, Aug 2002.

[16] H. Hameleers, and C. Johansson, "IP Technology in WCDMA/GSM Core Networks", *Ericsson Review No.1, 2002*.

[17] S. Christensen, "Voice over IP Solutions", *http://www.juniper.net/solutions/ literature/white_papers/*, *White Paper, Juniper Networks,* Jun 2001.

[18] TIA TR-41, "Voice over IP Standards"

[19] IETF Session Initiation Protocol (SIP) Working Group, http://www.ietf.org/html.charters/sip-charter.html.

[20] ITU H.323, "Packet-based Multimedia Communications Systems"

[21] 3GPP TR 25.933 v5.3.0, "IP Transport in UTRAN (Release 5)", Jun 2003.

[22] IETF IP Routing for Wireless/Mobile Hosts (Mobile IP) Working Group, http://www.ietf.org /html.charters/mobileip-charter.html.

[23] S. Dixit, Y. Guo, and Z. Antoniou, "Resource Management and Quality of Service in Third Generation Wireless Networks", *IEEE Communications Magazine*, 39(2), pp. 125-133, Feb 2001.

[24] J. Kalliokulju, "Quality of Service Management Functions in 3rd Generation Mobile Telecommunication Networks", Proc. of *IEEE WCNC 1999*, Sep 1999.

[25] M.L.F. Grech, M. Torabi, and M.R. Unmehopa, "Service Control Architecture in the UMTS IP Multimedia Core Network Subsystem", Proc. of *Third International Conference on 3G Mobile Communication Technologies 2002*, May 2002.

[26] N. Dimitriou, R. Tafazolli, and G. Sfikas, "Quality of Service for Multimedia CDMA", *IEEE Communications Magazine*, 38(7), pp. 88-94, Jul 2000.

[27] D. Lister, S. Dehghan, R. Owen, and P. Jones, "UMTS Capacity and Planning Issues", Proc. of *First International Conference on 3G Mobile Communication Technologies 2000*, May 2000.

[28] K. Parsa, S.S. Ghassemzadeh, and S. Kazeminejad, "Systems Engineering of Data Services in UMTS W-CDMA Systems", Proc. of *IEEE ICC 2001*, Jun 2001.

[29] D. Goderis, et al, "Service Level Specification Semantics, Parameters and Negotiation Requirements", draft-tequila-diffserv-sls-00.txt, *IETF Internet Draft*, Nov 2000.

[30] R. Neilson, et al, "A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment", *Internet2 Qbone BB Advisory Council*, Aug 1999.

[31] B. Davie, et al, "An Expedited Forwarding PHB", *RFC 3246*, Mar 2002.

[32] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group", *RFC 2597*, Jun 1999.

[33] A. Demars, S. Keshav, and S. Shenker, “Analysis and Simulation of a Fair Queuing Algorithm”, Proc of *ACM SIGCOMM ’89*, Sep 1989.

[34] S.J. Golestani, “A Self-clocked Fair Queuing Scheme for Broadband Applications”, Proc. of *IEEE INFOCOM '94*, Apr 1994.

[35] M. Shreedhar, and G. Varghese, “Efficient Fair Queuing Using Deficit Round Robin”, Proc. of *ACM SIGCOMM '95*, Sep 1995.

[36] J.C.R. Bennett, and H. Zhang, “WF2Q: Worst-case Fair Weighted Fair Queuing”, Proc. of *IEEE INFOCOM '96*, Mar 1996.

[37] P. Goyal, H.M. Vin, and H. Chen, “Start-time Fair Queuing: A Scheduling Algorithm for Integrated Services”, Proc. of *ACM SIGCOMM ‘96*, Aug 1996.

[38] S.I. Maniatis, E.G. Nikolouzou, and I.S. Venieris, “QoS Issues in the Converged 3G Wireless and Wired Networks”, *IEEE Communications Magazine*, 40(8), pp. 44-53, Aug 2002.

[39] Internet2 Qbone BB Advisory Council, *http://qbone.internet2.edu/bb/*.

[40] IETF Simple Network Management Protocol (SNMP) Working Group, *http://www.ietf.org /html.charters/OLD/snmp-charter.html*.

[41] IETF IP Performance Metrics (IPPM) Working Group, *http://www.ietf.org/html.charters/ippm-charter.html*.

[42] Network Measurements Working Group (NMWG), *http://www-didc.lbl.gov/NMWG/*.

[43] MCI Service Level Agreements, *http://global.mci.com/uunet/terms/sla/*.

[44] C. Filsfils, and J. Evans, “Engineering a Multiservice IP Backbone to Support Tight SLAs”, *Computer Networks*, 40(1), pp. 131-148, Sep 2002.

[45] S. Wang, D. Xuan, R. Bettati, and W. Zhao, “Providing Absolute Differentiated Services for Real-time Applications in Static-priority Scheduling Networks”, Proc. of *IEEE INFOCOM 2001*, Apr 2001.

[46] A. Kos, B. Klepec, and S. Tomazic, “Real-time Application Performance in Differentiated Services Network”, Proc. of *IEEE TENCON 2001*, Aug 2001.

[47] T. Ferrari, G. Pau, and C. Raffaelli, “Measurement Based Analysis of Delay in Priority Queuing”, Proc. of *IEEE Globecom 2001*, Nov 2001.

[48] T. Ferrari, and P. Chimento, “Measurement-based Analysis of Expedited Forwarding PHB Mechanisms”, Proc. of *IEEE IWQoS 2000*, Jun 2000.

[49] T. Bonald, A. Proutiere, J.W. Roberts, “Statistical Performance Guarantees for Streaming Flows using Expedited Forwarding”, Proc. of *IEEE INFOCOM 2001*, Apr 2001.

[50] S. Floyd, and V. Jacobson, “Link-sharing and Resource Management Models for Packet Networks”, *IEEE Trans. on Networking*, 3(4), Aug 1995.

[51] S.C. Borst, and D. Mitra, “Virtual Partitioning for Robust Resource Sharing: Computational Techniques for Heterogeneous Traffic”, *IEEE JSAC*, 16(5), Jun 1998.

[52] C. Dou, and F-C. Ou, “Performance Study of Bandwidth Reallocation Algorithms for Dynamic Provisioning in Differentiated Services Networks”, *Computer Communications*, 24(14), pp. 1472-1483, Aug 2001.

[53] S.K. Biswas, S. Ganguly, and R. Izmailov, “Path Provisioning for Service Level Agreements in Differentiated Services Networks”, Proc. of *IEEE ICC 2002*, Apr 2002.

[54] S. Tong, D. Hoang, and O. Yang, “Bandwidth Allocation and Preemption for Supporting Differentiated-Service-Aware Traffic Engineering in Multi-service Networks”, Proc. of *IEEE ICC 2002*, Apr 2002.

[55] S. Bakiras, and V.O.K. Li, “Efficient Resource Management for End-to-end QoS Guarantees in Diffserv networks”, Proc. of *IEEE ICC 2002*, Apr 2002.

[56] H. Shimonishi, I. Maki, T. Murase, and M. Murata, “Dynamic Fair Bandwidth Allocation for DiffServ Classes”, Proc. of *IEEE ICC 2002*, Apr 2002.

[57] J.M. Mao, W.M. Moh, and B. Wei, “PQWRR Scheduling Algorithm in Supporting of DiffServ”, Proc. of *IEEE ICC 2001*, Jun 2001.

[58] J.Y. Le Boudec, and P. Thiran, *Network Calculus*, Springer Verlag LNCS 2050, Jun 2001.

[59] A. Charny, and J.Y. Le Boudec, “Delay Bounds in a Network with Aggregate Scheduling”, Proc. of *QoFIS 2000*, Sep 2000.

[60] R. Sutton, and A. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.

[61] D.P. Bertsekas, and J.N. Tsitsiklis, *Neuro-dynamic programming*, Athena Scientic, Belmont, MA, 1996.

[62] L. Kaelbling, M. Littman, and A. Moore, “Reinforcement Learning: A Survey”, *Journal of Artificial Intelligence Research*, vol. 4, pp. 237-285, May 1996.

[63] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd edition, Prentice Hall, Upper Saddle River, NJ, 1999.

[64] P. Marbach, O. Mihatsch, and J.N. Tsitsiklis, “Call Admission Control and Routing in Integrated Service Networks Using Neuro-Dynamic Programming”, *IEEE JSAC*, 18(2), pp. 197-208, Feb 2000.

[65] P. Marbach, O. Mihatsch, M. Schulte and J.N. Tsitsiklis, “Reinforcement Learning for Call Admission Control and Routing in Integrated Service Networks”, Proc. of *Advances in Neural Information Processing Systems*, Dec 1998.

[66] H. Tong, and T.X. Brown, “Adaptive Call Admission Control under Quality of Service Constraints: A Reinforcement Learning Solution”, *IEEE JSAC*, 18(2), pp. 209-221, Feb 2000.

[67] T.X. Brown, H. Tong, and S. Singh, “Optimizing Admission Control while Ensuring Quality of Service in Multimedia Networks via Reinforcement Learning”, Proc. of *Advances in Neural Information Processing Systems*, Dec 1999.

[68] C.J.C.H. Watkins, and P. Dayan, “Q-Learning”, *Machine Learning*, vol. 8, pp. 279-292, 1992.

[69] A.F. Atlasis, and A.V. Vasilakos, “LB-SELA: Rated-Based Access Control for ATM Networks”, *Computer Networks and ISDN Systems*, 30(1998), pp. 963-980, 1998.

[70] S. Singh, and D. Bertsekas, “Reinforcement Learning for Dynamic Channel Allocation in Cellular Telephone Systems”, Proc. of *Advances in Neural Information Processing Systems*, Dec 1996.

[71] J. Nie, and S. Haykin, “A Dynamic Channel Assignment Policy through Q-Learning”, *IEEE Trans. on Neural Networks*, 10(6), pp. 1443-1455, Nov 1999.

[72] S. Senouci, and G. Pujolle, “Dynamic Channel Assignment in Cellular Networks: A Reinforcement Learning Solution”, Proc. of *IEEE ICT 2003*, Feb 2003.

[73] C.K. Tham, and Y. Liu, "Minimizing Transmission Costs through Adaptive Marking in Differentiated Services Networks", Proc. of *IEEE MMNS 2002*, Oct 2002.

[74] Y. Liu, C.K. Tham, and T.C.K. Hui, “MAPS: A Localized and Distributed Adaptive Path Selection Scheme in MPLS Networks”, Proc. of *IEEE HPSR 2003*, Jun 2003.

[75] R.S Sutton, “Generalization in Reinforcement Learning: Successful Examples using Sparse Coarse Coding”, Proc. of *Advances in Neural Information Processing Systems*, Dec 1995.

[76] J.C. Santamara, R.S. Sutton, and A. Ram, “Experiments with Reinforcement Learning in Problems with Continuous State and Action Spaces”, *Adaptive Behavior*, 6(2), pp. 163-218, 1998.

[77] G.A. Rummery, *Problem solving with reinforcement learning*, PhD thesis, Cambridge University, 1995.

[78] C.K. Tham, *Modular On-Line Function Approximation for Scaling Up Reinforcement Learning*, PhD thesis, Cambridge University, England, 1994.

[79] R.J. Williams, “Simple Statistical Gradient-following Algorithms for Connectionist Reinforcement Learning”, *Machine Learning*, 8(3), pp. 229-256, 1992.

[80] V. Gullapalli, *Reinforcement Learning and its Application to Control*, PhD thesis, University of Massachusetts, Amherst, MA, 1982.

[81] B. Widrow, and M.E. Hoff, “Adaptive Switching Circuits”, *IRE WESCON Convention record Part IV*, pp. 96-104, 1960.

[82] IETF Integrated Services (IntServ) Working Group, http://www.ietf.org/html.charters/intserv-charter.html.

[83] The ATM Forum, http://www.atmforum.com/.

[84] S. Jamin, P.B. Danzig, S.J. Shenker, and L. Zhang, “A Measurement-based Admission Control Algorithm for Integrated Service Packet Networks”, *IEEE Trans. on Networking*, 5(1), pp. 56-70, Feb 1997.

[85] R. Gibbens, and F. Kelly, “Measurement-based Connection Admission Control”, Proc. of 15th ITC, Jun 1997.

[86] J. Qiu, and E.W. Knightly, “QoS Control via Robust Envelope-based MBAC”, Proc. of *IEEE IWQoS 98*, May 1998.

[87] L .Breslau, S. Jamin, S. Shenker, “Comments on the Performance of Measurement-based Admission Control Algorithms”, Proc. of *IEEE INFOCOM 2000*, Mar 2000.

[88] C. Oottamakorn, and D. Bushmitch, “A Diffserv Measurement-based Admission Control Utilizing Effective Envelopes and Service Curves”, Proc. of *IEEE ICC 2001*, Jun 2001.

[89] K. Mase, and Y. Toyama, “End-to-end Measurement Based Admission Control for VoIP Networks”, Proc. of *IEEE ICC 2002*, May 2002.

[90] S. Chandramathi, and S. Shanmugavel, “Adaptive Allocation of Resources with Multiple QoS Heterogeneous Sources in ATM Networks – A Fuzzy Approach”, Proc. of *IEEE ICC 2002*, May 2002.

[91] P. Siripongwutikorn, S. Banerjee, and D. Tipper, “Adaptive Bandwidth Control for Efficient Aggregate QoS Provisioning”, Proc. of *IEEE GLOBECOM 2002*, Nov 2002.

[92] L.-D. Chou, and J.-L.C. Wu, “Bandwidth Allocation in ATM Networks using Genetic Algorithms and Neural Networks”, Proc. of *IEEE GLOBECOM ’97*, Nov 1997.

[93] H. Wang, C. Shen, and K.G. Shin, “Adaptive-Weighted Packet Scheduling for Premium Service”, Proc. of *IEEE ICC 2001*, Jun 2001.

[94] S. Floyd, and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance”, *IEEE Trans. on Networking*, 1(4), 1993, pp. 397-413, 1993.

[95] R.R.-F. Liao, and A.T. Campbell, “Dynamic Core Provisioning for Quantitative Differentiated Service”, Proc. of *IEEE IWQoS 2001*, Jun 2001.

[96] Z. Sahinoglu, and S. Tekinay, “A Novel Bandwidth Allocation: Wavelet-Decomposed Signal Energy Approach”, Proc. of *IEEE GLOBECOM 2001*, Nov 2001.

[97] J. Ilow, “Forecasting Network Traffic using FARIMA Models with Heavy Tailed Innovations”, Proc. of *IEEE ICASSP 2000*, Jun 2000.

[98] J.R. Gallardo, D. Makrakis, and M. Angulo, “Dynamic Resource Management Considering the Real Behavior of Aggregate Traffic”, *IEEE Trans. on Multimedia*, 3(2), pp. 177-185, Jun 2001.

[99] N. Semret, R.R.-F. Liao, A.T. Campbell, and A.A. Lazar, “Pricing, Provisioning and Peering: Dynamic Markets for Differentiated Internet Services and Implications for Network Interconnections”, *IEEE JSAC*, 18(12), pp. 2499-2513, Dec 2000.

[100] K. Malinowski, "Optimization Network Flow Control and Price Coordination with Feedback: Proposal of a New Distributed Algorithm", *Computer Communications*, 25(2002), pp. 1028-1036, 2002.

[101] R. Garg, R.S. Randhawa, H. Saran, and M. Singh, "A SLA Framework for QoS Provisioning and Dynamic Allocation", Proc. of *IEEE IWQoS 2002*, May 2002.

[102] C. Chuah, L. Subramanian, R.H. Katz, and A.D. Joseph, "QoS Provisioning Using A Clearing House Architecture". Proc. of *IEEE IWQoS 2000*, Jun 2000.

[103] S. McCanne, and S. Floyd, *ns-2 – The Network Simulator*, available from http://www.isi.edu/nsnam/ns/.

[104] IETF IP Routing for Wireless/Mobile Hosts (Mobile IP) Working Group, http://www.ietf.org /html.charters/mobileip-charter.html.

[105] P. Reinbold, and O. Bonaventure, "A Comparison of IP Mobility Protocols", Proc. of *IEEE SCVT 2001*, Oct 2001.

[106] S. Das, et al, "Integrating QoS Support in TeleMIP's Mobility Architecture", Proc. of *IEEE ICPWC 2000*, Dec 2000.

[107] K. Mitchell, and K. Sohraby, "An Analysis of the Effects of Mobility on Bandwidth Allocation Strategies in Multi-Class Cellular Wireless Networks", Proc. of *IEEE INFOCOM 2001*, Apr 2001.

[108] A. Klemm, C. Lindemann, and M. Lohmann, "Traffic Modeling and Characterization for UMTS Networks", Proc. of *IEEE GLOBECOM 2001*, Nov 2001.

[109] S. Choi, and K.G. Shin, "A Comparative Study of Bandwidth Reservation and Admission Control Schemes in QoS-sensitive Cellular Networks", *ACM Wireless Networks*, 6(4), pp. 289-305, 2000.

[110] M. Naghshineh, and M. Schwartz, "Distributed Call Admission Control in Mobile/Wireless Networks", *IEEE JSAC*, 14(4), pp. 711-717, May 1996.

[111] C.C. Wu, and D.P. Bertsekas, "Admission Control for Wireless Networks", *IEEE Trans. on Vehicular Technology*, to appear.

[112] B.M. Epstein, and M. Schwartz, "Predictive QoS-based Admission Control for Multiclass Traffic in Cellular Wireless Networks", *IEEE JSAC*, 18(3), pp. 523-534, Mar 2000.

[113] A. Aljadhai, and T.F. Znati, “Predictive Mobility Support for QoS Provisioning in Mobile Wireless Environments”, *IEEE JSAC*, 19(10), pp. 1915-1930, Oct 2001.

[114] F. Yu, and V.C.M. Leung, “Mobility-based Predictive Call Admission Control and Bandwidth Reservation in Wireless Cellular Networks”, Proc. of *IEEE INFOCOM 2001*, Apr 2001.

[115] A.K. Talukdar, et al, “MRSVP: A Resource Reservation Protocol for an Integrated Services Network with Mobile Hosts”, *ACM/Kluwer Wireless Networks*, 7(1), pp. 5-19, 2001.

[116] A. Mahmoodian, and G. Haring, “Mobile RSVP with Dynamic Resource Sharing”, Proc. of *IEEE WCNC 2000*, Sep 2000.

[117] S. Yoon, J.-H. Lee, K.-S. Lee, and C.-H. Kang, “QoS Support in Mobile/Wireless IP Networks using Differentiated Services and Fast Handoff Method”, Proc. of *IEEE WCNC 2000*, Sep 2000.

[118] C. Oliveira, J.B. Kim, and T. Suda, “An Adaptive Bandwidth Reservation Scheme for High-speed Multimedia Wireless Networks”, *IEEE JSAC*, 16(6), pp. 858-874, Aug 1998.

[119] J. Misic, Y.B. Tam, “Adaptive admission control in wireless multimedia networks under non-uniform traffic conditions”, *IEEE JSAC*, 18(11), pp. 2429-2442, Nov 2000.

[120] B. Li, L. Yin, K.Y.M. Wong, and S. Wu, "An Efficient and Adaptive Bandwidth Allocation Scheme for Mobile Wireless Networks Using An On-line Local Parameter Estimations", *ACM/Kluwer Wireless Networks,* 7(2), pp. 127-138, 2001.

[121] S. Kim, and P.K. Varshney, “An Adaptive Bandwidth Reservation Algorithm for QoS Sensitive Multimedia Cellular Networks”, Proc. of *IEEE VTC 2002-Fall*, Sep 2002.

[122] T. Zhang, et al, “Local predictive resource reservation for handoff in multimedia wireless IP networks”, *IEEE JSAC*, 19(10), pp. 1931-1941, Oct 2001.

[123] S.K. Das, R. Jayaram, N.K. Kakani, and S.K. Sen, “A Call Admission and Control Scheme for Quality-of-Service (QoS) Provisioning in Next Generation Wireless Networks”, *ACM/Kluwer Wireless Networks,* 6(1), pp. 17-30, 2000.

[124] C.-T. Chou, and K.G. Shin, "Analysis of combined adaptive bandwidth allocation and admission control in wireless networks", Proc. of *IEEE INFOCOM 2002*, Jun 2002.

[125] J.H. Lee, et al, "An Adaptive Resource Allocation Mechanism Including Fast and Reliable Handoff in IP-Based 3Gwireless Networks", *IEEE Personal Communications Mag*azine, 7(6), pp. 42-47, Dec 2000.

[126] J. Cai, L.F. Chang, K. Chawla, and X. Qiu, "Providing Differentiated Services in EGPRS through Packet Scheduling", Proc. of *IEEE GLOBECOM 2000*, Dec 2000.

[127] Y. Guo, and H. Chaskar, "A Framework for Quality of Service Differentiation on 3G CDMA Air Interface", Proc. of *IEEE WCNC 2000*, Sep 2000.

[128] C.-C. Lo, and M.-H. Lin, "QoS Provisioning in Handoff Algorithms for Wireless LAN", Proc. of *IEEE Seminar on Accessing, Transmission, Networking 1998*, Feb 1998.

[129] A. Veres, A.T. Campbell, M. Barry, and L.-H. Sun, "Supporting Service Differentiation in Wireless Packet Networks using Distributed Control", *IEEE JSAC*, 19(10), pp. 2081-2093, Oct 2001.

[130] D. Qiao, and K.G. Shin, "Achieving Efficient Channel Utilization and Weighted Fairness for Data Communications in IEEE 802.11 WLAN under the DCF", Proc. of *IEEE IWQoS 2002*, May 2002.

[131] I. Mahadevan, and K.M. Sivalingam, "Architecture and Experimental Framework for Supporting QoS in Wireless Networks Using Differentiated Services", *ACM/Baltzer Mobile Networks and Applications Journal*, 6(4), pp. 385-395, 2001.

[132] P. Ramanathan, K.M. Sivalingam, P. Agrawal, and S. Kishore, "Dynamic resource allocation schemes during handoff for mobile multimedia wireless networks", *IEEE JSAC*, 17(7), pp. 1270-1283, Jul 1999.

[133] K. Venken, D. De Vleeschauwer, and J. De Vriendt, "Designing a Diffserv-capable IP-backbone for the UTRAN", Proc. of *Second International Conference on 3G Mobile Communication Technologies 2001*, Mar 2001.

[134] S. Nananukul, and S. Kekki, "Simulation Studies of Bandwidth Management for the ATM/AAL2 Transport in the UTRAN", Proc. of *IEEE VTC 2002-Fall*, Sep 2002.

[135] A. Abella, V. Friderikos, and H. Aghvami, "Differentiated Services versus Over-provisioned Best-effort for Pure-IP Mobile Networks", Proc. of *IEEE MWCN 2002*, Sep 2002.

[136] MPLS Forum, http://www.mplsforum.org/.

[137] Y. Guo, Z. Antoniou, and S. Dixit, "IP Transport in 3G Radio Access Networks: an MPLS-based Approach", Proc. of *IEEE WCNC 2002*, Mar 2002.

[138] V. Vassiliou, et al, "A Radio Access Network for Next Generation Wireless Networks based on Multi-Protocol Label Switching and Hierarchical Mobile IP", Proc. of *IEEE VTC 2002-Fall*, Sep 2002.

[139] T. Robles, et al, "QoS Support for an all IP System beyond 3G", *IEEE Communications Magazine*, 39(8), pp. 64-72, Aug 2001.

[140] L. Becchetti, et al, "Enhancing IP Service Provision over Heterogeneous Wireless Networks: A Path toward 4G", *IEEE Communications Magazine*, 39(8), pp. 74-81, Aug 2001.