INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

WiMAX QoS

Service Flow management

Nuno Miguel Marques Rodrigues

Dissertação para obtenção do grau de Mestre em:

Engenharia de Redes de Comunicações

Júri

Presidente:	Professor Luís Rodrigues
Orientador 1:	Professor Rui Manuel Rodrigues Rocha
Orientador 2:	Professor Fernando Mira da Silva
Vogal 1:	Professor Mário Serafim Nunes

Setembro de 2008

Sumário

Proporcionar Qualidade de Serviço (QoS) a clientes, desde a rede de acesso até ao core, é possível com redes baseadas em 802.16. Com este fim, é proposto o uso do *Session Initiation Protocol* (SIP) para sinalização dos requisitos dos clientes. Estes requisitos são processados por um *Proxy* SIP modificado e enviados para um *Broker* localizado na rede de acesso. Este Broker provisiona dinamicamente as ligações pedidas pelos clientes e, se necessário, comunica com um Broker localizado no core da rede, por forma a que seja conseguido o mapeamento entre os paradigmas de QoS de ambas as redes. Este mapeamento permite que as características de QoS sejam mantidas desde o acesso até ao core da rede.

Os testes de desempenho do protótipo mostram que a configuração dinâmica das ligações na rede de acesso pode ser conseguida numa janela temporal inferior a 100 ms. Esta janela temporal é um preço razoável a pagar pela configuração dinâmica de ligações com QoS para a rede de acesso.

Palavras-Chave

WiMAX, Qualidade de Serviço, Gestão Dinâmica

Abstract

Providing Quality of Service (QoS) to clients from the access to the core network is possible with an 802.16 based access network. For this purpose, it is proposed the use of the Session Initiation Protocol (SIP) for signaling of client's QoS requirements. These requirements are processed by a modified SIP Proxy and sent to a Broker in the access network. This broker dynamically provisions connection requests from clients and, if necessary, communicates with the core Broker, so that a mapping between QoS paradigms of both networks is accomplished. This mapping allows that the QoS characteristics are maintained from the access to the core network.

The prototype's performance tests show that dynamically provisioning connections in the access network can be accomplished in a time-window smaller than 100 ms. This time window is a reasonable price to pay for dynamic configuration of QoS-enabled connections for the access network.

Keywords

WiMAX, Quality of Service, Dynamic Management

Contents

List of Figures

List of Tables

List of Appendices

List of Acronyms

16-QAM 16-state Quadrature Amplitude Modulation

64-QAM 64-state Quadrature Amplitude Modulation

AAA Authorization, Authentication and Accounting

AC Admission Control

AF Application Function

AP Access Point

AREQUIPA Application Requested IP over ATM

AS Autonomous System

ASN Access Service Network

ASP Application Service Provider

ATM Asynchronous Transfer Mode

B-ISDN Broadband Integrated Service Digital Networks

BE Best Effort

BPSK Binary Phase Shift Keying

BS Base Station

BWA Broadband Wireless Access

CAC Call Admission Control

CBR Constant Bit rate

CID Connection Identifier

CPE Customer Premisses Equipment

CPS Common Part Sub-layer

CS Convergence Sub-layer

CSN Connectivity Service Network

DiffServ Differentiated Services

DSA Dynamic Service Addition

DSC Dynamic Service Change

DSCP Differentiated Services Code Point

DSD Dynamic Service Deletion

ertPS Extended real time Polling Service

ETSI European Telecommunications Standards Institute

FDD Frequency Division Duplex

FTP File Transfer Protocol

H-NSP Home Network Service Provider

HTML Hypertext Markup Language

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IntServ Integrated Services

IP Internet Protocol

IPv6 Internet Protocol version 6

L2 Layer-2

L3 Layer-3

LOS Line of sight

MAC Media Access Control

MAN Metropolitan Area Network

MOS Mean Opinion Score

MPLS Multiprotocol Label Switching

MPEG Moving Picture Experts Group

MS Mobile Station

NAP Network Access Provider

NEs Network Elements

NHRP Next Hop Resolution Protocol

NLOS Non Line of Sight

NMS Network Management System

NRM Network Reference Model

nrtPS Non-real-time Polling Service

NSIS Next Steps in Signaling

NSLP NSIS Signaling Layer Protocol

NSP Network Service Provider

PDU Protocol Data Unit

PESQ Perceptual Evaluation of Speech Quality

PEVQ Perceptual Evaluation of Video Quality

PHS Payload Header Suppression

PHY Physical Layer

PMP Point to Multipoint

PPP Point-to-Point Protocol

QNE QoS NSIS entitie

QNI QoS NSIS Initiator

QoS Quality of Service

QPSK Quadrature Phase Shift Keying

QSPEC QoS specification

RRM Radio Resource Management

RSVP Resource Reservation Protocol

rtPS Real-time Polling Service

SDU Service Data Unit

SF Service Flow

SFID Service Flow Identifier

SIP Session Initiation Protocol

SLA Service Level Agreement

SNR Signal-to-noise ratio

SS Subscriber Station

TDD Time Division Duplex

TLM Thread Launching mechanism

TOS Type of Service

UGS Unsolicited Grant Service

USB Universal Serial Bus

V-NSP Visited Network Service Provider

VBR Variable Bit Rate

VoIP Voice over IP

WEIRD WiMAX Extension to Isolated Research Data Networks

WiMAX Worldwide Interoperability for Microwave Access

1 Introduction

The IEEE 802.16/WiMAX standard is one of the most promising technologies in terms of Broadband Wireless Access technologies. The definition of the standard itself comprises access to the network with fixed [14] and mobile terminals [18]. As a Metropolitan Area Network (MAN) it is most suited to areas where the installation of copper or cable-based technologies are not economically viable.

One of the most interesting features in te IEEE 802.16 standard is that it was designed with QoS support embedded, providing tools for the application of QoS in the access network. This QoS support follows a connection-oriented approach, which defines, for each connection, the scheduling service and QoS parameter set, like bandwidth, jitter or latency.

Underlying the QoS concept are issues like connection admission control, packet scheduling algorithms, connection management and network inter-connection. These issues were not included in the 802.16 standard and are current topics of research.

Solutions for WiMAX-specific problems (like admission control and packet scheduling) are typically combined and tend to take advantage of features of the 802.16 standard ([19] and [20]). These are normally not concerned with network inter-connection issues.

When concerns are extended to the integration of WiMAX technology with other QoS architectures (DiffServ or IntServ), arise interesting solutions ([21] and [23]) that combine the different paradigms typically with a cross-layered approach.

These solutions are scientifically relevant, as they have a strong theoretical component, with promising simulation results and solid architectural definition. Still, they lack a more practical implementation, which rises different types of problems.

There has been, however, work performed in terms of practical implementations, namely in the Heterogeneous Networks field. WEIRD [24], EuQoS [26] and MESCAL [27] are research projects, which aim at the definition and implementation of an end to end system for integrated QoS support.

EuQoS and MESCAL offer a more generic solution in terms of Heterogeneous Networks, while WEIRD focuses on 802.16-compliant networks. These solutions offer a complete integrated End to End QoS architecture, including testbeds for evaluation of these architectures.

This dissertation aims at using WiMAX-specific optimization solutions (namely at the admission control level) and additionally integrating some concepts that were introduced by the architectural references of Heterogeneous Networks projects, to create a prototype that is capable of dealing with dynamic management of connections in the access network and additionally preserve the QoS characteristics of these connections when they cross to a core network.

The prototype includes the implementation of a modified SIP Proxy, QoS Brokers and a mapping definition so that traffic is able to cross networks. Concerning the Proxy, it is able to process requests from clients and send client requirements to the QoS Broker of the access network. When the Broker

receives these requests, these will be subject to an admission control function. This function will allow the new connection if there are enough resources available in the air interface. If this is not the case, it will try to degrade low-priority connections, so that new connections are allowed in the network.

For cases that require that traffic crosses to the core network, it will contact a Core Broker, so that a mapping that is able to preserve connection's QoS characteristics is provided to this traffic.

1.1 Motivation and goals

The native QoS support in the 802.16 standard brings value-added in the access network, as it allows a distinction between Best Effort and non-Best Effort traffic. This feature introduces a challenge in terms of connection management in the WiMAX domain. However, one should not forget the heterogeneous characteristics of today's networks, which poses also a challenge in terms of QoS paradigm mapping.

As typical WiMAX specific solutions tend to be rich in terms of ideas and theoretical results, but poor in practical terms and inter-connection concerns, it is necessary to put into practice some of these ideas and evaluate their behavior. The evaluation should address resource usage optimization combined with the definition of an admission control function targeted specifically at the 802.16 standard.

However, one should not forget the main guidelines of reference projects in the area of Heterogeneous Networks, namely those that use WiMAX as an access network. In this context, WEIRD is a reference project, which is concerned with the definition of an end to end QoS architecture. Despite this fact, the prototype should not be as complex nor as wide in terms of features.

Providing differentiated QoS to end-users using, for example, Web sensitive content, is an unusual but rather interesting point. This allows different contents (like audio, video, text), with different QoS requirements, to be differentiated. This problem is solved with Application level information and can be considered a value-added feature.

This points in the direction of the conception and engineering of a prototype able to handle QoS requests, optimizing resource usage in the WiMAX domain and finally integration of application layer information, so that traffic is differentiated and served according to its QoS requirements.

1.2 Organization

The remainder of this dissertation is organized as follows. The second and third chapters introduce the state of the art in terms of the 802.16 technology and optimization strategies, plus the state of art in terms of Heterogeneous networks. In this chapter are approached solutions are generic to the problem of heterogeneous networks, but the focus will be on those that use WiMAX as the access technology. Additionally, an overview is given on the concept of QoS based on application data. In each of the subsections is a brief discussion about the analyzed concepts.

Chapter 4 contains the architectural view of the system, while the fifth chapter focuses on implementation issues. The sixth chapter presents the evaluation of the system in terms of functionality and in terms of performance. The final section will draw some conclusions.

2 Technology overview and concepts

In this chapter, the 802.16 technology (also known as WiMAX) will be introduced. The standard evolution will be presented and a quick overview of some concepts of the 802.16 architecture are presented. This section also introduces some QoS concepts (in IP networks and also in the 802.16 standard).

First, an overview of the 802.16 standard evolution is given, pointing out how the development started and how it evolved till the present date.

Next, some protocol layering fundamentals are presented. The main functions of the 802.16-2004 standard will be depicted. A special emphasis will be given on the 802.16-2004 MAC layer and on its native QoS support.

After giving an overview of the protocol layering, the view of a major player in the 802.16 world is outlined. The WiMAX Forum is acting as a certification entity which is trying hard to achieve inter-operability between different vendors. Along with the network reference model, some basic concepts about the 802.16 standard are described.

The subject that follows is QoS. First, an overview of the QoS concept is given. In the following section, some background info about QoS architectures is given.

The last section is dedicated to QoS from the 802.16 Standard point of view

2.1 Standard evolution

Few years ago, industry and users were demanding a Broadband Wireless Access (BWA). With the success of 802.11 (known as WiFi), users and industry wanted more. They wanted something that had the throughput of xDSL technologies but with the mobile and wireless characteristics of 802.11. Therefore, some people from the Institute of Electrical and Electronics Engineers (IEEE) 802 formed the 802.16 group to define a new BWA standard for Metropolitan Area Network (MAN). In 2001, the first document was approved. This was the begining of the stadardization process for 802.16. In this document was defined the Point to Multipoint (PMP) architecture for the system in the 10-66 GHz band. This frequency band required Line of sight (LOS) between sender and receiver because of its high operating frequency.

The group continued to work to improve the standard. Their goal was to make a Non Line of Sight (NLOS) standard. This would improve the usability of the technology, because sender and receiver would not need to be in LOS of each other to transmit. This improvement required a lower frequency band along with the definition of a new air interface. This air interface would have to be more robust to deal with different electromagnetic phenomena like reflection, scattering or multi-path. This way, in 2003 the IEEE approved the 802.16a standard. This standard extended the 802.16 specification to radio systems in the 2-11 GHz frequency (both licensed and unlicensed bands), providing NLOS usage. With these additions, the 802.16 standard was now extended and enhanced to work in rural or city areas where

NLOS is required in most cases.

Short time after the 802.16a standard approval, a new player entered the game. This player is known as WiMAX Forum. Like the WiFi Alliance, it also aims at inter-operability between different vendors.

In 2004, a new 802.16 standard was approved. It was the 802.16d standard. This standard reviews and consolidates the IEEE standards 802.16-2001, 802.16a and 802.16c. Today, most people refer to it as the 802.16-2004 standard, but it also known as the 802.16REVd. In future references in this document, this standard will be addressed as 802.16-2004.

To provide support for nomadic users, a new standard was developed. This standard is the 802.16e-2005. It was approved in 2005, enhancing the 802.16-2004 standard in PHY and MAC layers and introducing the Mobile Station (MS) concept. The new specification allowed users traveling at vehicular speeds to send and receive data. The 802.16 family had finally been extended to a fixed/mobile hybrid environment.

In 2006, the first WiMAX Forum certified products appeared. Like the WiFi Alliance for the 802.11 standard, this group is also interested in certifying products for inter-operability. The certification process by the WiMAX Forum is an important step towards the inter-operability of equipment from different vendors.

2.2 802.16 layers

The scope of the 802.16 standard comprises the definition of Physical Layer (PHY) and Media Access Control (MAC) layers. These layers are identified in figure 1.

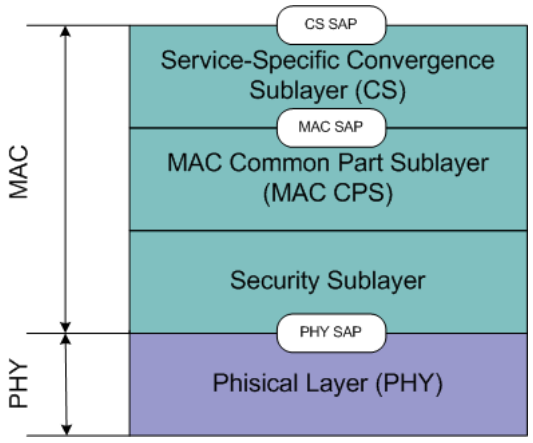

Figure 1: 802.16 layers

In figure 1 it is possible to see that the MAC layer comprises three sub-layers: the Convergence Sub-layer (CS), the Common Part Sub-layer (CPS) and the Security Sub-layer. The functions of the different sub-layers were summarized in table 1.

Layer	Sublayer	Functions
MAC	CS	Transform/map external network data into MAC Service Data Unit (SDU). Classify external network SDUs and associate them to the proper MAC Service Flow Identifier (SFID) and CID. Payload Header Suppression (PHS).
	CPS	System Access. Bandwidth allocation. Connection establishment. Connection maintenance. Applies QoS to the transmission and scheduling of data over the PHY.
	Security	Authentication. Secure key exchange. Encryption.
PHY		Support to different PHY specifications.

Table 1: Function of different layers in 802.16

The 802.16 standard defined two general CS: the packet and the ATM convergence sub-layers. The former is for use with packet based networks (IP, Point-to-Point Protocol (PPP) and 802.3 - Ethernet), while the latter is for use with cell-based networks (like Asynchronous Transfer Mode (ATM)). This sub-layer receives data from higher-layer protocols (through the CS Service Access Point (SAP) - communication point with packet or cell based network layers) and classifies it into the appropriate connection (it may be IEEE 802.1Q VLAN, IP address, etc), delivering it to the CPS.

The classification is how a MAC SDU is mapped onto a particular connection to be transmitted between MAC peers. This SDU is then associated with a connection, which in turn is associated with the service flow characteristics of the connection. This Service Flow contains the QoS constraints associated with the connection, which will guarantee the QoS requirements.

While the classification of the higher-layer protocols is made by the CS, the CPS is responsible for bandwidth allocation, connection maintenance and sending data over the PHY.

One of the most important features of the 802.16 technology is the Quality of Service (QoS) native support. Unlike other Wireless technologies (like 802.11), 802.16 was developed with QoS support. The QoS function is provided by the MAC layer. Because different applications have different requirements, four different scheduling services were defined in the 802.16-2004 standard: Unsolicited Grant Service

(UGS), Real-time Polling Service (rtPS), Non-real-time Polling Service (nrtPS) and BE.

The classes and corresponding applications are summarized in table 2. The different traffic classes are ordered by priority, being the first one, the one with most priority.

Class Name	**Traffic Type**	**Application (e.g.)**
UGS	CBR	T1/E1, VoIP without silence supression
rtPS	VBR	MPEG streams
nrtPS	non-real time	FTP
BE	non-real time	HTTP

Table 2: 802.16-2004 Traffic classes and relevant characteristics

The 802.16 standards family was also designed with security requirements from scratch. The CPS is the entity responsible for this.

Besides the MAC layer, the standard also defines some PHY specifications. Next, a brief overview of the PHY layer is given. The 802.16 standard defines both TDD and FDD configurations. In both cases, a per-SS burst profile, modulation and coding schemes may be adjusted on a frame-by-frame basis. The FDD case supports full-duplex SSs and half-duplex SSs. The latter are not capable of receiving and transmitting information simultaneously.

The standard allows the use of four different modulation techniques: Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), 16-state Quadrature Amplitude Modulation (16-QAM) and 64-state Quadrature Amplitude Modulation (64-QAM). The dynamic use of different modulation schemes allows robust or efficient data transmission. For example, a user that is far away from the BS may require the use of a more robust modulation scheme (BPSK), while a user near the BS will use a 64-QAM modulation to efficiently use his network connection.

The 802.16 further defined four different PHY. Table 3 summarizes the different types of PHY for the 802.16 standard. The WirelessMAN-SC operates in the 10-66GHz and is for LOS use, while the other PHY defined for the standard operate at a lower frequency (2-11GHz) and work on a NLOS environment. The WirelessMAC-SC was the first one to be defined in the 802.16 standard and it is primarily used for point to point communication. The others are more suited for a PMP topology.

2.3 Architecture and basic concepts

This section intends to describe some architectural details and to provide the reader some basic concepts on the 802.16 reference architecture and the QoS issue. In this field, the WiMAX Forum worked towards

PHY	Frequency (GHz)	How it works
WirelessMAN-SC	10-66	Based on a single-carrier modulation
WirelessMAN-OFDM	2-11	Uses Orthogonal Frequency Division Multiplexing
WirelessMAN-OFDMA	2-11	Uses Orthogonal Frequency Division Multiple Access
WirelessMAN-SCa	2-11	Uses single-carrier modulation

Table 3: 802.16-2004 PHY definition

a definition of an architecture with all components well defined [3]. The Network Reference Model (NRM) defined in these documents is presented in figure 2.

The NRM is a logical representation of the network architecture. It defines the functional entities and reference points. In figure 2, the functional entities are the Subscriber Station (SS)/MS, the Access Service Network (ASN) and the Connectivity Service Network (CSN), while the reference points are defined by R1-R5.

Let's take a closer look at each of these entities. The following descriptions do not intend to be exhaustive. The unambiguous, complete and formal definition of every single component is described in [3].

The left grey block is the Network Access Provider (NAP). This block represents a business entity that provides the radio access to subscribers. It is composed by one or more ASN and makes the connection between SS/MS and Network Service Provider (NSP).

There are two types of NSP: the home and the visited and the main functions associated with it are those related to services. They only exist because of the nomadic ability of subscribers. In a non-nomadic environment, only one of these entities exists and is called simply NSP. It is defined as being the Service Provider that establishes Service Level Agreement (SLA) with subscribers and is responsible for AAA functions. When it comes to mobility, the NSP is called Home Network Service Provider (H-NSP) and it should possess roaming relationships with other NSP - Visited Network Service Provider (V-NSP).

The SS/MS station is defined as the client device. In an 802.16-2004 environment, it will be an SS, while in the 802.16e-2005 it is called an MS, because users can be nomadic and roam between different NSP.

The ASN is defined as the set of functions needed to provide radio access to a subscriber. It should provide Layer-2 (L2) connectivity with the SS/MS, transfer Authorization, Authentication and Accounting (AAA) messages, network discovery, relay functionality for Layer-3 (L3) connectivity and Radio Resource Management (RRM). To support user mobility, some other functionalities were defined, like ASN anchored mobility, CSN anchored mobility, paging and ASN-CSN tunneling.

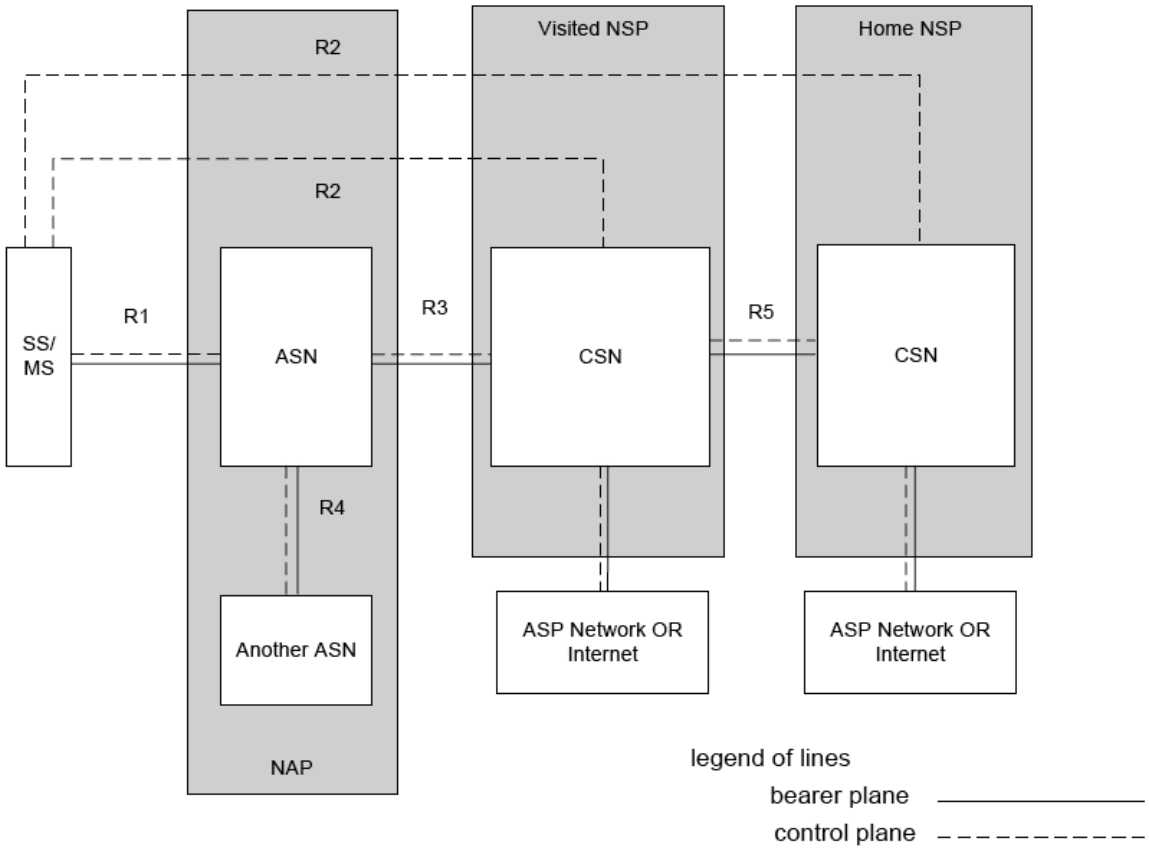

Figure 2: Network Reference Model

The CSN block is another component of the model. It is responsible for providing Internet Protocol (IP) connectivity services to subscribers. It may have functions like IP address allocation, admission control, Internet Access, AAA proxy or server, billing, inter-ASN mobility or services specific to the technology (like location based services). It may be composed by different Network Elements (NEs), like routers, gateways, databases or AAA servers.

Just a last note about the functional entities to say that they do not need to be separated physically, i.e., some of the functions may be aggregated physically (like a vendor solution) or they could all be separated, depending on the choice of who deploys it.

To assure inter-operability between equipment from different vendors, the WiMAX Forum decided to create specific reference points. These points can be seen in figure 2. They simply define the protocols that should be used to communicate between the different entities.

What was just defined is the network reference module. It divided the system into functional blocks. Let's take a different look at the system. Figure 3 presents its components in a different perspective.

Figure 3 is a typical 802.16 environment. Subscribers may have a fixed 802.16 SS, providing network access to switches and 802.11 devices. This should be the 802.16-2004 typical environment: SS are fixed.

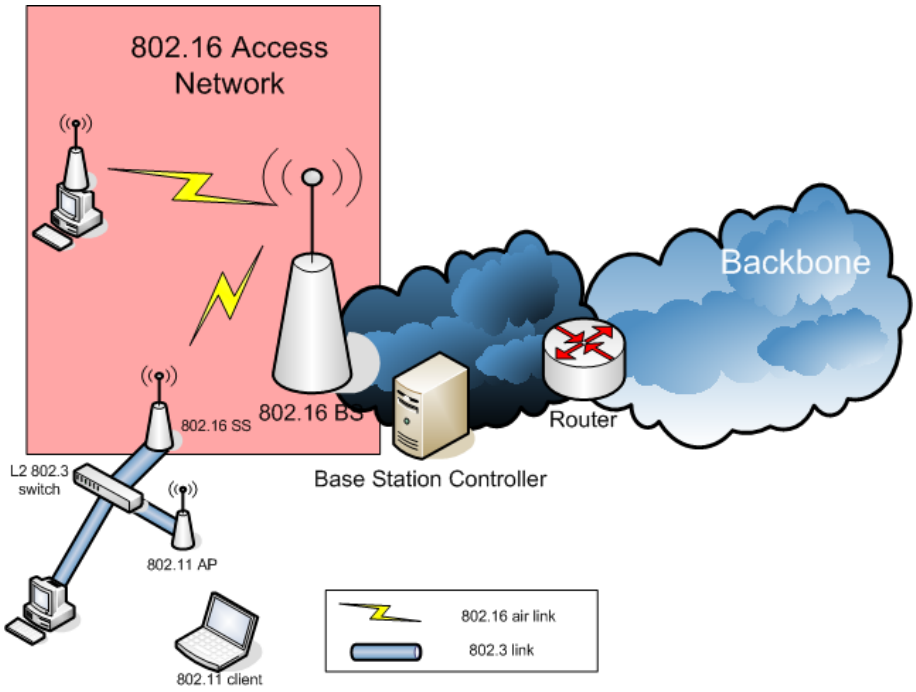

Figure 3: Typical 802.16 environment

In the 802.16e-2005, subscribers should be able to move between different Base Station (BS). Those client devices typically use a built-in device or a Universal Serial Bus (USB) card connecting to a computer.

It is also possible to see that 802.16 is an access technology that provides PMP access to users. In fact, the PMP topology is defined in the standard as mandatory to every 802.16 implementation. Still, the standard defines an optional topology: mesh. The difference between these two topologies is that in the PMP case, SS communicate only with the BS, while in the mesh mode, users communicate with both BS and SSs.

Another concept that is important in 802.16 is the mode of data transmission between BS and subscribers. This data transmission is made in a connection-oriented fashion. Furthermore, no data transmission is allowed before a connection is established between BS and SS. These connections are always defined unidirectionally, connecting SS and BS and are identified by an unique identifier: the Connection Identifier (CID). The direction of a connection is defined as being Downlink (from BS to SS) or Uplink (from SS to BS).

2.4 The Quality of Service challenge

QoS is a concept that can be analyzed from different perspectives. For instance, the ITU-T defined in [4] some recommendations related to QoS. It is defined as *service performance which determines the degree of satisfaction of a user of the service*. This definition can be considered a subjective evaluation of a

service.

As far as evaluation goes, there is a particular measurement indicator that evaluators try to reach: the Mean Opinion Score (MOS). There are several MOS methods. In these, users are invited to perform a measurement of quality. They assign an MOS rating to the subject of their evaluation which is typically scaled from 1 to 5. The rating means the following: 1 - bad, 2 - poor, 3 - fair, 4 - good, 5 - excellent. The MOS is the arithmetic mean of all the individual scores. This measure is typically adequate for multimedia services like video or audio.

The ITU-R defined recommendation [5] for the subjective assessment of television pictures. This recommendation gives information about important aspects of environment, monitor characteristics, source signals, sample selection, among others.

The telephony world has also been subject of recommendations. In this case, [6] provides directives for scenario, loudness, pleasantness of tone, among others.

There have also been defined some evaluation methods as the Perceptual Evaluation of Video Quality (PEVQ) or Perceptual Evaluation of Speech Quality (PESQ). The first is a measurement algorithm that provides MOS of video quality (wether it is IPTV, Video Streaming, video-conference or even mobile TV). The latter produces also a MOS result, but it focuses on speech quality. ITU-T defined [7] for PESQ. Both models are based on a human model to provide the opinion score.

These previously mentioned methods and recommendations are powerful tools to assess how multimedia services are performing in a network. However, these same networks that support these services may be also subject to evaluation. These evaluations are typically defined by network providers.

These providers define metrics like: downtime, throughput, jitter, packet loss or delay. These are typically the parameters negotiated with clients. For instance, an e-commerce company may have severe requirements in availability, while a VoIP provider has requirements in terms of packet loss, delay and jitter. Depending on the client's requirements, different parameters are subject to more or less sensitivity.

2.5 QoS in IP networks

IP is the current network level protocol used in the Internet. The network is composed by routers where clients are connected. It works in a very simple way: connectionless. Datagrams are sent by client devices to a destination. When routers receive these datagrams, they inspect the destination address in the datagram header and send them to their destination. If they have no direct route to the destination, they forward the datagram to a neighbor router who knows the path. If IP networks are under heavy load, routers will start to discard packets and great delays occur. This happens because IP networks work on a Best Effort (BE) scheme, i.e., routers will forward datagrams as fast as they can, without guaranteeing QoS.

In order to guarantee QoS in IP data networks, two different protocol architectures were proposed by

the Internet Engineering Task Force (IETF): Integrated Services (IntServ) and Differentiated Services (DiffServ). The former requires applications to signal their service requirements to the network through a reservation request. This means that an application that wishes to traverse the network will have to reserve resources along a path. In this case, reservation is made on a per-data-flow basis. For a network to support this kind of architecture, all routers in the path must support this reservation signaling to establish the path. Currently, the protocol used to make the end-to-end signaling is Resource Reservation Protocol (RSVP). One obvious disadvantage of this architecture is its ability to scale and the complexity to provide end-to-end paths. One advantage behind IntServ is its ability in giving on a per-flow basis assured QoS requirements. IntServ is documented in [8], [9] and [10].

In response to the complexity of the IntServ architecture, a different approach was developed: DiffServ. Sometimes in literature, this architecture is considered as an "in the middle" approach: it does not provide the fine-grained QoS specification of IntServ (which acts on a per-flow basis), nor the BE approach (where packets are forwarded on a per-packet basis).

Instead of having network resources reserved end to end, DiffServ divides traffic in classes. These classes are called Forwarding Classes. Each of these classes has a specific code, which goes into the IP packet header (Differentiated Services Code Point (DSCP) or Type of Service (TOS) fields). Further, each of these forwarding classes have a specified treatment in terms of drop priority and bandwidth allocation.

In a Differentiated Services network, nodes that are located in borders of domains (edge or boundary nodes) have different responsibilities of those in the core network. The former have the responsibility of classifying and traffic shaping, while the latter forwards packets based solely on forwarding classes in the packet header.

The differences of the IntServ and DiffServ architecture may be summarized as follows: resource allocation is applied to aggregated traffic rather than individual flows. DiffServ uses edge nodes to make the policing of traffic, leaving the core nodes the job of forwarding based on the traffic class. Also, DiffServ serves traffic based on pre-defined forwarding behaviors, instead of end-to-end services. The documentation regarding DiffServ may be found in [11] and [12]. Also, a good guide for IntServ, DiffServ, MPLS and Traffic Engineering is [13].

2.6 QoS in the 802.16 Standard

2.6.1 Theory of Operation

This subsection provides an overview of the QoS control mechanisms in the 802.16 standard that provide QoS in the access network.

The main mechanism that allows the QoS in an 802.16 network is the mapping of packets traversing the MAC interface into Service Flow (SF). These SFs are identified by a CID. A SF is defined as being

an unidirectional flow of packets that ensures certain levels of QoS. Both BS and SS provide this QoS according to the QoS Parameter Set defined for the particular SF the packets correspond to.

Service Flows exist in both directions: Uplink and Downlink and they may even exist without being activated. In this case, they will not transport any data. All SF have a unique 32-bit SFID. When in admitted or active state, SF also have a 16-bit CID.

The QoS features in the 802.16 standard define transmission ordering and scheduling on the air interface, but these features may need to be used jointly with other mechanisms to enable end-to-end QoS.

2.6.2 Object Model

The major blocks that compose the object model are presented in Figure 4. Each object has a unique identifier (underlined). The attributes that are in brackets are optional. The objects are connected with association lines. At the end of these lines are the number of elements that may be associated with each other.

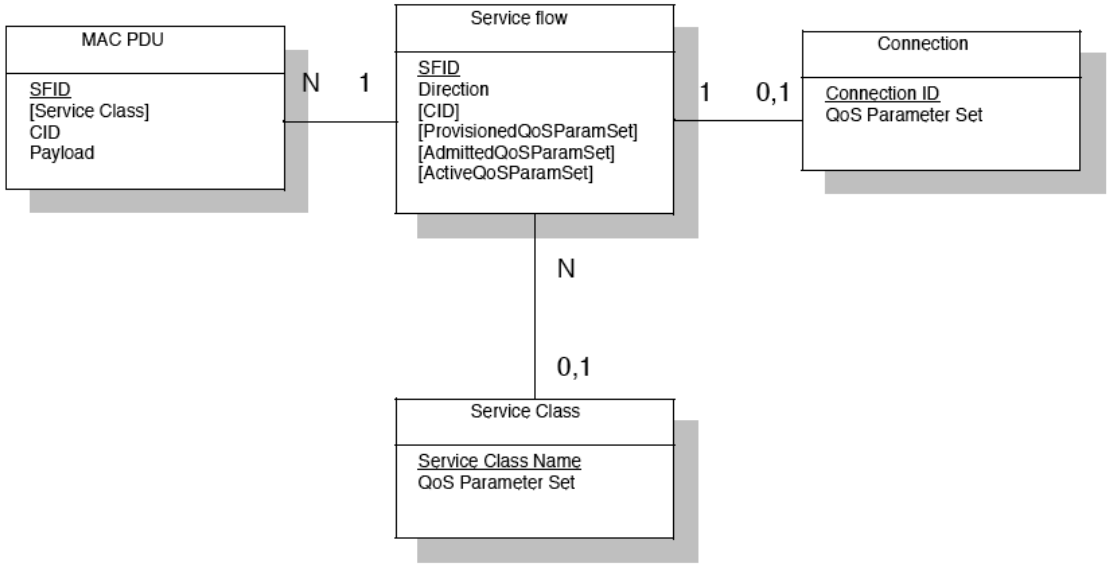

Figure 4: QoS object model (source: [14])

The central concept is the Service Flow (described in more detail in 2.6.3). The attributes make a description of the object. It has an unique identifier (SFID), a direction and optionally a CID, a Provisioned QoS Parameter Set, an Admitted QoS Parameter Set and an Active Parameter Set.

As one can see in Figure 4, each SF has associated 0 to N Protocol Data Unit (PDU). The 0 means that it is not carrying any data. When a Service Flow is operational, many PDUs may be associated with it.

Each Service Flow may have associated 0 or 1 connections. It depends on its state. If it is in a provisioned state, it will have no connection associated, but if it is in Admitted or Active state, it must have a Connection associated with it.

The Service Class object is an optional object that may be implemented in the BS side and is referenced by an ASCII identifier (name). It has associated a set of QoS Parameters.

2.6.3 Service Flows

As mentioned earlier, a SF may be defined as a MAC service that provides unidirectional transport of packets. It features a set of QoS parameters (e.g. latency, jitter, throughput).

In table 4 is a brief description of the components that define a Service Flow.

Attribute	**Description**
SFID	Unique identifier assigned to every SF.
CID	Connection Identifier. Only exists if the SF has an admitted or Active state.
ProvisionedQoSParameterSet	Set of parameters provisioned via, for example, a Network Management System (NMS).
AdmittedQosParameterSet	Set of parameters for which BS (and SS) are reserving resources.
ActiveQoSParameterSet	Set of parameters that define the characteristics of the service being delivered. Only Active SF may forward packets.
Authorization Module	A logical function that approves or denies changes to the QoS Parameters and classifiers associated with a SF.

Table 4: Service Flow Attributes

The QoS Parameter Sets presented in table 4 have relationships between them, i.e., the Active QoS Parameter set is a subset of the Admitted Qos Parameter set, which in turn is a subset of the Provisioned Qos Parameter Set.

For cases where a dynamic authorization model is used, another set of parameters is used: the Authorized QoS Parameter Set, which is a subset of the Provisioned QoS Parameter Set and a superset of the QoS Admitted Parameter Set.

Service Flows come in three different flavors (note that are more than these 3. These are considered the basic):

- Provisioned

- Admitted
- Active

A Service Flow may be provisioned but not immediately activated. It is sometimes called *deferred*, i.e., the flow may be provisioned but admission and activation are deferred. In this case, the Admitted and Provisioned QoS Parameter Sets and the CID are null.

The concept of the Admitted Service Flow is similar to the concept used in telephony applications. It follows a two-phase activation model: resources are first admitted and then, after the negotiation is complete, resources are activated.

A Service Flow that has the Active QoS Parameter Set is said to be active. It is requesting and being granted bandwidth for transport of data packets. When, for example, an admitted Service Flow provides the Active QoS Parameter Set, if accepted, the Service Flow changes its state to Active. This is the final stage of the two-phase activation model.

It is not necessarily needed that the service flow creation process passes through the activation model. A Service Flow may be provisioned and immediately activated, as long as authorization (and resources) is granted for this Service Flow.

2.6.4 Authorization

All changes to a Service Flow's QoS Parameters need to be submitted to an authorization module. This applies to the addition of new Service Flows and change of QoS Parameters (augment or reduction).

The model can be implemented in two different ways: static or a dynamic. When a static model is implemented, admission and activation are always permitted, as long as Parameter sets are valid. This way, a SS is provisioned when system operation initiates.

When a dynamic authorization model is used, the operation is a bit different. The BS must contact a policy server to decide if it should or should not allow change/addition of new Service Flow parameters. This policy server may even signal the authorization in advance.

2.6.5 Service Flow Management

Service Flows may be created, changed or deleted. These actions are accomplished through DSx MAC messages: Dynamic Service Addition (DSA), Dynamic Service Change (DSC) and Dynamic Service Deletion (DSD). As the names suggest, the DSA message is used to add a new Service Flow, the DSC message is used to change a Service Flow's characteristics and the DSD message serves the purpose of deleting it.

This yelds a state machine for Service Flows, which is represented in Figure 5.

For each Service flow exists a state machine like the one illustrated. The null state represents the non-existance of the Service Flow. Once a DSA message is received by the BS, the SF is in an operational

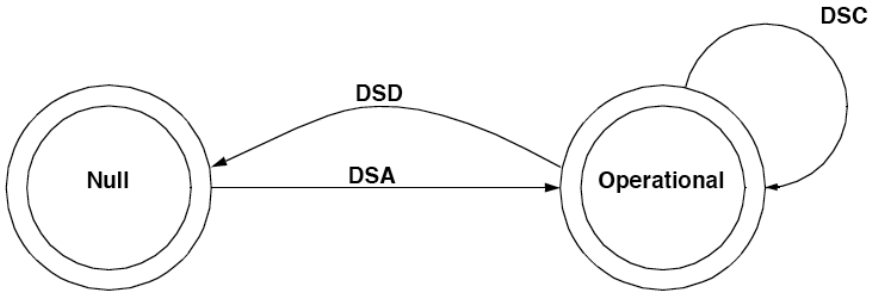

Figure 5: Service Flow state machine (source: [14])

state. It is at this point that an SFID is created and associated with the SF. Whenever DSC messages are received they affect only the Service Flow which corresponds to its SFID.

3 Related Work

One of the main characteristics of the 802.16 standard is that it has QoS support natively embedded. This provides network operators the ability to make the separation of traffic in the access network segment. This requires WiMAX specific solutions to provide control over resources given in the air interface. However, the problem of quality of service is not circumscribed to the access network.

The heterogeneity of today's networks poses a challenge in terms of assuring that the QoS specifications are assured throughout the network. This challenge is in terms of protocol layering adaptation and also in terms of mechanisms that assure that resources are reserved for a specific data flow. The former is concerned with issues in the network edge (e.g. edge router), while the latter is concerned with resource reservation over different domains.

In terms of resource reservation, the mechanism may comprise the use of a signaling protocol or Application layer information. The latter allows applications to establish QoS-enabled connections based on application information. This information describes the application's requirements (e.g. in terms of bandwidth or priority).

3.1 QoS based on application data

In the mid-90's, a research group started some research on the theme: Application Requested IP over ATM (AREQUIPA) [15]. Their main goal was to deliver Quality of Service to Web users. In the proposed solution, applications open a direct (ATM) connection to the client, with the application's requirements. The idea is to associate META-data with sensible information (data flows with QoS requirements). This addition enabled the creation of circuits that provide requirements like bandwidth, latency, etc to the sensible contents of the Web Server, on-demand.

Few years later, in [16] was proposed a similar approach, but with the variant of being deployed on an Internet Protocol version 6 (IPv6) DiffServ network. The implications of performing the same tasks on a DiffServ network imposed some differences from the work deployed by AREQUIPA.

In the AREQUIPA environment, the usage of the mechanism implied altering the client device (the requestor), while in the new approach, the modifications were made in the Web server. One of the problems of AREQUIPA was that it implied great modifications in the ATM stack. The simple modification introduced in this new approach, allowed (QoS-unaware) clients to benefit from privileged treatment. This was accomplished by the use of determined ports in the Web Server. Traffic coming from these ports was marked in the router with a Forwarding class that assured the QoS of the data flow. Naturally, requests that were addressed to the default BE port had to be redirected to the port that assured the QoS requirements of the data flow. As in [15], the META-data defined the QoS of each resource.

The previous approach presented a solution for enabling QoS in traffic flows. However, this approach only provided QoS in one way (Downstream) through the use of META-info. Still, it was appropriate,

because the testbed included a Web Server (traffic tends to circulate more in one way).

If a system needs to assure QoS in both ways (Upstream and Downstream), the problem gets more complex. One of the obvious approaches is signaling. Using Session Initiation Protocol (SIP), it is possible to signal the needs of a specific traffic flow. An interesting approach using SIP is the one described in [17].

The authors suggest the enhancement of a SIP proxy. This proxy (further called Q-SIP server) intercepts the SIP messages sent by clients and alter them. The messages sent by the Q-SIP server have extra-fields that indicate the requirements of the connection being initiated (QoS-Info headers). This way, another QoS aware SIP proxy is able to intercept the message and take measures for provisioning.

In order to make the provisioning of traffic flows, the Q-SIP server must communicate with a Bandwidth broker. This broker is responsible for the admission control and resource allocation. The interaction is accomplished with the COPS protocol.

This approach has the advantage of not requiring the clients to be modified. The weight of enhancement is on the Q-SIP servers. The authors also stress their architecture's support to two different QoS models: QoS assured and QoS enabled.

In the assured model, the call is only established if the requested/required QoS of that specific call is set along the path (from caller to called person). This means that the QoS setup is a precondition for the call. In the QoS enabled model, the QoS setup is not a precondition, i.e., the call is setup even if the QoS resources are not available throughout the path.

The above approach enabled the use of Up/Downstream resource reservation through different networks. This was accomplished introducing new fields in SIP messages. Like AREQUIPA, the use of descriptive information of traffic flows enabled the introduction of QoS. As one can see, the introduction of META-info revealed itself very useful for resource allocation.

3.2 WiMAX specific solutions

The scope of the IEEE 802.16 standards include the definition of the air interface, namely medium access control layer and multiple PHY specifications for fixed BWA [14]. It further defines enhancements to [14] to support nomadic subscriber stations (subscribers moving at vehicular speeds) [18]. It provides a specification for fixed and mobile BWA. In [18] it further defines the higher layer handover procedures between base stations or sectors.

What the IEEE 802.16 standards family does not define is the way packet scheduling and Call Admission Control (CAC) should be made. These subjects were left open to provide vendors a competitive environment in 802.16 solutions. Naturally, research efforts in this area are being made. Solutions available in literature typically tend to combine both aspects that were left open by the standard, developing QoS-aware architectures. These architectures cover the mentioned research topics plus some other interesting topics. In this section relevant references are summarized, to give an overview of these research

topics.

The following section will focus on some relevant resource allocation models for the WiMAX domain. These models are an effort to optimize overall system performance. Next, follows a section of QoS-aware architectures that are concerned with overall system enhancement, introducing some interesting ideas.

3.2.1 Resource Allocation

In [19], the authors propose a dynamic admission control model. The authors take advantage of the scheduling services characteristics.

The nrtPS scheduling service admits a bandwidth interval, i.e., it defines minimum and maximum bandwidth values. The bandwidth that is available to this scheduling service varies in this range, which is defined as: [b_{nrtPS}^{min}, b_{nrtPS}^{max}], being b the bandwidth allocated to the nrtPS flow.

The authors propose the use of a degradation model to borrow bandwidth from the nrtPS flows. It is important to mention that the only scheduling services that are allowed to borrow bandwidth are those with higher priority (UGS and rtPS). In this degradation model the authors consider the following definitions:

B - total bandwidth available

U - bandwidth used exclusively by UGS scheduling services

b_{UGS} - bandwidth required by a new UGS connection

b_{rtPS} - bandwidth required by a new rtPS connection

b_{nrtPS} - bandwidth required by a new nrtPS connection

l_{nrtPS}^{n} - degradation step

δ - amount of bandwidth degraded for each degradation step

When a new request for an UGS flow arrives at the BS, it is accepted if the bandwidth currently in use by all ongoing UGS connections plus b_{UGS} is less or equal to U. Otherwise, the new connection is rejected.

When a new rtPS connection arrives at the BS, it will be accepted if the bandwidth in use by all non-UGS connections plus b_{rtPS} is less or equal to B-U. When this is not the case, the model tries to borrow bandwidth from the nrtPS connections. In each step, it borrows $l_{nrtPS}^{n}\delta$. If the maximum degradation level is reached and there is still not enough bandwidth to give to the new connection, it will be refused.

When a new nrtPS connection arrives at the BS, if the bandwidth set aside for ongoing connections plus b_{nrtPS}^{max} - $l_{nrtPS}^{n}\delta$ is less or equal to B-U, this connection is accepted. If this is not the case, the degradation level will increase. If the maximum degradation level is reached and there is still not enough bandwidth available to accept the new connection, it will be refused.

For the case of new BE connections, they will always be accepted. This happens because no bandwidth is set aside for these new connections. The BE connections are only served after all the other have been

served.

In [20], the authors present a different, but still interesting approach. In this particular case, the authors do not take into account the characteristics of the scheduling services. The objective is to achieve the minimization of bandwidth provisioning, while keeping the MAC signaling to a minimum.

To characterize the system, there are a few parameters that are defined by the authors:

C_M - Maximum bandwidth that can be reserved

C_m - Minimum bandwidth that can be reserved

T - Fixed threshold with a value inferior to that of C_m

$C(t)$ - the instant bandwidth at a given t.

The bandwidth reservation varies according to the rules:

- When $C(t)$ is less than C_m, the bandwidth reserved will be C_m.
- When $C(t)$ is equal to C_m, the reserved bandwidth will move from C_m to C_M.
- When the current reserved bandwidth is C_M, but $C(t)$ reaches the threshold of T, the reserved bandwidth will decrease from C_M to C_m.

The authors use pre-determined steps of reserved resources. The model alternates between these steps as the network reaches low or high peaks of load. This way, when many clients are using the network simultaneously, the higher step provides many reservations, but when there are few clients using the network, the number of reserved resources decreases significantly to the lower step.

This method certainly decreases the amount of signaling messages exchanged, because the amount of resources reserved does not need to change every time there is a new client entering or leaving the network.

Besides this two step model, the authors propose a generalization model that provides n-steps. This incurs on an increase of MAC signaling, but the amount of resources that are over-provisioned at a given t, decreases.

Discussion:

The two models proposed ([19] and [20]) represent a sample of how researchers are working in order to control (dynamically) resource allocation.

In [19], a degradation model was presented. The most interesting part of the proposal is the degradation of the nrtPS connections, to allow the entrance of new connections. This allows the reduction of over-provisioning of resources, reducing the nrtPS allocated resources to a minimum in case of network congestion.

In [20] the authors make an *a-priori* reservation of resources. When the network is under heavy load, resources provisioned increase, but when the network is under low utilization, resources are freed.

Both models work well separately but, if they were combined, this could prove to be a good improvement in terms of network usage. On one hand, the system would benefit from an *a-priori* network reservation, as it would allow both the reduction of MAC signaling and waiting time for a new connection also. On the other hand, the degradation model could degrade the nrtPS connections to allow new connections to enter the network.

Still, these two models alone, tend to focus on the problem of resource reservation. The following section will give an overview on some QoS architectures for the WiMAX domain that could be combined with these models to improve the overall system performance.

3.2.2 QoS architectures

In the WiMAX domain, some architectures were suggested to improve system performance. Examples of these architectures are [21], [22] and [23]. The common point between these three architectures is that they all are cross-layered.

In [21], the authors propose an architecture to provide multi-layer integrated QoS control for the 802.16 standard. The IP QoS architectures supported are IntServ and DiffServ. Figure 6 illustrates the proposed architecture.

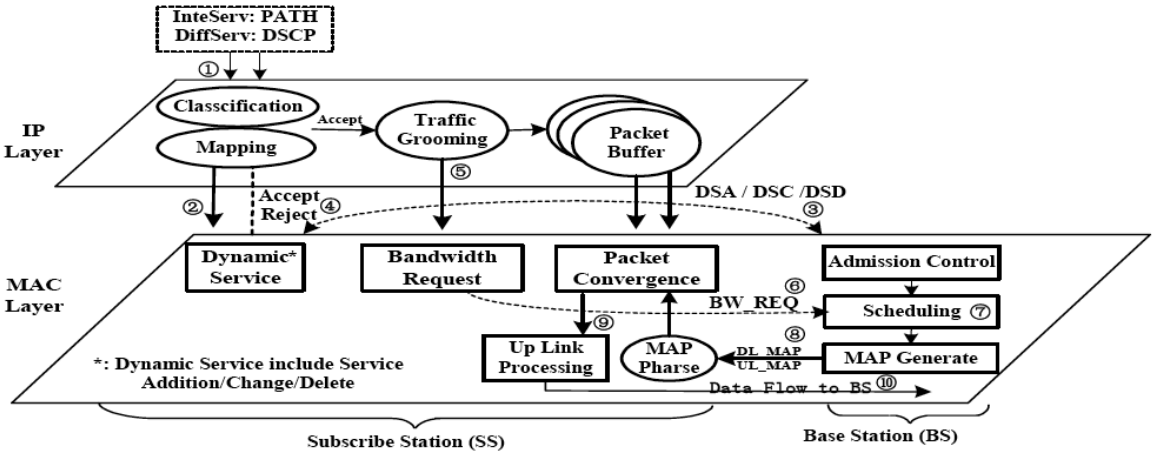

Figure 6: Multi-layer integrated QoS control architecture (source: [21])

The model is clearly cross-layered (IP and MAC Layer) and applies to SS and BS. The numbers represent the sequence of events that occur. In step 1, a client is trying to provision a new Service Flow. The client may be trying to establish this connection marking the DSCP field of IP packets (DiffServ) or sending a PATH message (IntServ). Table 5 illustrates the mapping rules for IntServ Services. It is possible to observe that the different traffic classes are described in terms of bandwidth requirements and delay, jitter and loss rate. Depending on their characteristics, they are classified into the MAC Layer Services of the 802.16-2004 standard.

Traffic Class	**Bandwidth Requirements**	**Delay / Jitter / Loss Rate**	**MAC Layer Services**
Hard QoS guarantee (e.g. VPN tunnel, E1/T1)	Constant Bandwidth	Minimum packet delay, jitter and loss rate	Unsolicited Grant Service
Soft QoS guarantee (e.g. VoIP, VoD, FTP...)	Guaranteed	Regular delay, jitter require	Real-Time Polling Services
	Not guaranteed	long delay, jitter required	Non-real-time Polling Service
Best effort (e.g. HTTP)	Only Basic connection	N/A	Best Effort

Table 5: Mapping rules for IntServ Services

In table 6, the same traffic classes are mapped to MAC Layer Services, but this time with the reference to the DS Octet. Those bits in the DS Octet column represent bits from 5-3 of the DSCP field. The 101 bits correspond to the Expedited Forwarding (EF) Class, while the 100 / 011 / 010 / 001 correspond to Assured Forwarding (AF) Classes. The three most priority ones are assigned to the rtPS MAC Layer Service, while the one with less priority is associated with the nrtPS service.

Traffic Class	**Service Description**	**DS Octet (DS5-3)**	**MAC Layer Services**
Hard QoS guarantee (e.g. VPN tunnel, E1/T1)	Critical	101	Unsolicited Grant Service
Soft QoS guarantee (e.g. VoIP, VoD, FTP...)	Flash Override, Flash, Immediate	100 / 011 / 010	Real-Time Polling Services
	Priority	001	Non-real-time Polling Service
Best effort (e.g. HTTP)	Runtime	000	Best Effort

Table 6: Mapping rules for DiffServ Services

In step 2, the connection is mapped from IP to MAC layer using the mentioned rules and an exchange of DSx messages is triggered. A DSA message will be sent to the Admission control module in the BS (step 3).

The Admission Control model used is based on very simple principles. The connection is accepted

if there is enough bandwidth available in the BS. In the case of admission of the flow, the Scheduling module is notified and the SS is notified (step 4).

When the SS is notified of the acceptance, the model moves to the traffic grooming module. According to the result of this module, the SS sends a BW_REQ message (step 6) to the BS. The scheduling module in the BS receives this request (step 7) and generates new UL and DL-MAP messages (step 8).

Finally, the SS may start the transmission of data, packing the IP packets into PDUs and then uploading them to the BS, according to the UL-MAP (steps 9 and 10).

In [22] there were made a few enhancements to the model proposed in [21]. Basically, the mapping rules presented in tables 5 and 6 are the same, as the admission control mechanism.

The improvements come from a fragment control mechanism and a remapping scheme. The authors claim that fragmentation is a required function of the BS as the size of each slot (Maximum Transmission Unit - MTU) in the 802.16 standard is much smaller than the size of IP packets.

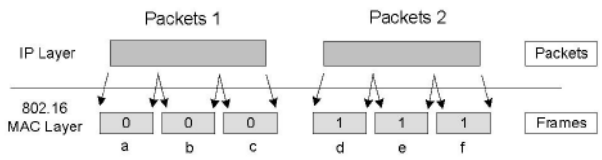

Figure 7: Fragment control (source: [22])

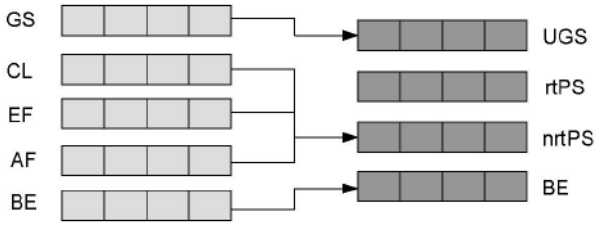

Figure 8: Remapping (source: [22])

The objective of the Frament Control is to group IP packets as they are treated in MAC level (they assume that fragments are not inter-leaved in the buffer). This is accomplished through the marking of a bit in the 802.16 MAC frame (Reserved bit). Packets are marked with '0' or '1' alternately as figure 7 shows. This will allow that, in case of congestion, fragments of the same IP packet are removed together, saving unnecessary frame transmissions.

The proposed remapping scheme aims at better buffer utilization and reduce frame dropping. Because the rules that apply to the mapping of packets are those of tables 5 and 6 and this mapping is static, there are cases where the rtPS queues may go into an overflow state (due to the bursty characteristics of this traffic), while the nrtPS queue still has space to accept more frames.

To overcome this problem, the authors propose a Queue monitor plus two threshold parameters. The Queue Monitor polices the queues, while the thresholds (Upper and Lower-Bound) control the remapping of frames.

The remapping mechanism works as follows. When buffer utilization of the rtPS queue reaches its Upper threshold, the remapping scheme is triggered and frames stop being allocated to their original queue (rtPS) and start being allocated to the nrtPS queue. Figure 8 shows the packets in Controlled Load (CL), EF and AF buffers being mapped to the nrtPS queue, leaving the rtPS buffer without

incoming frames.

When the remapping is occurring and the Lower-Bound threshold of the rtPS queue is reached, then the frames being allocated to the nrtPS queue stop and the original mapping rules are applied.

In [23] a new QoS aware architecture is proposed. This architecture introduces new interesting features. As the previously mentioned architectures, it is cross-layered.

The first difference between this architecture and those proposed by [21] and [22] is the mapping. In [23], the incoming data packets are classified based on the IEEE 802.1p recommendation. The authors introduce the Data Classifier module, which is responsible for the identification of packets according to this recommendation.

Another module that was introduced by the authors is the Database Module. It keeps records of all active connections, including QoS parameters associated to those connections (delay, jitter, bandwidth). Whenever a change is made, this module is notified of the changes.

Finally, a new module is introduced, which wasn't introduced so far: the Channel Adapter and SNR sniffer. This module is responsible for the evaluation of propagation conditions. It allows the inspection of the wireless medium, providing information related to SNR, fading, etc. This is useful to adapt the profile of the uplink/downlink connection in accordance to the propagation conditions.

Discussion:

This section gave an overview of some research projects in the WiMAX domain. The specific solutions presented considered mainly an integrated cross-layered architecture. In [21], a mapping rule between IP and MAC layer was suggested. The mapping rules presented are consistent with the characteristics of the flows, but there is one problem that arises with the QoS architectures mapping: it is assumed that the client device is doing the initial marking of IP packets (for the DiffServ case). This raises the following problem:

Imagine a client that is aware of this architecture. It may signal to the network that all packets that he is sending have the highest priority. This will guarantee him that his packets always get a *Premium* treatment (at least in the WiMAX segment).

Another issue that may be of concern is the lack of definition of drop precedence in table 6. Although the definition is correct to separate the DiffServ classes: Expedited Forwarding (EF), the four Assured Forwarding (AF) classes and the Best Effort class, some consideration should have been made in the drop precedence of the AF classes. This would allow more options in terms of drop priority in case of congestion. The definition of mapping rules for these cases would allow the less priority flows to be dropped with a higher probability. This would benefit the connections which have Hard QoS requirements and are sensitive to packet drop and delay.

Another problem that arises with this architecture is the way of reserving resources. In [21], it is the responsibility of the SS to make requests to the BS to allocate new Service Flows. This may be a problem, as in the IEEE 802.16-2004 standard, this is not a mandatory function, i.e., different vendors

may opt to not implement this functionality. This would leave this architecture inoperable.

Despite the problems mentioned, this particular architecture turns out to be very efficient when it comes to handling the IntServ QoS architecture. The number of messages exchanged is reduced because when a PATH message is sent by the client device, the Dynamic Service module is able to map this request immediately into a DSA message to the BS (refer to figure 6). This leads to less signaling when an IntServ path is established.

In [22], some improvements were made to guarantee QoS, but at a different level. The authors still used the mapping rules that were proposed by [21], but decided to make some additional efforts, namely at frame level. They introduced the fragment control, that allows that IP packets are grouped at L2. This way, in case of congestion, frames that belong to the same IP packet will be dropped all at a time. This allows that unnecessary frame transmissions are avoided.

The other important functionality introduced at frame level was the remapping scheme. This allows the rtPS buffer to not overflow in case of congestion, distributing the frames to the nrtPS buffer. This will avoid packet drops when the nrtPS buffers are not completely full. This feature could be extended to control the delay of frames. This way, when the rtPS buffer queue starts to grow, the same threshold mechanism could control the delay, being applied to delay-sensitive applications.

The new architecture proposed by [23] is still a cross-layered approach to the problem, but it involves one more layer than the other two: physical.

The introduction of the Channel adapter and SNR sniffer allows this architecture to also have information about the wireless medium, which can improve the overall utilization of the system. This allows a better adaptation of resources to the propagation conditions of the medium.

Summarizing, all the architectures aimed at improving QoS in the WiMAX domain. In all of the cases this was accomplished through a cross-layer mechanism. It is clear that a focus on different layers of the system benefits the overall system performance. Still, all these architectures: [21], [22] and [23] do not have many concerns related to the backbone network. The next section will give an overview of how QoS may be accomplished through different technological domains (Heterogeneous Networks).

3.3 QoS solutions for Heterogeneous Networks

As stated before, the 802.16 standard was designed with QoS. Still, as it is mainly used as an Access Network, it will have to inter-work with other QoS architectures. The previously mentioned QoS architectures are strong candidates (specially DiffServ). As such, an inter-connection and adaptation must be made in order to provide QoS to data flows that traverse both the 802.16 and the backbone/core network. All the advantages of the QoS architecture of the 802.16 standard would be lost if there was no relation/signalling between the Access and the core network. In this context, the WiMAX Extension to Isolated Research Data Networks (WEIRD) group defined an architecture which supports end to end

QoS [24]. The architecture defined is shown in figure 9.

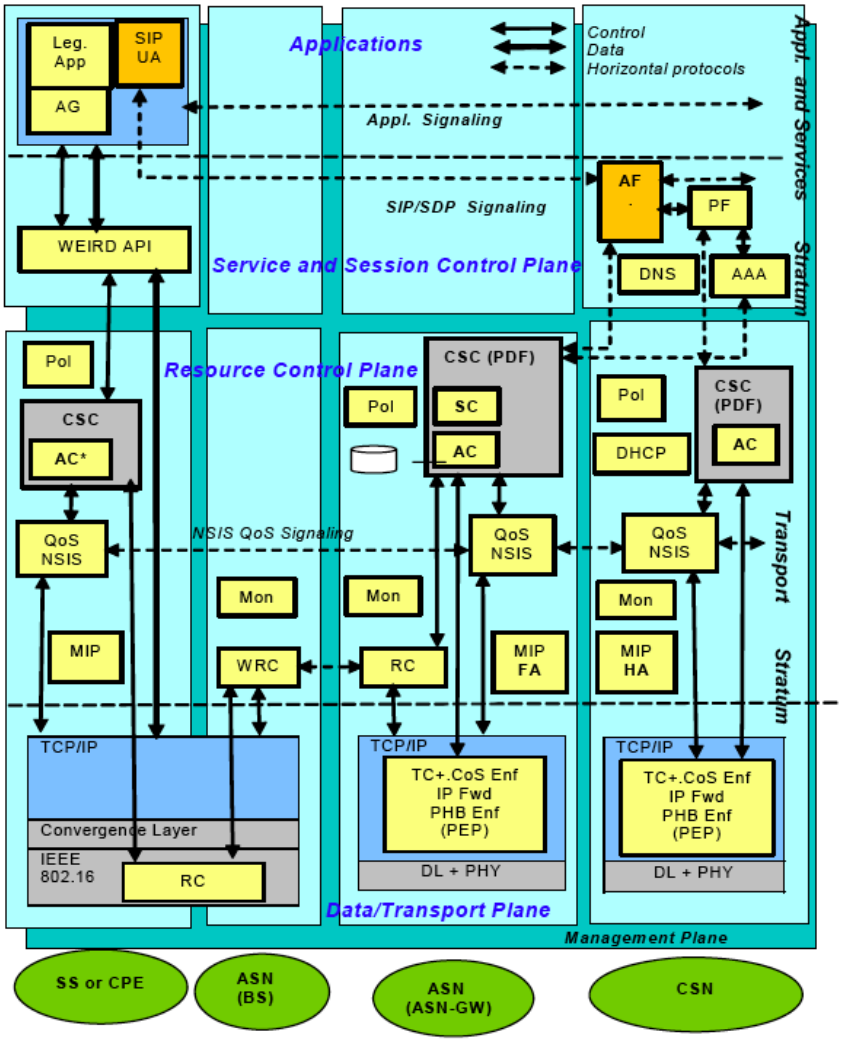

Figure 9: WEIRD architecture

WEIRD supports both signaling capable applications and legacy applications [25]. The former uses SIP, while the latter does not have any signaling capabilities. When an application is using SIP signaling, the Application Function (AF) will intercept the SIP/SDP messages and extract the application's QoS requirements (worst-case scenario). WEIRD considered the signaling process as a two-phase process. The first phase considers the admission and verification of resources. In the 802.16 segment, resources are in an admitted state (SF is not active until a confirmation is given).

The SDP parameters extracted from the SIP messages are passed to CSC modules in the ASN-GW and in the CSN. To interface with the CSC module in the ASN-GW, the Gq' Diameter interface was selected.

To verify the availability of resources in the core network, the NSIS protocol is used (by means of an NSIS query), while in the 802.16 segment, this verification is provided by the Admission Control (AC) function. If the availability of resources is guaranteed, the process may go into step two. The second phase is typically initiated by a SIP 200 OK message. This message will be the confirmation that everything is fine along the path and connection may be established. Finally, resources in the 802.16 segment pass from admitted to provisioned and an NSIS Reserve message is sent through the core network, to establish the QoS required from end to end.

As mentioned earlier, WEIRD also supports legacy applications. Because these applications do not send any information about their bandwidth requirements, the WEIRD architecture includes an agent to deal with the resource reservation. This agent interacts directly with the RC function in the Customer Premisses Equipment (CPE). The NSLP agent in the CPE is the one who starts the messaging for provisioning of resources along the network (it is called a QoS NSIS Initiator (QNI)). This QNI sends a QoS specification (QSPEC) object with all the suitable parameters to describe the service flow that needs to be setup between BS and SS (or CPE). This QSPEC object is then intercepted by the ASN-GW NSLP agent. This agent supports both the WiMAX QoS Model and (for instance) the DiffServ QoS Model. The process still needs to have a confirmation of the resource availability through the data path (802.16 and core network) - same as the SIP scenario. After this confirmation, the NSLP agent builds a new QSPEC based on the DiffServ model and forwards it to the next QNE, that is in the CSN and supports the DiffServ model.

WEIRD was developed to support 802.16, but there are other research groups that also address the subject of Heterogeneous Networks. EuQoS is an example of such projects [26].

In the context of the EuQoS project, an End to End QoS architecture was defined. The aim is not to provide QoS-enabled paths to every traffic flow, but instead, to give Multimedia flows their requirements in terms of bandwidth, jitter, latency, etc when they require them.

The main architectural processes are the following:

- Provisioning
- Invocation
- Operation and Maintenance

The provisioning process is responsible for path setup. Because the scenario is an Heterogeneous network with many Autonomous System (AS), it is responsible for resource provisioning throughout the different domains. In this architecture, these paths are called EQ-PATHs. Because BGP is the *de facto* protocol used by network providers for AS interconnection, it is natural that the EuQoS uses such protocol. In fact, the protocol used is qBGP, which is a variant of BGP, which integrates QoS features. The information provided is then computed to find paths that satisfy flow's requirements.

Once the Provisioning process finishes, it is time for the Invocation process to take place. The architecture has an integrated Resource Manager. The main functions performed at this phase are Verification of available resources (CAC) and forwarding of QoS requirements to the following Resource Manager. The Resource Manager then communicates with a Resource Allocator module (technology dependent) and makes the respective resource provisioning.

The Operation and Maintenance process is responsible for monitoring the EQ-PATH. It should guarantee that the PATH is not overflown and that fault management is guaranteed.

The above gives a brief overview of the architectural components of the EuQoS system. It's purpose is to build, use and monitor end to end QoS paths, with QoS guarantees.

The MESCAL project addressed the problem of *How to engineer the Internet to support QoS across multiple domains*. In [27], the authors start by defining a business model where they identify several actors: Service Providers (SP), IP Network Providers (INP), Physical Network Providers and Users. In MESCAL, the key actors are the INPs. They are the entities responsible for QoS provisioning in their own domain and they need to communicate with other INPs to guarantee QoS over defined paths. This is accomplished through the extension of the intra-domain QoS model proposed by the TEQUILA project [28].

To accomplish their goal, they define a set of QoS-based services. These services are defined as value-added QoS services to customers and establish a relationship between INPs and customers. The relationship between these two is defined as follows: the customer subscribes QoS based services to a certain provider, being this subscription based on an SLA agreement. This SLA contains an SLS, which is the technical part of the SLA (contains the service's characteristics like bandwidth, delay or jitter).

In terms of the SLSs, there were specified two types: Customer SLS (cSLS) and peer SLS (pSLS). The first is the SLS established between the customer and the INP while the latter corresponds to the SLS established between adjacent INPs.

The authors also introduce the notion of QoS class (QC). This notion defines the transfer capability of a provider. This capability includes a set of attribute-value pairs, where packet transfer performance parameters are expressed (e.g. delay, packet loss, jitter). The provider's domain supported QCs are divided in two:

- local QC (l-QC)
- extended QC (e-QC)

The l-QC denotes the QoS transfer capability in the local provider's domain while the e-QC denotes the transfer capability provided by both the local domain and other domains. The e-QC is provided by combining the local capabilities of a domain (l-QC) with a varying number of other domain's capabilities (e-QCs).

The authors also define an Inter-domain peering model. The term peering is used to define the interaction between two adjacent providers to expand their QoS-based service offer. Figure 10 gives an overview of the operation of the model.

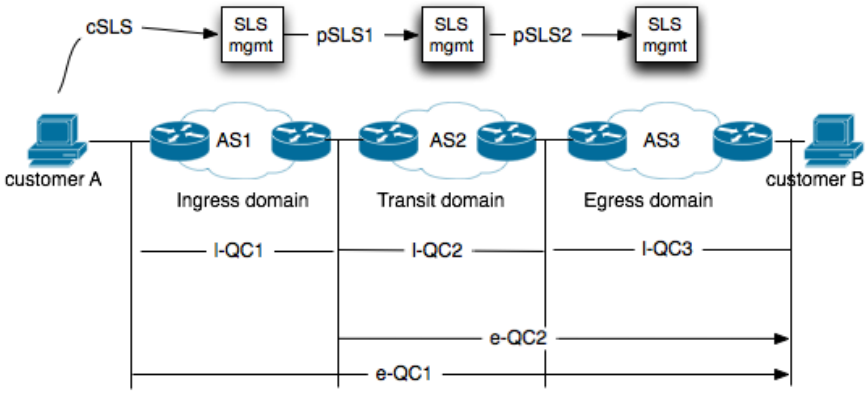

Figure 10: MESCAL cascaded QoS peering model

The model suggested by the authors, and presented in figure 10, is a cascaded model. In this cascaded model only the adjacent Network Providers make pSLS contracts. Thus, the agreements settled are between Network providers that are only one hop away. This type of peering agreement is used to provide QoS connectivity from a customer to destinations that may be several domains away.

In figure 10, l-QC1, l-QC2 and l-QC3 represent the local Quality of Service transfer capabilities of AS1, AS2 and AS3 respectively. pSLS2 represents the contract established by AS2 and AS3. This contract enables the creation of e-QC2. This QC allows customers from AS2 to reach the AS3 domain. In figure 10 it is also represented e-QC1. This QC is the result of l-QC1 (local transfer capability of AS1) plus e-QC2, i.e., AS1 and AS2 negotiate this establishment because only adjacent domains communicate.

For the above establishments to happen, it is necessary that the ASs advertise their l-QCs and e-QCs. This advertisement will allow the adjacent domains to discover the established e-QCs of their neighbors. In [27] it is further proposed a detailed functional architecture that is structured in three planes: Management, Control and Data Plane.

In the Management plane it is worth mentioning the following functionalities: QoS Advertisement / Discovery, the SLS monitoring (to monitor the contracted SLS), Intra/Inter-domain monitoring and Traffic Engineering (to build e-QCs meeting service requirements and optimize intra/inter-domain network resources).

In the Control plane, the emphasis goes to the SLS management (which handles customer's SLS requests and CAC) and Traffic Engineering, with dynamic resource control (intra-domain) and dynamic routing between domains (inter-domain).

The Data plane is responsible for packet classification, identification and traffic shaping (according to

the defined SLS), PHB enforcement and IP forwarding. Further details can be found in [27].

Discussion:

The three projects described: WEIRD [24], EuQoS [26] and MESCAL [27] all deal with the QoS issue in Heterogeneous networks.

WEIRD is more focused on the WiMAX segment. It signals the applications' QoS requirements through the SIP signaling protocol. It then uses this information to make reservations throughout the network. For this effect, the NSIS protocol helps the WEIRD architecture to make QoS reservations along the path.

EuQoS on the other hand is not specific to any specific technology. The architecture proposed focuses on allocating the resources to multimedia streams (which have QoS requirements). The architecture has three main processes (provisioning, invocation and operation and maintenance) which provide the necessary steps towards the path setup and maintenance.

The MESCAL project defined a business model which aims at the creation of relationships between the different entities. The different domain operators (INPs) are seen as different entities who establish contracts (through SLSs) and then discover / advertise paths to neighbors. The cascading model introduced proves to be scalable as only adjacent INPs communicate with each other.

The different projects presented gave an overview of how the Heterogeneous Networks theme may be approached. Three different approaches were presented. The WEIRD project focuses on WiMAX, putting emphasis on the technology and using SIP and NSIS to establish QoS paths. The other two are generic architectures which are technology-agnostic.

When the use of some ideas that are specific to the technology, like the ones presented in [19] and [20], providing a degradation model to new flows that enter the network plus an a-priori flow reservation, are combined with cross-layered QoS architectures like [21], [22] and [23] (which make optimizations at different layers) plus concerns about the rest of the IP network, the result should be a framework that is able to provide the best of both worlds. First, a WiMAX optimized (cross-layered) QoS framework and second, the concern with the inter-connection and path establishment to the rest of the network.

4 Architecture

4.1 Requirement analysis

In the previous section were presented some research efforts in terms of Heterogeneous Networks, namely those that consider WiMAX as the access network. There were addressed subjects like resource provisioning and optimization in the access network segment as well as mapping strategies between domains. Thus, this will be where the main requirements of this dissertation will have their foundations established. The proposed requirements will address each of those subjects, with focus on the access network and on network edge.

Note that requirements 1 to 4 should be considered high-priority requirements in terms of prototype development, while 4 to 6 represent a non-critical status. Also, the non-functional requirements should be seen as *guidelines* in prototype development.

4.1.1 Functional Requirements

1. Interface with the WiMAX Network Elements for resource provisioning

   In terms of the access network, it is necessary to develop an interface with the equipment that provides the dynamic provisioning of resources in the Access network for Downstream and upstream data flows, with the according classifying rules. These resources are considered connections with specific QoS requirements and should be added, deleted or changed according to defined policies.

2. Inter-domain connection

   This requirement aims at the definition of a mapping strategy to adapt QoS-enabled data flows coming from the WiMAX access network to the core and vice-versa. It should define not only the policies used, but also define what are the traffic flows that require quality of service and when should the policies be used.

3. Resource requests

   In order to use higher priority connections, clients must signal their QoS requirements through the use of a signaling protocol. QoS-aware applications may also require the use of QoS-enabled connections based on client requests.

4. Establish interface between Signaling and Resource provisioning

   Signaling and resource provisioning by themselves are not enough, thus, there is a clear need to provision resources after signaling. Integrating the signaling with the provisioning mechanism will allow that connections are added when they are needed and released when they are not necessary.

5. Establish Admission Control policies

The admission control function is necessary in the access network to define policies that allow or deny the entrance of new connections. These policies will define when the connections managed by the resource provisioning function should be added, deleted or simply changed.

6. Integrate the Admission Control policies with the provisioning and signaling mechanism

   In order to apply the defined admission control policies, it is necessary to integrate their function in the system. Thus, the process of signaling of resources must be used by clients, then these requests are processed by the admission control function and finally, if there are enough resources, the connections should be enabled in the network, by means of the resource provisioning mechanism.

4.1.2 Non-functional Requirements

Interoperability:

In terms of interoperability, the prototype should address some concerns. In terms of the protocol stack, the network protocols supported should be IPv4 and IPv6. This is advisable because current networks are in a transition process from IPv4 to IPv6. In terms of equipment, the prototype should be portable across 802.16 compliant equipment (Base Station and Subscriber Station). In terms of OS-dependability, the prototype should be portable to the major OSs (Windows, Linux and MAC OS). In terms of end-user equipment or applications, the use of QoS-enabled connections should not require any changes.

Performance

In terms of performance, there can be considered two different points. The first one relates to dynamic provisioning of resources. This provisioning should not interfere with the user's QoS opinion. The second one is related to resource usage. The introduction of admission control policies, should improve overall bandwidth usage, privileging the higher priority connections.

Network Discovery

The introduction of network discovery can be considered a value-added feature, as it gives the prototype the ability to integrate in a working environment.

Modularity

With a modular approach, the prototype will allow the addition of new features, the change of the signaling protocol used, the change of the mapping rules, if necessary, or even the addition of new functional entities that weren't initially thought of.

4.2 System components

4.2.1 Overview

Figure 11 gives an overview of the main system components.

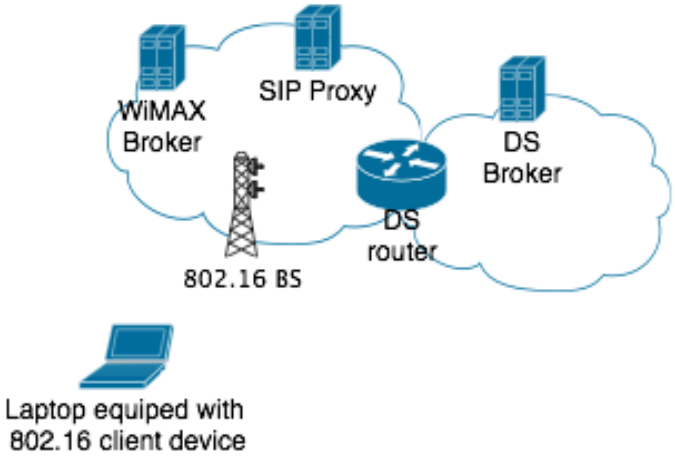

Figure 11: Logical architecture

In the figure are represented two network clouds. These clouds are considered different domains. The differences between these domains are not only in terms of administrative rights, but also in terms of technology. Because of the administrative rights, the policies for resource provisioning may vary. In terms of technology, the used technologies are different, which also implies a different QoS paradigm.

On the right side of figure 11 is represented a core network based on IP and Differentiated Services. Within this cloud there are two entities represented: a router and a DS broker. The router is the entry of the DiffServ network (from the access network perspective) and his functions are traffic shaping/policing and marking. Typically, all the marks that come from the access network are ignored and a new mark is inserted, based on some criteria of the network administrator.

On the left side of figure 11 is represented an 802.16 enabled access network. From a downlink perspective, the BS operates on a point-to-multipoint basis, with the capability of handling multiple independent sectors. From an Uplink perspective, the Subscriber Stations share the wireless medium, competing for transmission. Depending on the class of service being used, the QoS requirements vary, as does the right to transmit. Also represented is a SIP Proxy that will be an entity responsible for the mediation of SIP-enabled applications in the access network. It is there that will be intercepted the QoS requirement's of clients.

The Brokers are considered the *brains* of the operation, with the responsibility of managing the domain policies they are responsible for and also configuring and admitting requests from clients. Thus, the WiMAX broker is responsible for the admission control function and NE configuration in the WiMAX

network, while the DS Broker plays a similar role, with the specificity of a Differentiated Services' domain. Based on specific events, it may change the marking and shaping policies defined in the entry point of the domain (DS router).

An alternative to the use of bandwidth brokers was the inclusion of the broker's functions in the Network Elements themselves (network gateways). This alternative has the advantage of centralizing the functions in one entity (thus allowing a less complex mapping strategy implementation), but has the disadvantage of putting load and unnecessary processing in the network elements. The proposed architecture (with the inclusion of brokers) clearly decouples the data forwarding plane from the control plane.

4.2.2 WiMAX QoS Broker

Figure 12 depicts the components of the access network in terms of functionality.

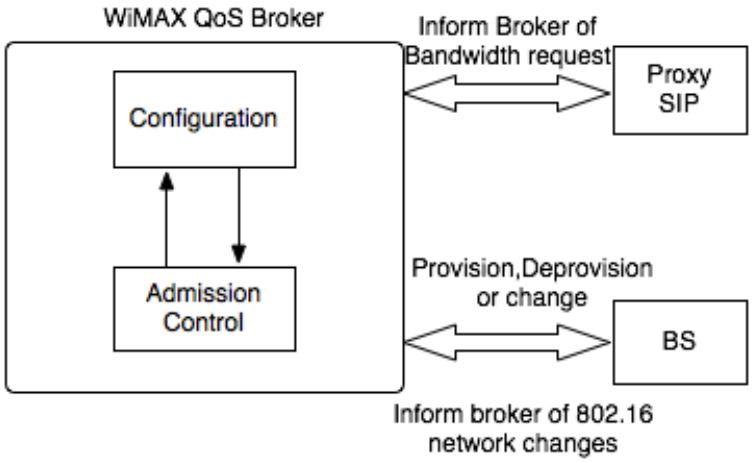

Figure 12: Architecture of the components of the access network

Within the access network there are three main entities: the Base Station, the SIP proxy and the WiMAX QoS Broker. The SIP proxy is responsible for the representation of clients and the inspection of SDP messages to detect communications that have QoS requirements. (The SDP messages inspection will be detailed later. For now, let's only assume that every communication that has QoS requirements needs to signal it.) The SIP Proxy will handle the inspection of messages and will inform the QoS broker that there are new flows trying to enter the network with QoS requirements.

The WiMAX broker will receive the incoming requests from the SIP Proxy and take action based on the requests. Also represented are the components of the WiMAX QoS Broker. These components are called *Configuration* and *Admission Control*. The Configuration component is responsible for every action that is related to the provisioning or de-provisioning of resources in the WiMAX Base Station. In

some special cases it may also change the initially provisioned characteristics of connections. The decision of the admission of resources in the network is done by means of the admission control module.

The Base Station also plays a fundamental role in this architecture. It has information about clients that join or leave the network. Thus, it is part of its' functions to inform the WiMAX QoS Broker when there are network changes. For example, a client that is leaving the network is an excellent situation for freeing the associated resources and giving them to another client that may be needing them.

The purpose of these entities is that they provide the tools to dynamically provision resources in the WiMAX domain. The provisioning of resources dynamically clearly presents itself as an excellent approach (opposed to the static approach), as it allows that resource usage is optimized and QoS is given to clients that are needing it in a determined moment.

4.2.2.1 Configuration

The Configuration module may be considered the main block of the system. Figure 13 is an architectural view of this module. Note that not all subcomponents are represented.

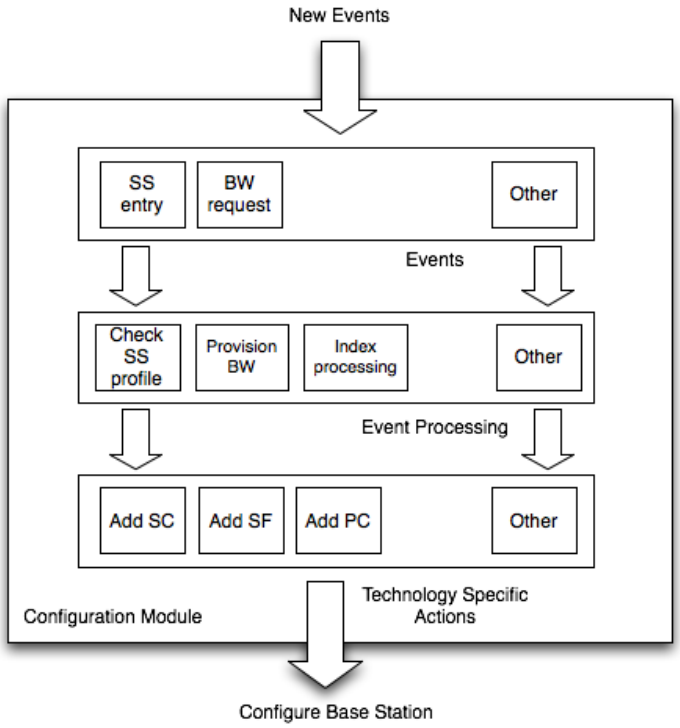

Figure 13: Configuration module architecture

There are three different layers: *Events*, *Event processing* and *Technology Specific Actions*. The upper layer (represented as Events) is the interface to requests from SIP Proxy, Admission Control and Base Station. Each of the events has one or more associated actions that need to be taken before making any

change to the configuration of NEs. This processing is done in the Event Processing layer. When all the processing is done, *Technology Specific Actions* are taken, to make configuration persistent in the NEs.

Let's see an example, to make things more clear. Imagine that an SS entry was detected. The Base Station will send a message to the configuration module. The message is then passed to the Event processing layer. Here, based on an identifier of the SS (mac address), the profile for this specific device will be checked out from the Database and the identifiers for the classifiers, Service Flows, etc will be determined and passed on to the lower layer. From here, the information will be sent to the Base Station for provisioning of resources according to the Profile of the client. Note that, if the profile defined for the client defines classes other than Best Effort, a request must be done to the Admission Control function. Note also, that if the profile has classes of service other than Best Effort, some configurations are needed in the Diffserv domain, to allow the mapping between the 802.16 and the Diffserv domain.

4.2.2.2 Admission Control

This function is integrated in the WiMAX QoS Broker and is divided in two other functions: *Pool of Resources* and *Degradation Model*. Figure 14 illustrates the architecture of WiMAX QoS Broker with emphasis on the Admission Control function.

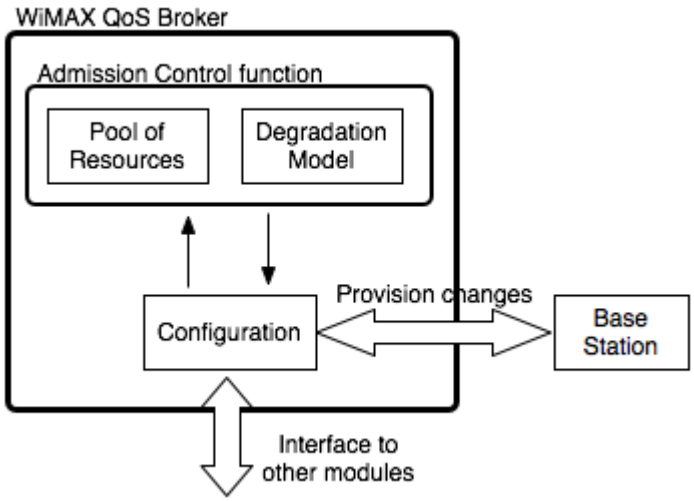

Figure 14: QoS Broker architecture with focus on the Admission Control function

The configuration module interacts with the Admission Control function to make admission control and to inform when network resources are released.

It is also possible to see that there are two functions or entities present in the Admission Control function. The Pool of resources is a mechanism for managing resources in the WiMAX domain, while the degradation model is used when the Pool of resources is exhausted and new connections are trying to enter the network. In these cases, the nrtPS flows are degraded, to allow the entrance of new connections

in the network.

The admission control algorithm is divided in two phases: initialization, where the input parameters are given and some initialization values calculated, and the running phase. The latter will process the incoming connections and finally make a decision. The initialization phase will be described below, while the running phase will be described in the implementation section.

Before going into great details about these phases, let's first look at the general working model. Let's first think of the Total Bandwidth as a whole. Having in mind that for the 802.16d standard we have 3 different scheduling types (UGS, rtPS and nrtPS), let's divide this bandwidth for the three. Think of this as slicing a cheese, where each slice corresponds to a connection.

Another feature of this pool is the concept of steps. These steps define the ammount of resources that are admitted at a given time t. Thus, the resources admitted should always be higher than the number of resources in use.

4.2.2.2.1 Initialization Phase

Let's start by defining some variables and some common language.

Data input values:

- $R_c^{conntype}$ - Rythm used by connection for each conntype (corresponds to a slice).
- conntype - Assumes the values of UGS, rtPS and nrtPS.
- δ_T - measures the reaction of the system to changes.
- δ_N - measures the system's reaction to connection "bursts".
- C - Total capacity of the system in terms of Mbps.
- Step - Number of flows provisioned for each level.

Running values:

- $C^{conntype}$ - Total capacity given to the conntype
- $\mathrm{F}^{conntype}$ - Number of flows in use for that particular conntype.
- F_{MAX} - Maximum number of flows.
- $\mathrm{P}^{conntype}$ - Number of flows provisioned for that particular conntype.
- $\mathrm{N}^{conntype}$ - Current level for a given conntype.
- $U_n^{conntype}$ - Upper Threshold for the level N, for a given conntype.
- $U_{MAX}^{conntype}$ - Maximum Upper Threshold for that conntype

- $L_n^{conntype}$ - Lower Threshold for the level N.
- $B_{available}^{conntype}$ - represents the bandwidth available for a given connection type
- $B_{Step}^{conntype}$ - represents the bandwidth required to go from $N_n \rightarrow N_{n+1}$

As described above, the values are divided in two separate groups: input and running values. The input values are necessary for the startup.

This way, there are a number of values that may be derived from the input values. The capacity given to each connection type may be found taking into account the probability of each of the connection types. Thus:

$$C^{conntype} = C.P(conntype) \tag{1}$$

Another important value that is required for the system to operate is the maximum number of flows available to a specific polling service. This represents the maximum number of slices/flows/connections that are available. The following formula is used to calculate that number:

$$F_{MAX} = \frac{C^{conntype}}{R_c^{conntype}} \tag{2}$$

The level is another fundamental parameter of the system. It is necessary to calculate the Upper and Lower threshold. The N_0 value will be calculated as the rest of the division of F_{MAX} by Step. This way, it is possible to have an initial step, which is variant, depending on the system's input parameters and a fixed Step value. The different N values are calculated as follows:

$$N_0 = restof(\frac{F_{MAX}}{Step}) \rightarrow integerdivision \tag{3}$$

$$N_n = N_{n-1} + Step \tag{4}$$

Based on the values of N_n, δ_N and δ_T, the values of the Upper and Lower threshold for each level are found:

$$U_n^{conntype} = N_n - \delta_N \tag{5}$$

$$L_n^{conntype} = U_n^{conntype} - \delta_T, n > 0 \tag{6}$$

Figure 15 is a graphical representation of how these values are related. The value of δ_N is defined as being the difference between the maximum number of flows for a given level n (N_n) and the upper threshold (U_n). This threshold, as was previously mentioned, is the value that makes the system provision a determined number of extra flows and evolve to the next step.

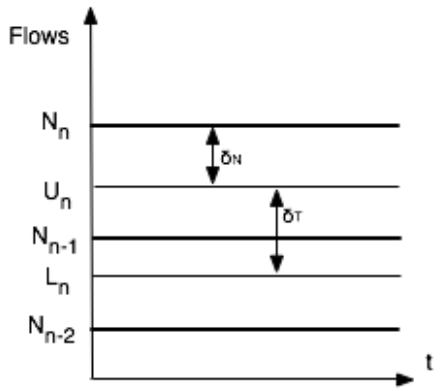

Figure 15: Graphical representation of δ_N, δ_T and N_n and how they are related.

The difference (δ_N) is basically the "burstiness" reaction of the system to new connections, i.e., with a large value of δ_N, the system may absorb a large number of connections at each moment, without having the resources exhausted ($\delta_N + 1$ more precisely).

Still, there is a disadvantage of setting this value very high, as the value of δ_N is the number of resources that the system is not using (it is over-provisioning). Thus, it is necessary to be very careful when choosing this value.

The other value that is of particular interest is the value of δ_T, which will dictate how nervous the system is, i.e., a value of δ_T small, will probably make the system unstable. This will cause a lot of variations, causing a lot of switching between levels.

On the other hand, it is not very efficient to set this value very high, as the system will react very slowly to changes and have a significant number of resources over-provisioned. Remember that the number of used flows is somewhere between U_n and L_n.

Note: the equation $\delta_T + \delta_N > Step$ must be assured. This will be guaranteed if the value δ_T is always equal or greater than the value of Step. This should be assured. Otherwise, the values the thresholds for each level take, do not fit correctly in the windows that figure 15 shows.

Another interesting feature is the opportunity to steal resources from one class to another. If we take into consideration a static model, where resources are evenly distributed through classes, it is likely that at a given time, when many resources of a given type are active simultaneously, one of the classes of the pool is full and the others have plenty of space. To mitigate this problem, let's assume that a given class may steal resources to the other classes if the two following points are true:

- A given class has reached the maximum threshold
- At least one of the other classes has unused capacity available that is equal or higher than the value of $Step.R_c^{conntype}$

If the previous points verify, then the class can steal bandwidth to the class that has the capacity available. The method for determining what class to steal from, goes from a low to high priority (i.e., nrtPS is considered to be of the lowest priority and UGS the highest priority). The algorithm first tries to steal bandwidth from lower priority and then passes to the upper priority classes. This algorithm tries to optimize the capacity used at each time, depending on the network usage of the classes. This mechanism from here on will be called *Stealing Mechanism*.

4.2.2.2.2 Model generalization

The previously described Admission control model is useful if the system is only operating with one profile per Polling Service, i.e., it admits that every new connection that enters the network is only based in one profile. This assumption is normally not adequate with today's access network business models.

Typically, the Access Network providers offer their clients a variety of service bundles and, normally, these are available with different bandwidth profiles. With an offer like this in a WiMAX access network, the previously described Admission Control model would leave a lot of allocated bandwidth without use. This would happen because the value $R_c^{conntype}$ would have to correspond to the maximum bandwidth value offered by the provider.

Let's imagine a simple example to make things more clear. Table 7 presents a hypothetic service offer from a provider.

Package number	**Bandwidth offered (kbps)**
1	512
2	1024
3	2048

Table 7: Hypothetic service offer by an access provider

Imagine that the Service Provider was offering three packages for the nrtPS scheduling service, like table 7 shows. If this operator was using the Admission Control model proposed, he would have to use the value of $R_c^{conntype}$ equal to 2048 kbps. If the clients that occupied the network were all using this profile, he would have no problem, as he was reaching a very good network efficiency level. If, on the other hand, the clients who joined the network were using the other profiles, the network efficiency level would decrease dramatically. For example, when a client using a profile of 512 kbps enters the network, the percentage of bandwidth wasted is 75%.

To solve this problem, let's introduce some changes to the model first described. Summarizing the generalization model, it can be considered as a weighted approach to the value of the client profiles, i.e., each profile has an associated weight. This weight is calculated with equation 7 and is the relationship between the bandwidth used by the profile and the maximum value considered for that polling service.

$$profileWeight = \frac{R_{profile}}{R_c^{conntype}} \tag{7}$$

The introduction of weights in the profiles now allows that the system has allocated, for example 3.5 connections. Although this value is not meaningful in terms of connections provisioned, it symbolizes that, for example, at a particular time, the connections that were allowed in the system correspond to three connections with the highest bandwidth profile plus another, which has an associated bandwidth profile that is half of the maximum.

The weight concept implies that the calculation of the initial step, previously calculated using equation 3, has a decimal part associated. So, the new equation for calculating the initial step(N_0) is represented in equation 8.

$$N_0 = restof(F_{MAX} \div Step) + fractionalpartof(F_{MAX}) \tag{8}$$

Equation 8 adds a fractional part to the initial step value. This is necessary because the number of connections used at a moment may be any value in the range $[0, F_{MAX}]$. The calculation of the current level (equation 4) and the calculation of thresholds (equations 5 and 6) do not need changes. Still, their values will be affected, as the value of N_0 changes.

The method of calculating the number of used connections at a given time also suffers changes. Equation 9 shows the evolution of the method of calculation of $U_c^{conntype}$ (Used connections for a given connection type).

$$U_c^{conntype} = A^{conntype} \rightarrow U_c^{conntype} = \sum_{n=P_1}^{P_n} [A_{(n)}^{conntype}.W_{(n)}^{conntype}] \tag{9}$$

The left side of 9 represents the method of calculation of the used connections when the weight of the Profiles used is considered to be 1, i.e., the profile used for admitting new connections is always the same. $A^{conntype}$ represents the number of active connections for a given connection type(UGS,rtPS and nrtPS). Thus, the value of the total used connections is simply $A^{conntype}$.

On the right side of 9, the number of used connections of a given connection type suffers changes. $A_{(n)}^{conntype}$ denotes the number of active connections of a given profile n, while $W_{(n)}^{conntype}$ denotes the weight of the connections that use that specific profile. Thus, the value of the used connections is calculated as a weighted sum of every profile.

For the purpose of stealing resources, there are two other parameters that are important to the system: the bandwidth available for a given conntype ($B_{available}^{conntype}$) and the bandwidth required to go from $N_n \rightarrow N_{n+1}$. These values can be calculated using Equations 10 and 11.

$$B_{available}^{conntype} = C^{conntype} - (U_c^{conntype}.R_c^{conntype}) \tag{10}$$

$$B_{step}^{conntype} = Step.R_c^{conntype} \tag{11}$$

4.2.2.2.3 Degradation Model

Another component of the Admission Control function is the Degradation Model. This models takes advantage of the scheduling services characteristics (namely, nrtPS). Remember that this scheduling service is designed for data streams that require a minimum data rate (e.g. FTP). For this purpose, Maximum Sustained Traffic Rate and Minimum Reserved Traffic Rate need to be defined as QoS parameters. It may happen that these values are different. If this is the case, there is a range of values in which the real bandwidth of the nrtPS scheduling service may vary.

Let Max_A, Min_A and $DiffA = Max_A - Min_A$ be the Maximum Sustained Traffic Rate, the Minimum Reserved Traffic Rate and DiffA, the difference between Max_A and Min_A respectively, where A represents a Service Class profile. When the network has few resources allocated, the bandwidth given to each nrtPS connection is Max_A, but when the resources reach a critical level, the value Max_A decreases, to a minimum where $Max_A = Min_A$.

Unlike the approach given in [19], where the authors assume a unique Service class profile, this model uses a weighted approach to multiple Service Class profiles.

Figure 16 illustrates a typical scenario where there are defined N profiles for the nrtPS Scheduling Service. Each of these profiles may have associated a random number of clients at each time.

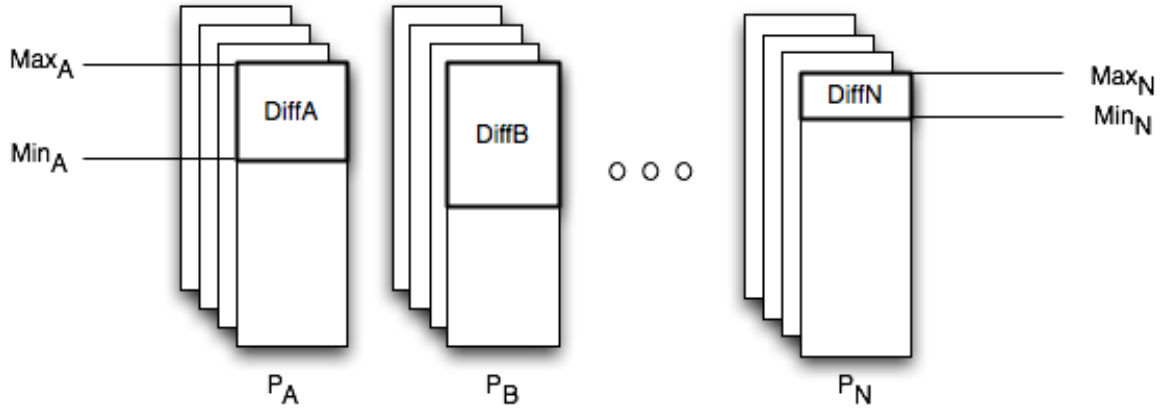

Figure 16: Degradation Model with multiple Profiles

Each profile, in figure 16, has an associated Maximum Sustained Traffic Rate and a Minimum Reserved Traffic Rate different from the other profiles. Furthermore, each of the classes is considered to have an associated weight, represented in figure 16 as P_A, P_B and P_N.

As each of the the profiles have an associated weight and an associated bandwidth available for

degradation, the algorithm will distribute the burden of deterioration among classes based on the relative weight. The definition of the calculation of each value is in section 5.3.1.2.3

4.2.3 DiffServ QoS Broker

Like the QoS Broker described for the WiMAX domain, the DiffServ domain also possesses a similar entity. This broker is the brain of the DiffServ operation. It's main functions are definition of traffic shaping policies, installation of these policies and dynamic provisioning of new policies in the edge router. Figure 17 depicts the context of the DiffServ QoS Broker.

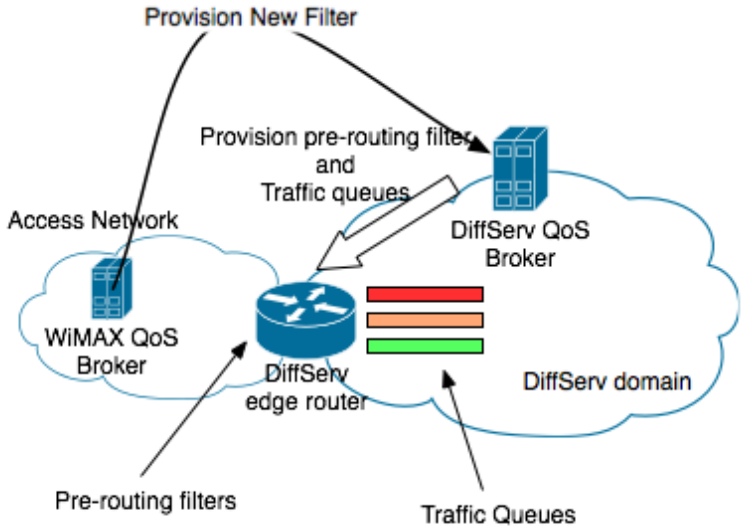

Figure 17: DiffServ QoS Broker contextualization

The figure shows the edge router as the center of attentions. Because it is an edge router, it is responsible for shaping the traffic that enters the DiffServ domain. The figure shows Traffic queues (denoted in red, orange and green) that symbolize the different traffic classes considered (red for EF, orange for the AFxx classes and green for the BE queues). These queues are mark oriented, i.e., the router aggregates the traffic that comes from the Access Network and according to pre-routing filters, marks the packets. According to the mark that was inserted in the pre-routing chain, the packet will be queued in the respective priority queue.

4.3 Protocol layering

The 802.16 quality of service concept is different from the one used in DiffServ. Figure 18 points out the protocol stack layers they are associated to. The 802.16 standard's concept of quality of service is at MAC layer, using a connection-oriented approach, while the IP DiffServ concept is at network layer and uses a non-connection oriented approach.

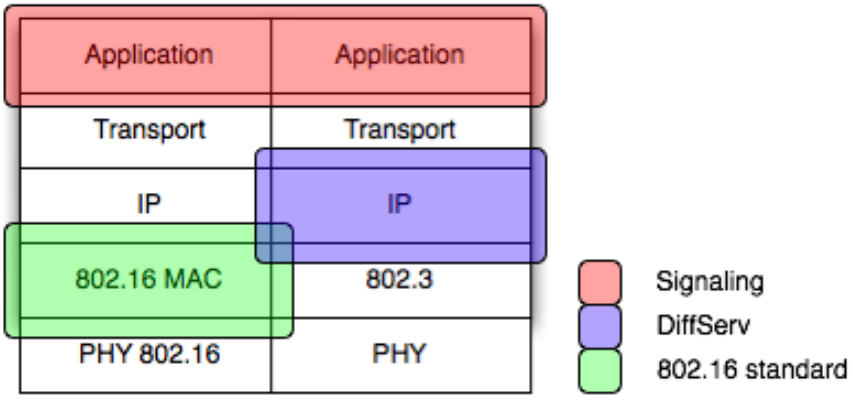

Figure 18: Protocol Layering

At a first glance, the figure clearly shows a challenge that is imposed by the nature of connecting two domains with a different QoS paradigm. A first analysis shows that a mapping strategy needs to be defined for adapting the QoS requirements from one network to another. On the left side of the figure is represented the WiMAX domain, which has the QoS paradigm defined in Layer 2, while on the right, the QoS paradigm is at Layer 3. The Configuration of Network Elements and the Admission Control is performed at these layers, by the corresponding brokers.

On top of the protocol stack is represented the Signaling function, (represented in red) which will be used as the main source for information about client's QoS requirements. Based on the signaling mechanism, the SIP Proxy (which is the associated component) will inspect the client's needs and inform the QoS Broker (which will perform Admission Control and configuration - if necessary). The signaling mechanism is done at the Application Layer and is domain-agnostic, i.e., it works on top of the 802.16 and DiffServ domain.

4.4 Signaling

To allow dynamic allocation of resources in the access network, a signaling mechanism is used. The signaling protocol used is SIP [29].

The idea behind the signaling mechanism is that users ask for bandwidth. Let's assume that, by default, client devices don't have a QoS-enabled path. This path is based on the flows' characteristic and it is first provisioned in the access network and then mapped to the DiffServ domain using a mapping strategy that will be defined ahead. Furthermore, this path should be established when the signaling mechanism is initiated and removed when clients signal that they don't need to use this path anymore.

For this purpose, the entity that takes care of this function is the SIP Proxy. It is the Proxy that is responsible for the mediation of the signaling between clients and also by the inspection of signaling

messages.

4.4.1 Signaling Model

The signaling model used is a typical signaling model with a Proxy performing the mediation of signaling. The particularity of this model is that it considers both SIP-enabled and non-SIP enabled applications. The SIP-enabled applications are those that have embedded SIP-signaling (e.g. VoIP or videoconference applications). The non-SIP enabled Applications are called *Legacy Applications* as they do not possess a signaling mechanism embedded. To deal with the lack of signaling capabilities, a SIP-driver is proposed to interface with the Proxy. Figure 19 illustrates the scenario.

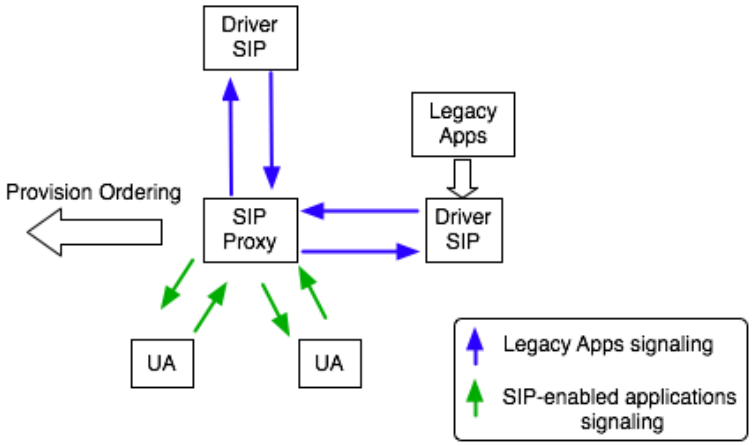

Figure 19: Signaling model

In the center of the scenario is the SIP Proxy. As was previously said, the proxy mediates the signaling between clients. Figure 19 also shows the separation of the SIP-enabled clients (represented as the UAs) from the Legacy Applications.

The Legacy Applications need to have an extra module which enables the signaling characteristics. This module is called a Driver SIP. This driver has two different components. The first one is an interface that Legacy Applications use, while the second one is the calling party of these Applications. This mechanism will obligate the signaling messages to pass through the proxy, allowing their inspection.

There are two possible ways to use the SIP driver: 1) Applications are modified and run the driver in their initialization functions and 2) a wrapper application is created which first calls the driver and then runs the application. For the initialization of the driver it is mandatory that it has access to the application's QoS requirements. The mandatory information is the maximum bandwidth required and the type of flow to be transmitted. The information is then processed by the Driver SIP and is encapsulated in a SIP INVITE message, which is sent to the other SIP driver instance. The SIP Proxy intercepts this communication and will send a Provision Ordering message to the QoS Broker.

The SIP-enabled applications will only require that the underlying SDP protocol is used and it uses the $b=$ modifier (currently optional). Without this, high priority flows will not be provisioned.

4.4.2 QoS based on Application Data

The integration of Application data for connection provisioning is also considered for this project. Figure 20 depicts the general architecture. In red is denoted the path to establish between the Web Server

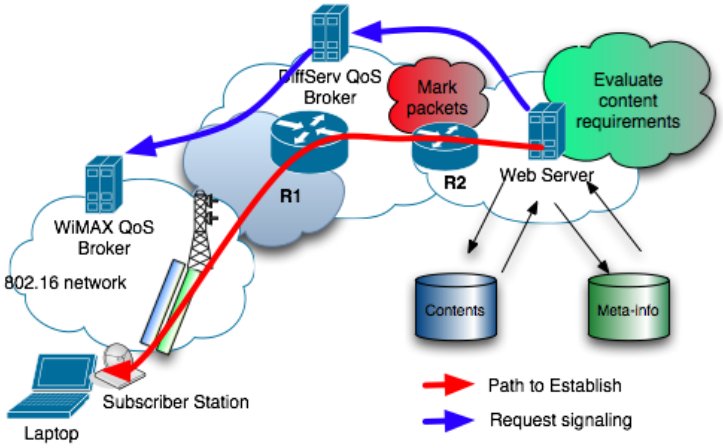

Figure 20: Establishing a Downlink only connection

and the end-user. This path crosses the DiffServ to the WiMAX domain. This path may be considered a *virtual* QoS path, as QoS requirements are preserved across networks.

For the establishment of the path, it is necessary that the Web Server is QoS-aware, thus being capable of differentiating contents' requirements. When a client issues a request to the Web Server, it looks for the appropriate QoS requirements and notifies the Broker of the DiffServ network. This Broker will provide the necessary mapping and forward the request to the WiMAX QoS Broker, where a new connection is provisioned, with the appropriate QoS requirements. The signaling request messages are noted in blue.

5 QoS solution implementation

5.1 Generic Implementation Details

5.1.1 Non-functional requirements matching

Inter-operability:

The prototype was implemented using the Java development environment. This environment allows the prototype to work in every Operating System, provided that the Java environment is installed in the OS that is running the prototype. The reason for the use of the Java development environment is justified because of its OS-independent features. This development fits into the non-functional requirement of inter-operability, which stated that the prototype should be OS independent.

In terms of performance, the Java platform inserts some delay when the application is starting. When the application is running it is expected that the behavior of Java Applications is satisfying. Despite this fact, the prototype is not supposed to have a fast start, so the Java platform is a reasonable working environment for the prototype.

In terms of the support for both versions of IP (v4 and v6), the prototype currently supports IPv4 only. This requirement was softened because of equipment limitations. Currently, the working WiMAX Subscriber Stations only support Classifier Rules in IPv4. The support for IPv6 is not ready at the time of writing, development or testing, so the feature is not yet implemented.

The other non-functional requirement that was initially established: *the ability to make configuration on different vendor equipment* is not completely fulfilled. The explanation is the following. The configuration of Service Flows, Classifier Rules and Service Classes needs to be implemented directly in the Base Station through means of the SNMP protocol or through a Northbound interface. These are the two means that allow the configuration of the WiMAX Base Station (and clients).

The Northbound interface presents itself as a Web Service-based interface that works with XML and SOAP technology, is vendor specific and uses the NMS that shipped originally with the components. Because this interface is vendor specific, the implementation of this interface is not standardized in the 802.16 standard and it is embedded on the vendor's NMS (not providing enough flexibility) , the configuration through this interface was discarded.

The implemented interface with the WiMAX Base Station is the approach based on SNMP. The use of SNMP for configuration of the WiMAX Network Elements is standardized in the 802.16f standard - [31]. This standard defines the management model and the management information base for fixed Broadband Wireless Access networks. Figure 21 shows the reference model defined by the 802.16f standard.

It consists of a Network Management System, managed nodes and Service Flow Database. The managed nodes from the Subscriber and Base Station collect and store the managed objects that are made available to the NMS through the SNMP protocol. The Service Flow database contains the associated

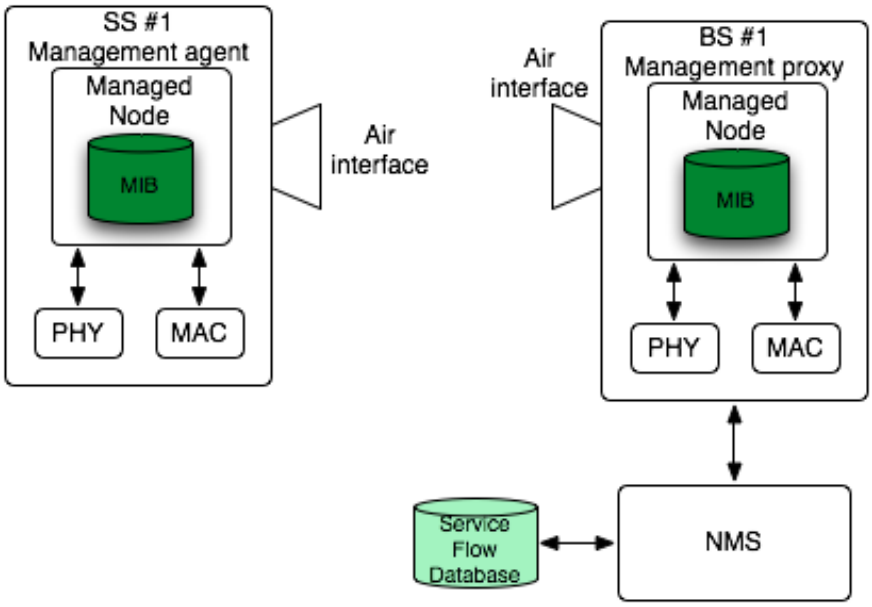

Figure 21: Management reference model for 802.16f Standard

information about QoS. This information should be populated into the managed nodes when they enter the network.

Although the defined Management Information Base is standardized for the managed objects, there are some objects that are vendor specific. As such, the prototype is prepared only to deal with the Airspan Hypermax Base Station managed nodes. This goes against the initial specification of non-functional requirements, where it was defined that it should not be vendor-specific. Still, with few modifications, it should be possible to adapt the prototype to different vendors, provided that they implement the 802.16f standard.

In terms of end-user equipment, the goal of not requiring configuration in end-user equipment was achieved. To address this concern, there was defined a centralized control model (with the definition of Brokers in both networks), which puts the weight of configuration and discovery of client's requirements, in terms of QoS characteristics, in the network, instead of end-user equipment.

Performance:

When it comes to performance, the non-functional requirements defined that the dynamic provisioning of resources should not interfere with the user's QoS opinion. This is relevant because we are talking about network communication, which may add delay in the configuration and cleaning actions.

The version of SNMP used is *v2c*. It provides multiple GETs and SETs to be performed, which saves time in terms of configuration. Thus, for provisioning entries in a table, the Protocol Data Unit (PDU)

is constructed with multiple pairs attribute-value, which improves performance.

Additionally it is considered another aspect. For each SNMP GET or SET operation, there are at least two sockets that need to be opened. In the case of the GET, the socket sends a query for a determined OID and then expects the response from the managed node. The SET operation implies a different mechanism, but still with two sockets needed. One of the sockets is used to send the new value to provision and the second socket awaits the response from the managed node. Because the operations to close sockets tend to be costly in terms of time, these operations should be treated with extreme care.

The interface with the SNMP interfaces of the Base Station are done through means of an open-source java API: SNMP4J ([32]). This API defines an object oriented SNMP API for managers and agents. In terms of performance, this API costs time when it comes to closing all the network sockets. Because we are dealing with a lot of information gathering and setting at each time a new resource enters the network, a method for reducing the time consumed for socket closing operations was implemented.

This method consists in launching a new Thread to close all the sockets that were left open by the SNMP4J API (from here on called TLM or Thread Launching Mechanism). This way, when the application needs to GET or SET values in the Base Station, it creates a new Thread to close all the open sockets. This allows the program to continue and the time that was normally spent closing sockets is decreased.

In terms of the API used to deal with SNMP requests, there was considered another alternative [33]. It is also open-source, but the analysis of the documentation and discussion forums of both open-source projects led to the choice of the SNMP4J API.

Network Discovery:

The network discovery function was not included in the prototype because of time restrictions and the lower importance of this feature.

Modularity:

Modularity was one of the requirements that needed to be inserted in the component specification, in order to be fulfilled. As such, components were designed with this guideline in mind. Along the specification section it is possible to see that the different components are separated and treated as modules. In the case of the Configuration module, it is even divided into layers, that allow modularity of the system. With the fulfillment of this requirement, the addition of new features or processing should be easy to integrate in the already implemented prototype.

5.2 Mapping strategy

The Mapping strategy may be considered a collection of policies that need to be established between both domains. These policies should be defined and provisioned before any data traverses the domains. Note that the definition of these policies does not assume that static rules are implemented in the Network Elements of both networks. In fact, it should be noted that these rules are only applied when there are QoS-enabled data flows traversing the network. Figure 22 illustrates a layered view of the strategy used.

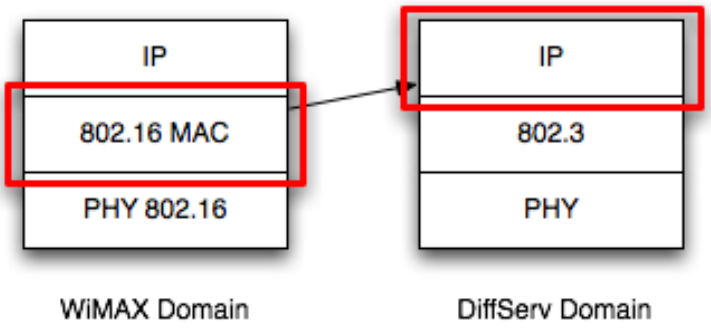

Figure 22: Layers at which the different domains operate

On the left side of figure 22, the WiMAX domain is represented. Remember that the QoS in the 802.16 standard is defined at Layer 2 and is connection oriented. Every client device must have an associated Service Flow (one for each direction). These Service Flows have different scheduling types and the 802.16 frames are served according to these types. Also, the scheduling type is only meaningful in the Uplink direction, because the client devices compete for medium access. This fact justifies the arrow pointing to the DiffServ domain, i.e., it is only relevant to make the Mapping of scheduling types in one direction.

On the Downlink direction, the resource usage is controlled centrally in the Base Station, with the scheduling defined based on traffic characteristics defined by the QoS Parameter Set (e.g. Minimum Reserved and Maximum Sustained Traffic Rate) plus the priority of the Service Flow.

On the right side of figure 22, the DiffServ domain is represented. In the figure, the layer marked with a red rectangle represents the network Layer. It is at this layer of the stack that DiffServ makes it Classification, Queueing and Scheduling or CQS. The protocol considered is the Internet Protocol (IP).

For CQS purposes, the two versions of IP(IPv4 and IPv6) have different fields. In IPv4, the IP packet has an 8 bit field called Type of Service or TOS, while in IPv6 this 8 bit field is called Traffic Class. It is based on this byte that traffic is queued in different classes of traffic. This task is normally performed by edge routers.

What was described shows clearly a difference in the QoS paradigm used by both technologies. First of all, DiffServ is meant to be simple, scalable and make a coarse-grained management of traffic. This imposes a connectionless approach, opposed to the 802.16 approach (which is connection oriented). Sec-

ond, the QoS paradigm design itself is different. In the case of DiffServ, it is presented as a solution to apply in IP-based networks. On the other hand, 802.16 has an embedded QoS concept.

Such differences impose the establishment of certain rules when the traffic is passed from one network to another. Table 8 represents the Mapping strategy from the access to the core network.

802.16d Scheduling class	**IP(DiffServ) - Per Hop Behavior**
Unsolicited Grant Service	Expedited Forwarding
-	Assured Forwarding 4x
real Time Polling Service	Assured Forwarding 3x
-	Assured Forwarding 2x
non-real Time Polling Service	Assured Forwarding 1x
Best Effort	Best Effort

Table 8: Mapping between WiMAX and DiffServ domain

The UGS traffic is directly translated into Expedited Forwarding. This is justified because UGS traffic has hard QoS requirements (VoIP or Leased Line E1/T1). The rtPS traffic is considered to map to the AF3 class because we are talking about traffic with soft QoS requirements. The nrtPS is considered to map to the AF1 class as we are talking about traffic with soft QoS requirements, but with less priority than the rtPS class.

The proposed mapping leaves out two Assured Forwarding classes: AF4 and AF2. These classes are not considered in the mapping model, but they could serve the purpose of future expansions to the core network administrator. These expansions are considered traffic with the same treatment in the access network, but with different treatment in the core network. Consider, for instance, a core operator who makes a distinction between gaming and video traffic. Both these types of traffic have real time requirements, so they will fit into the rtPS class (access), but in the core they may have different treatments (one of the classes mapping to AF4 and the other mapping to the AF3 class).

It should also be noted that the introduction of a new class (by the 802.16e standard - ertPS) will not bring any relevant modifications to the mapping scheme. As ertPS is most suited to VoIP with silence suppression applications, traffic should be treated as having hard QoS requirements, thus mapping from ertPS to EF.

Table 9 represents the mapping from core to access network.
The table shows the proposed mapping for Service Flows in the Downlink Direction. This strategy allows that traffic with different treatment in the core to have also a difference in scheduling in the access network, thus preserving the QoS characteristics of traffic. The traffic priority given to the 802.16 Service Flows is ordered from an upper value (6) to a lower value (1). Six is assigned to traffic with higher priority (EF) and one to traffic with the lowest priority (Best Effort).

IP(DiffServ) Per Hop Behavior	802.16 traffic priority
Expedited Forwarding	6
Assured Forwarding 4x	5
Assured Forwarding 3x	4
Assured Forwarding 2x	3
Assured Forwarding 1x	2
Best Effort	1

Table 9: Mapping Per Hop Behavior to traffic priority

Figures 23 and 24 show the functional entities involved in the mapping process and where are the policies applied in the uplink and downlink direction respectively. In the Uplink mapping, packets are

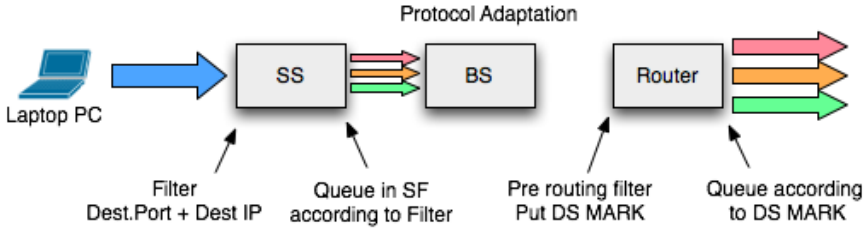

Figure 23: Mapping in the uplink direction

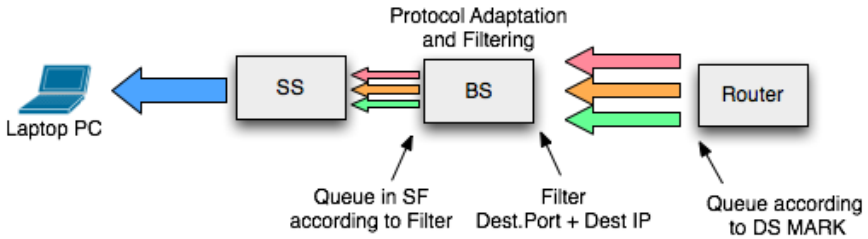

Figure 24: Mapping in the downlink direction

matched to a filter in the Subscriber and put into the according Service Flow. As previously mentioned, this Service Flow is served according to the QoS parameter Set and the associated Polling Service. The Base Station only has functions of protocol adaptation in this scenario (802.16 to 802.3). The router has installed pre-routing filters that mark packets coming from the access network and queues them accordingly.

In the downlink scenario, packets arrive marked to the router and are queued according to this mark. When they reach the Base Station, it will match these packets with a filter and queue them in the appropriate Service Flow(for this scenario the scheduling is done according to the priority).

In both scenarios, the filters provisioned in the Network Elements are based on Destination Port / Transport Protocol and Source IP / Source Mask.

There were other alternatives to this solution, based on 802.1Q (MAC Layer) or TOS byte (Network Layer). Concerning the alternative based on 802.1Q, packets belonging to the same Scheduling Type could be marked using a pre-determined VLAN ID (this would not alter the mapping defined). This would require clients to have a specific interface, ready to deal with packets marked with that VLAN ID.

Concerning a mechanism based on the TOS byte, it would also require that packets belonging to a Scheduling Service to be marked in the client interface. This would put weight once again on client device configuration, requiring all markers to be active before any traffic is sent.

The proposed mechanism based on both transport and network layers puts the weight of configuration in the brokers (through network element configuration). This assumes as a *Zero-Configuration* scenario, which allows rapid deployment in a working environment, with no requirements in terms of client device configuration.

5.3 Components Design

5.3.1 WiMAX QoS Broker

5.3.1.1 Configuration

The following lists will describe in detail the possible *Events*, *Event processing* and *Technology Specific actions* that are part of the Configuration module.

List of events:

- *New Subscriber Station Event* - waits for the entry of a new Subscriber Station in the Network. If this Subscriber Station has a profile associated with it, the provisioning will be made accordingly. If no profile is associated, the default policy is BE.
- *Provision Bandwidth Event* - waits for a bandwidth request. It simply gathers client's requests and sends for processing. Note: this event will also take care of deprovisioning orders.
- *Provision DiffServ Filter Event* - listens for requests with QoS requirements. Remember that, by default, every traffic is sent with no QoS tag in the IP header. This event will trigger the provisioning of a new marker (by the DiffServ QoS Broker) that will satisfy the traffic's requirements.
- *Degradation Event* - This event occurs when a new connection is entering the network but there is no bandwidth available. The degradation event is responsible for computing new nrtPS values, so that the new connection is allowed.

List of processing events:

- *Create Subscriber Station* - Involves the translation of the MAC Address to a value that can be passed to the *Technology Specific Actions*, checking the profile of the user associated to the Subscriber Station and aggregation of Service Flows, Packet Classifiers and Service Classes associated with the SS.

- *Create Default Profile* - The creation of a default Profile is pretty similar to the Subscriber Station creation. This process is triggered when a Subscriber Station does not have a specific profile. This way, clients may access the network on a Best Effort basis, even though they do not have an associated profile. The access to higher priority Scheduling Services will be made based on application requests from end-user applications (namely through SIP signaling).

- *Index Generation and Aggregation* - The indexes are only internal to the System, but it is necessary to generate new Indexes every time there is a change in the system, i.e., when a new Service Flow is created, it needs to be associated to an existing Service Class. Also, there are Packet Classifiers that need to be associated with the Service Flow. This Processing is necessary to guarantee that all these indexes are consistent and bonded.

- *Deprovision Subscriber Station* - The process of deprovisioning Subscriber Stations involves various actions. It involves checking what Service Flows are associated with the Subscriber Station and consequently what Packet Classifiers are associated with them. After discovering what parameters are these, it is possible to deprovision the resources and the Subscriber Station itself, from the Base Station.

- *Provision Bandwidth* - The provisioning of bandwidth is considered as a processing event, as it needs to gather information about classification rules, ports and addresses that are necessary to build the classifying rule in the WiMAX NEs.

List of Technology specific actions:

- *Add Packet Classifier* - Send message to the BS with the values associated to the classifying rules of a certain Service Flow. This classifier may work at different protocol layers (L2 - Ethernet, L3 - IP and L4 - TCP/IP or UDP). At the L2 level, it is possible to make the classification of the packets based on their source/destination mac address, vlan ID or protocol (e.g. ARP). At the IP level, it is possible to define source/destination address and Type of Service bits, while at TCP/IP or UDP level it is possible to classify packets based on their source/destination port.

- *Remove Packet Classifier* - Send a message to the BS to remove the classifying rules of a Service Flow. It should be invoked after the removal of a Service Flow.

- *Add Subscriber Station* - Send message to the BS with the profile defined for the Subscriber Station. The values added are, for example, the DSx allowed messages (which will add, delete and

change Service Flows associated to the SS), the Base Station Allowed Identifier (in cases where the Subscriber Station is only allowed in one sector), the Base Station Mask, SNR protection margins or the Vlan ID of the SS.

- *Remove Subscriber Station* - Will remove the Subscriber Station from the WiMAX domain. Note that this action should only be invoked after the deletion of Packet Classifiers and Service Flows associated with it.
- *Add Service Flow* - Send a message to the BS associating a determined Service Class (UGS, rtPS, nrtPS or BE) to a determined Subscriber Station. Note that Service Flows are unidirectional, so typically a SS has at least two of them provisioned at a time.
- *Remove Service Flow* - Send a message removing a determined Service Flow from a Subscriber Station.
- *Add Service Class* - Send a message with the QoS requirements of the class. These classes define the QoS requirements of a Service Flow. The class defines parameters, such as Scheduling type, jitter, maximum sustained rate, traffic burst or latency. Note that one service class may be used by many Service Flows.
- *Remove Service Class* - Send a message removing a class from the WiMAX domain.
- *Change Service Classes characteristics* - Will send a message to the BS changing the QoS characteristics of a Service Class.

For more information on the events, the reader is advised to consult Appendixes C and F for communication diagrams and message formats definition, respectively.

5.3.1.2 Admission Control

5.3.1.2.1 General Implementation Architecture

The main blocks that compose the architecture of the pool of resources are represented in figure 25.

As figure 25 shows, there are five blocks that compose the Pool:

- Request Listener
- Resource Provisioner
- Admission Control
- Resource Giver
- Resources

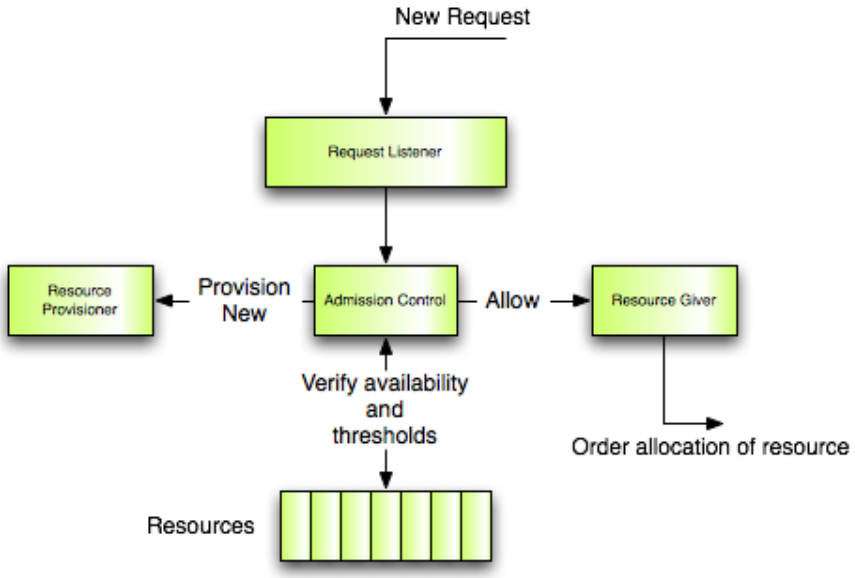

Figure 25: Pool of resources architecture

Each of these blocks has a different function. The Request Listener is responsible for listening for all new requests that arrive at pool, i.e., the information that a CPE needs a specific resource is explicitly requested to the pool. When this request arrives, it is processed by the Request Listener.

After processing the request, the Request Listener passes the information to the Admission Control block. Here, the request is processed in order to allow or deny the new connection. This processing verifies the availability in the Resources pool.

Whenever a new resource/connection is allowed in the network, the Admission Control block notifies the Resource Giver module.

After a resource is given, it is necessary to verify the value of used flows against the value of provisioned flows. A threshold will trigger the provisioning of new connections. This measure will guarantee that the pool does not exaust its' resources before time. In terms of implementation, the Resource Provisioner is integrated with the Admission Control function.

5.3.1.2.2 Running Phase

The initialization phase was already explained in the architecture chapter, explaining the main concepts of the algorithm. Here will be described how the algorithm is implemented.

Figure 26 represents the main states of the Pool. Some of the processing done in each of the conditions was abbreviated and summarized in one action. This is the case of *Steal from others* and *Provision New Request*. The state machine starts with a new connection request. The request is first checked against the profiles that belong to the same Scheduling Service. From these profiles, the Pool will choose the one

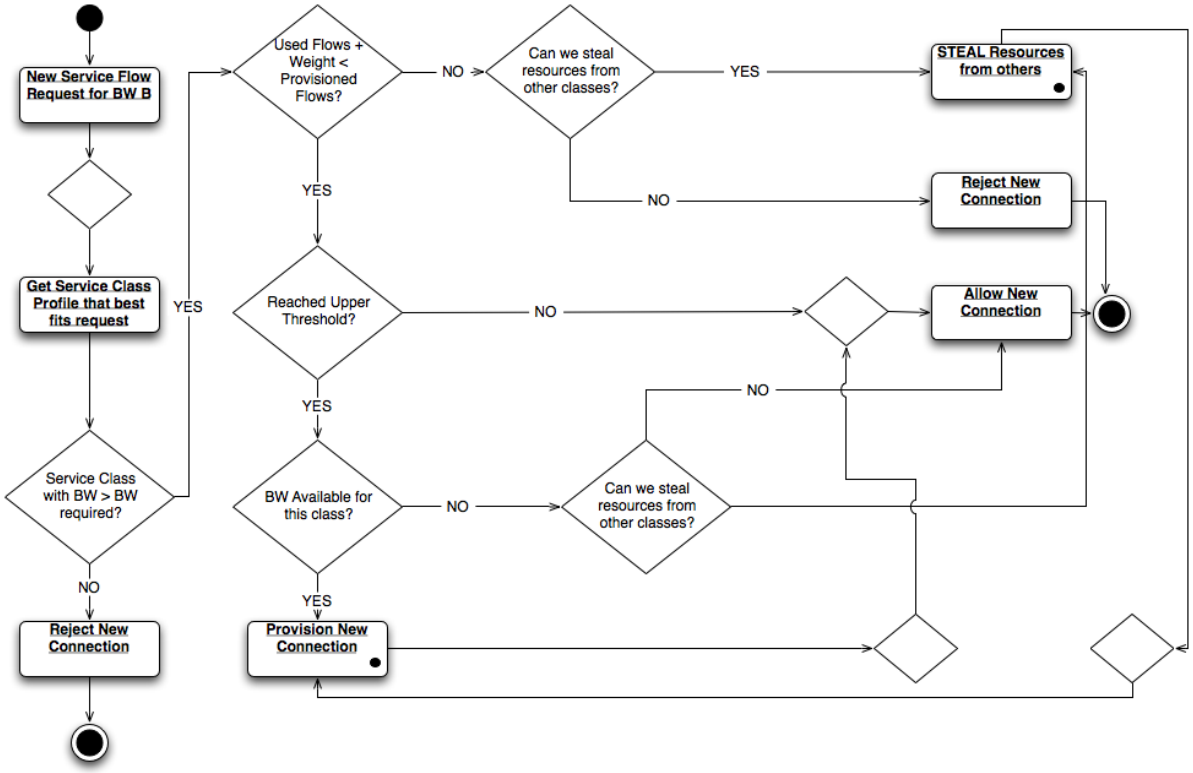

Figure 26: Main state machine of the pool of resources.

that best fits the client's request. In cases where the Bandwidth requested is higher than the maximum available Bandwidth, the connection is immediately rejected.

For the cases that the connection is matched against an existing profile, the algorithm continues. The associated profile has a specific weight associated with it(calculated with equation 7). If adding this value to the number of used connections ($U_c^{conntype}$ - refer to equation 9) is less than the number of currently provisioned network connections, the system must check if it already reached the upper profile for that Scheduling Service (5). If it hasn't, this means that the system is still in the range $[L^n, U^n]$ and the connection is allowed.

If the system has reached an Upper Threshold (U^n), it will need to check if there is still available bandwidth available in the class. If this bandwidth is available, new connections will be provisioned. When there isn't enough bandwidth available to provision new connections, the system will try to steal some resources to other classes.

Going back a little bit, to the point where the $U_c^{conntype} + weight$ of the new connection is checked against the number of provisioned flows, there may exist the situation where the system is near F^{MAX} and it has no bandwidth available. If this happens, it will check if it can steal resources to other classes. This happens if the equation $B_{available}^{conntype} > B_{Step}^{conntype}$ stands. The process of stealing resources and connection

provisioning are explained in more detail in appendix E.

5.3.1.2.3 Degradation Model

In terms of available bandwidth for degradation, the value associated for a given class A, can be calculated with equation 12. B_A, A_A and $DiffA$ represent the bandwidth available for degradation, the number of active connections and the difference between the Maximum and Minimum Traffic Rate respectively, for Profile A.

$$B_A = A_A.DiffA \tag{12}$$

The overall available bandwidth for degradation can, therefore, be calculated using equation 13, where B_O is calculated summing the values of the Available degradation values for each profile.

$$B_O = A_A.DiffA + A_B.DiffB + \cdots + A_N.DiffN \tag{13}$$

Also important is the value that will be degraded for each profile ($B^D_{Profile}$), which is calculated using equation 14, where a represents the relative weight of the profile A.

$$B^D_A = A_A.DiffA.a \tag{14}$$

This way, when a new request for bandwidth arrives, equation 15 must stand.

$$A_A.DiffA.a + A_B.DiffB.b + \cdots + A_N.DiffN.n = bw\_request \tag{15}$$

a, b and n are the associated weights of each class, while bw_request is the bandwidth that must be provisioned for a new bandwidth request. Because at each time a new request enters the network, there is are number of n variables, the associated values of the weights should be calculated relative to the value of a (assuming DiffA, the highest). Thus,

$$b = \frac{DiffA}{DiffB}, c = \frac{DiffA}{DiffC}, \cdots, n = \frac{DiffA}{DiffN} \tag{16}$$

This yields:

$$a = \frac{DiffA}{A_A.(DiffA)^2 + A_B.(DiffB)^2 + \cdots + A_N.(DiffN)^2} \tag{17}$$

The new values for the Maximum Traffic Sustained rates can be calculated with equation 18 :

$$NewMax_A = Max_A - DiffA.a, \cdots, NewMax_N = Max_N - DiffN.n \tag{18}$$

With the introduction of this model, the Pool suffers some changes in the original state machine (represented earlier in figure 26). The new state machine diagram is presented in figure 27.

In figure 27 there were introduced two action blocks and a decision point, before a connection rejection. Remember the state machine presented in figure 26. When there was no available bandwidth to steal from

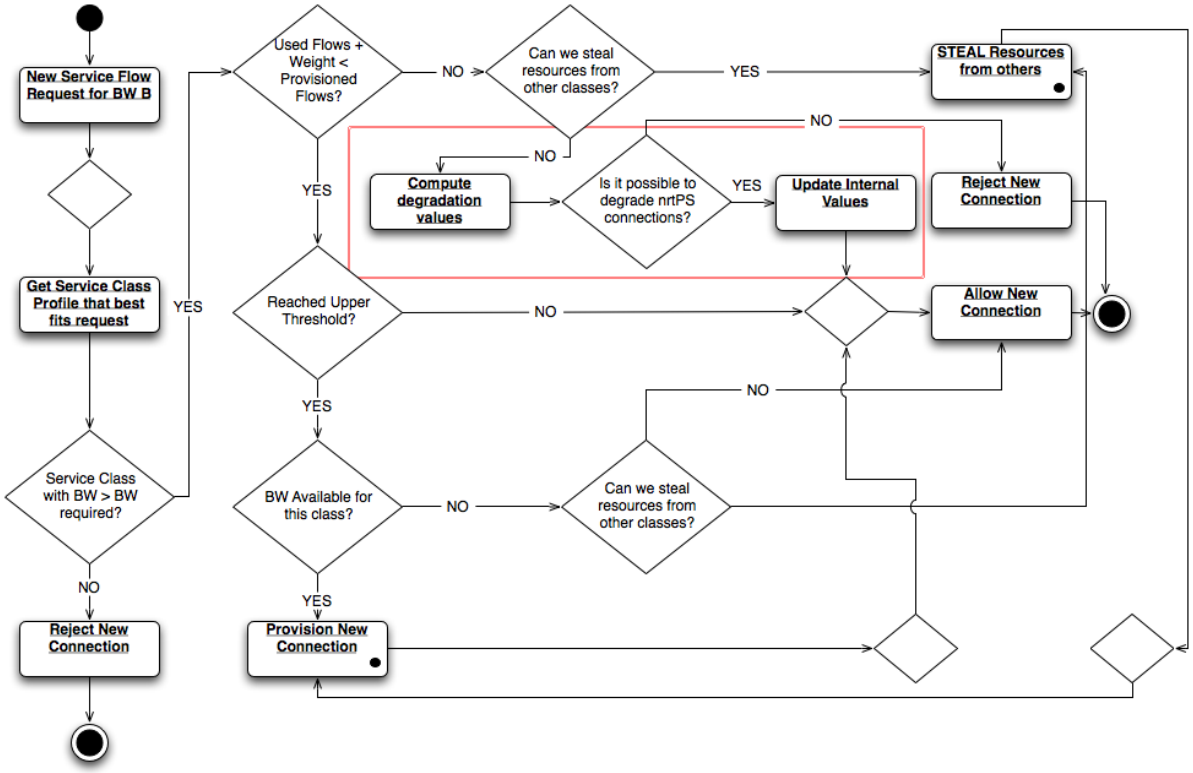

Figure 27: Pool state machine with focus on the degradation model

other classes, the pool would simply reject the new connection. With the degradation model, there is still possibility to allow the connection, if there is enough bandwidth to degrade from the nrtPS connections.

The new blocks inserted in the state machine of figure 27, noted with a surrounding red rectangle, show these blocks. The first step implies the calculation of the degradation values - these values can be computed using the algorithm described previously. Next, the overall bandwidth is calculated with 13. If this value is higher than the requested bandwidth, then the values for degradation are computed. This step implies the calculation of the weights of each Service Class Profile (16 and 17). With these values is possible to calculate the value that each profile will have to degrade (with 18) to allow the new connection in. It should be noted that the process of application of the degradation mechanism is not represented in figure 27 because it is not relevant to the calculation of the values, but it is still mandatory to happen. It requires that a message is sent to the Configuration module, informing it that there is one or more values of the profiles that need to be changed. The configuration module will then process the information and make it persistent in the Base Station. This way, the currently provisioned data flows that are using the nrtPS scheduling service suffer the degradation. For details regarding the communication, the reader should consult Appendix C.

5.3.2 DiffServ QoS Broker

Besides the WiMAX QoS Broker, it was also defined a broker to deal with the configuration and admission control in the core network. It is the DiffServ QoS Broker. For logistic purposes, the Broker is not yet implemented in a physically distinct machine from the WiMAX QoS Broker.

In terms of implementation, for now, the Broker does not control the configuration of traffic queues in the router (this functions is performed separately). Note that these queues are static and they are dimensioned so that packets coming and going to the access network are not dropped due to restrictions in these queues. Still, it has the capability of provisioning and removing dynamically the filters that are necessary to map packets coming from the access (with QoS) to the core network. For configuring these filters, the following attributes are mandatory:

- Source IP
- Network Mask
- Transport Protocol
- Source/Destination Port

The two first attributes are at network Layer. They intend to identify unambiguously the origin of the packets (considering there are no NAT-boxes). The second pair of attributes is considered a glue to the transport layer. Along with these attributes is an associated mark (depending on the traffic's QoS requirements). This forms a pair of (Attributes, mark) for which the packets are matched against.

This cross-layered attribute definition allows the originating host to have two important distinctions: 1) unambiguous distinction from other hosts and 2) a fine-grained distinction of traffic connections originated from the host. This allows the originating host to have different connections, each with different QoS requirements, for each of the requested service. Perhaps the term *connections* is a bit exaggerated in terms of a DiffServ QoS paradigm. However, when we are talking about VoIP, video streaming or multimedia sessions in general, the connection context applies. With this approach, the door opens to the creation of new traffic queues on demand, provided that the router possesses an Admission Control mechanism.

5.3.3 Base Station

The WiMAX Base Station is also considered a system component as it has information that is important to resource optimization. It is there that is possible to watch network entrance/exit of clients, thus providing the necessary information for initial resource reservation or release.

Let's first analyze the SS network entry initialization. Figure 28 shows an abbreviated version of this process (defined in [14] - pp. 168).

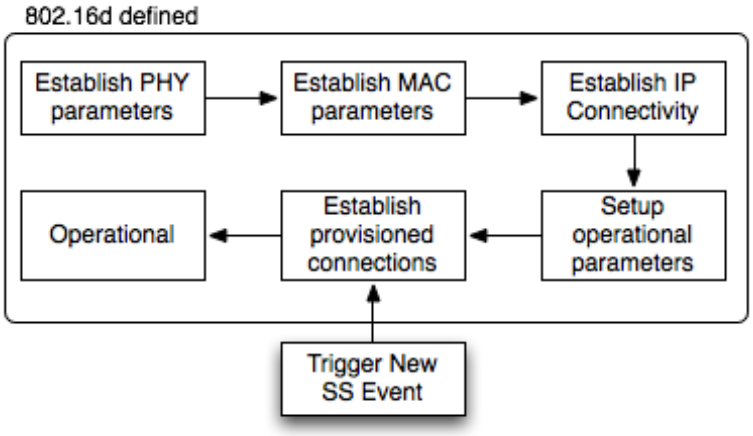

Figure 28: SS initialization overview

The figure is a summary of the tasks performed by an SS when it enters the WiMAX domain. First, it should establish the PHY related parameters (Uplink, Downlink frequencies, channel bandwidth, etc), then it should establish the MAC related parameters, so it can establish the management channels. Finally, it should setup network level and operational parameters. When these tasks are completed, it is still necessary that the Base Station provisions the pre-provisioned connections. It is in this phase that the Base Station should trigger the *New SS Event*, to provision connections associated to the SS's profile. (In terms of the 802.16 standard, the Base Station should send a series of DSA messages to add new connections to the SS.) When all these actions are complete, the Subscriber Station is considered to be in an operational state. For a communication diagram explanation, the user should consult Appendix C.

5.3.4 SIP Proxy

To deal with the message inspection that clients exchange, the Proxy needs to suffer some modifications. Typical Proxy implementations, lack an analysis module that let's them have access to information like Bandwidth required for call, port information and type of media session. Furthermore, the Proxy architecture should also have considered an interface to the entity that deals with the provisioning of resources in the network - in this case it is the WiMAX QoS Broker.

For this purpose, a SIP Proxy Server is enhanced to deal with QoS specific information. Figure 29 is illustrative of the proposed architecture for the modified Proxy.

Typical actors in a SIP environment are the SIP Proxy and the User Agents (represented with the *client signaling* arrow). Because these UAs are registered in the proxy, all the signaling passes through it, even if clients are able to communicate directly. As could be expected, the communication is not being mediated by the SIP Proxy. This task is left to clients.

The remaining entities are normally not present in a SIP proxy. If we take a closer look at the box

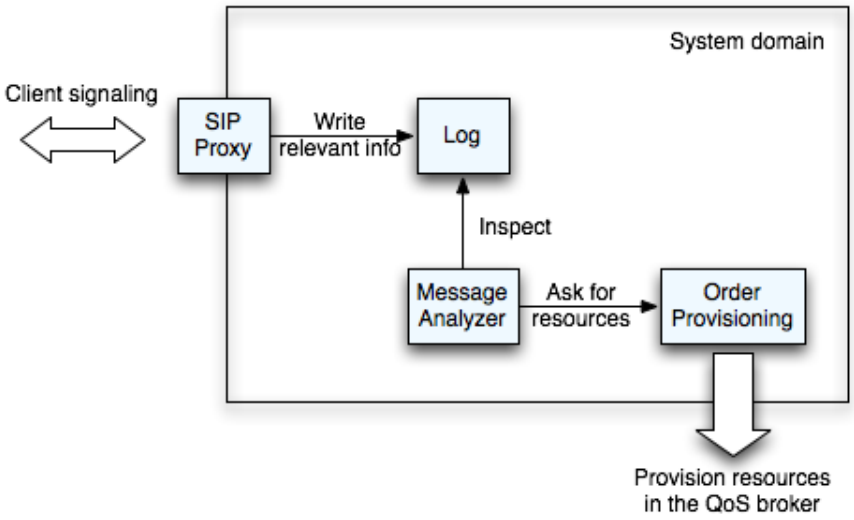

Figure 29: Proposed SIP Proxy Architecture

labeled *System domain*, it is possible to see that the Proxy is covered in the architecture. This happens because the Proxying function should be assured. Still, there is an enhancement that should be taken into consideration: Logging. Typically, the logging process is not enabled in common Proxy implementations. For this purpose, a Loop process should be enabled in the Proxy, giving it the ability to log every signaling message that passes through it.

While the logging process occurs, a new entity (represented in figure 29 as *Message Analyzer*) starts its task. This entity analyzes the SDP messages and keeps a record of every Call that is taking place through time. It should be considered as a list of active calls. The purpose of having such a list is justifiable because this way, the Proxy can control which resources are being used simultaneously. An alternative to these two entities is the coupling of the message analysis module and the logging in one single function. This could be accomplished through the use of a network sniffer (tcpdump is an example). This would not be as effective as the proposed solution because, besides the SIP messages, the application would also capture other packets, which introduces delay. Even with the use of protocol filters, which would improve the delay issue, there is another complex task, which is the parsing of messages, which is complex with these applications.

In figure 29 there is also another entity represented. It is called *Order Provisioning*. As the name indicates, it is responsible for sending information to the WiMAX QoS broker. With this information, the provisioning of resources will be done according to the requests from clients. These requests are considered to be of the type represented in Code Example 1.

Code Example 1 represents a sample of the message body of an INVITE message. It is possible to see in the message that the client is requesting two different sessions (audio and video), and giving the codec options available. What is really interesting for the Message Analyzer is the b=*:* identifier field. The

Code Example 1 Example of SDP message body

```
...
v=0
o=1010 123456 654321 IN IP6 2001:690:2100:b200::201
s=A conversation
c=IN IP6 2001:690:2100:b200::201
t=0 0
m=audio 7078 RTP/AVP 111 110 0 3 8 101
b=AS:80
a=rtpmap:111 speex/16000/1
a=rtpmap:110 speex/8000/1
m=video 9078 RTP/AVP 97 98 99
b=AS:500
a=rtpmap:98 H263-1998/90000
...
```

first field indicates the modifier, while the second indicates the bandwidth that the client is requesting. Note that, for the provisioning of resources occur, it is mandatory that clients send the message with the "b=" field.

In the case of Code Example 1, the UA is asking for 80 kbps for the audio session and 500kbps for the video session. The *Message Analyzer* extracts this information and sends a message to the *Order Provisioning* entity. In *Order Provisioning*, the message will be dispatched to the WiMAX QoS broker for admission control and subsequent provisioning. For more information on the messages exchanged, the reader is advised to read Appendix C.

The Call List that was mentioned previously requires some associations. Figure 30 depicts them.

For a given call there are two associated structures. On the left side are represented the SIP messages. This is a list of the messages already exchanged by the participants. Keeping this list allows the maintenance of state within a call. For instance, consider a conversation where both clients are using only voice and decide to add video to the conversation. In their call are added the new messages exchanged. Another advantage of keeping the messages exchanged is the ability to control in what stage the conversation is in. For example, it is not necessary to provision resources for the call if the calling party does not answer it.

Besides the record of the exchanged messages it is also useful to keep a record of the users that are participating in that call. On the right side of figure 30 is the representation of that process: a Call is considered to have a list of intervenients. This method allows to have a list of participants in a point to

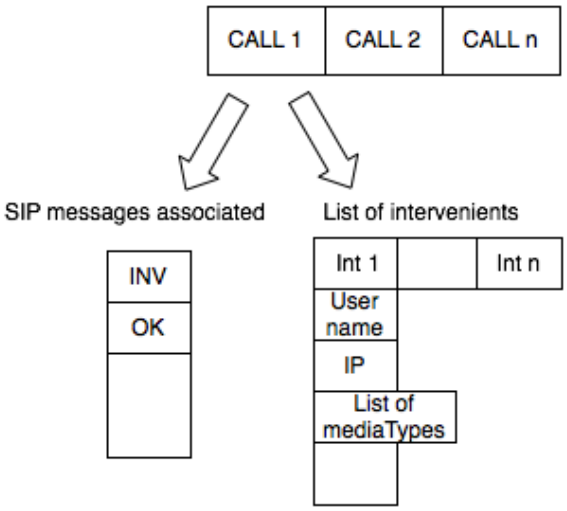

Figure 30: Associations between Call, SIP messages and Intervenients

point conversation or even on a conference call. Each of the participants has associated the information about its' IP address, username and list of media Types currently active. This gives the system the capacity to know who are the participants in the call, where are they and what kind of data flows they are exchanging.

5.4 Providing QoS based on Application data

The base for this feature is the work developed in [16]. To integrate this feature into WiMAX, it was necessary to implement a mechanism that creates a new connection in the WiMAX Base Station, providing the differentiation between contents.

The triggering of this mechanism happens when an 802.16 end-user makes a request for a given content. The web-server processes the request and checks if there is META-info associated to this requested content. If there is, it will extract the parameters Source Port, Destination IP, Source IP, bandwidth required and traffic type. Next, it will send a message to the QoS Broker, for provisioning of a new connection.

When this information arrives at the Broker, it is necessary to determine which Subscriber Station is the requester using. For this purpose, on a regular basis (30 s interval), the Broker checks which clients are associated to which Subscriber Station (consulting the ARP tables of the router and SSs). Thus, when the request arrives, the Broker can immediately know to which Subscriber Station the connection needs to be associated.

The de-provisioning of this connection happens when the end-user clicks the *stop* button in the interface. This triggers the de-provisioning of the connection. For a better understanding of what were the changes to the initial project, the reader should consult Appendix B.

6 Prototype Evaluation

6.1 Test scenario

Figure 31 shows the test scenario used to demonstrate the prototype's functionalities. Represented in

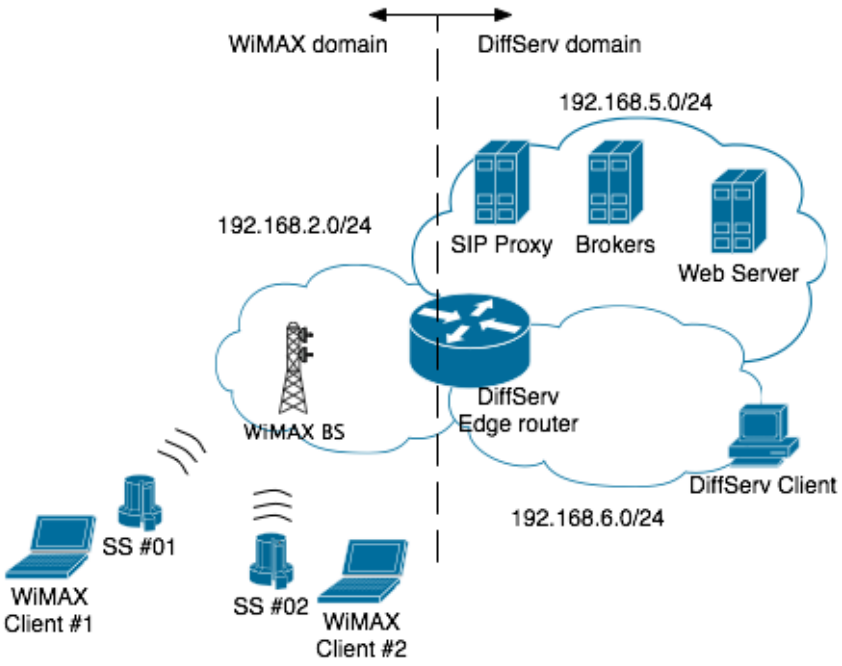

Figure 31: Base test scenario

figure are two different domains: WiMAX and DiffServ. On the WiMAX side, clients connect to a 802.16-2004 CPE. It is represented in figure as SS or Subscriber Station. The protocol layering of the Subscriber Station is described in Appendix A. The SS communicates with the laptop through 802.3 protocol, while the interface between the Base Station and SS is based on the 802.16-2004 standard ([14]).

To connect with the core network, traffic that comes from the WiMAX domain has to pass through a router. This router is the default gateway of the 192.168.2.0/24 network segment.

The DiffServ domain is divided in two different networks (192.168.5.0/24 and 192.168.6.0/24). One of these network contains the Services (in figure are depicted the SIP Proxy, Brokers and Web Server), while the other contains a client.

Ideally, there should be a WiMAX and a DiffServ QoS Broker, as should exist two proxy servers (one for each domain). This would require a great number of machines, so there exists only one SIP Proxy, which serves clients in both domains and the DiffServ QoS Broker was joined with the WiMAX QoS Broker in one machine.

The DiffServ edge router and the Web Server are modified versions of the work developed in [16] (the

modifications are pointed out in appendix B). The router is managed by the DiffServ QoS Broker and the Web Server contains statistics and multimedia contents.

Table 10 contains the IPv4 addresses of the previously mentioned hosts.

Machine Name	**IP Address**
SIP Proxy	192.168.5.101
WiMAX QoS Broker	192.168.5.100
WiMAX Base Station	192.168.2.98
WiMAX Clients	192.168.2.1-192.168.2.254
Edge router int. 0	192.168.2.254
Edge router int. 1	192.168.5.254
Edge router int. 2	192.168.6.254
DiffServ client	192.168.6.10
Web Server	192.168.5.10

Table 10: Mapping between Machine and IP address

The management entities and Network elements all have a defined IP address. The WiMAX clients have a dynamic address which ranges from 192.168.2.1 to 192.168.2.254.

Table 11 defines the profiles that will be used along the tests

ID	**Direction**	**Scheduling Type**	**Max Rate (kbps)**	**Min Rate (kbps)**	**Type**
100	Downlink	BE	8192	0	Pass all
	Uplink	BE	8192	0	Pass all
123	Uplink	nrtPS	1024	512	Port based
144	Uplink	rtPS	512	512	Port based
143	Uplink	rtPS	1024	1024	Port based
166	Uplink	UGS	128	128	Port based
163	Uplink	UGS	1024	1024	Port based

Table 11: Profiles used in tests

The profiles presented in table 11, can be considered connection's QoS characteristics joined with the classifying rules. The ID given to each of these profiles is the real Service class's ID, given by the prototype. It is in the Service Class that are all the QoS definitions. Note that the default profile applied to Subscriber Stations is the referenced in table 11 with ID 100.

For the purpose of testing, a web interface was developed. This web interface contains statistics about the router and the WiMAX Base Station, providing information about provisioned connections and classifying rules.

6.2 Test results and methodology

6.2.1 Functional Tests

6.2.1.1 SS Entry

To perform this test, the user only needs to connect a Subscriber Station. The Base Station detects the entrance of the new client and the WiMAX QoS Broker takes care of resource provisioning. The results are shown in figures 32 and 33. In the figures, it is possible to see, for a particular Subscriber Station,

Results for the Base Station

Mac Address:6.0.160.10.194.71.158
Associated Service Flows:2
Associated Packet Classifiers:2

Service Flow 1 characteristics

Direction	Downstream
State	Active
Scheduling Type	Best Effort
Max Latency (ms)	1000
Max Sustained Rate (bps)	8192000
Max Traffic Burst (bytes)	300000
Min Sustained Rate (bps)	0
Min Reserved Tolerated Rate (bps)	0
SDU Size (bytes)	49
Tolerated Jitter (ms)	1000
Traffic Priority	4

Figure 32: Provisioned Service Flow's characteristics

Packet Classifier characteristics for Service Flow: 1

Classifier Rule Identifier	1
Source Mac Address	00:00:00:00:00:00
Source Mac Mask	00:00:00:00:00:00
Destination Mac Address	00:00:00:00:00:00
Destination Mac Mask	00:00:00:00:00:00
Priority	1

Figure 33: Provisioned Classifier Rule's characteristics

the associated Service Flows and classifying rules. The description of these is summarized into tables. The page contains a description of every Service Flow (direction, state, Scheduling Type and the QoS parameters) and the classifiers that are active for that particular Service Flow. In terms of Service Flows, the parameters showed to the user are always the same. For classifiers, the information changes according to the filter that is provisioned, i.e., if a filter has rules for classifying a Service Flow by mac address, the information shown will be only concerning that particular parameter. If, for instance, the classifier has rules regarding, for example, Source IP or a combination of different parameters, these are all shown to the user viewing the page.

6.2.1.2 Dynamic Service Flow Provisioning

Dynamic Service Flow provisioning is triggered by the establishment of a call between two users. So, the results presented in this test result from the establishment of such a call. Figures 34 and 35 show the example of a multi-provisionioning action. The results shown relate to one WiMAX client and it is possible to see that there are two Service Flows provisioned. One is for the voice stream, which is

Service Flow 5 characteristics

Direction	Upstream
State	Active
Scheduling Type	Unsolicited Grant Service
Max Latency (ms)	200
Max Sustained Rate (bps)	128000
Max Traffic Burst (bytes)	100000
Min Sustained Rate (bps)	128000
Min Reserved Tolerated Rate (bps)	0
SDU Size (bytes)	49
Tolerated Jitter (ms)	100
Traffic Priority	6

Packet Classifier characteristics for Service Flow: 5

Classifier Rule Identifier	1
Destination Port Start	7078
Destination Port End	7078
Destination IP Address	192.168.2.17
Destination IP Address Mask	255.255.255.255
Priority	50

Figure 34: Dynamically Provisioned Service Flow - audio

Service Flow 6 characteristics

Direction	Upstream
State	Active
Scheduling Type	real Time Polling Service
Max Latency (ms)	500
Max Sustained Rate (bps)	1024000
Max Traffic Burst (bytes)	400000
Min Sustained Rate (bps)	1024000
Min Reserved Tolerated Rate (bps)	0
SDU Size (bytes)	49
Tolerated Jitter (ms)	250
Traffic Priority	4

Packet Classifier characteristics for Service Flow: 6

Classifier Rule Identifier	1
Destination Port Start	9078
Destination Port End	9078
Destination IP Address	192.168.2.17
Destination IP Address Mask	255.255.255.255
Priority	50

Figure 35: Dynamically Provisioned Service Flow - video

provisioned as an Unsolicited Grant Service and the other is provisioned as rtPS for the video stream. The Classifiers that apply to them are those that define the Ports and IP addresses of the destination host. Notice that these classifying rules have different priority from those in figure 33, where the priority was 1. This is considered a high priority filter and packets that match this rule use this Service Flow, instead of the lower priority classifiers.

6.2.1.3 Dynamic Path Establishment

The objective of this test is to see if there were provisioned resources in the Base Station (Service Flow and respective Classifier) and also that there was added a new Classifier in the router. As the provisioning of resources in the Base Station was already seen in the previous test, let's focus on the router. As the previous test, this also implies the establishment of a conference call. The two filters provisioned in the router are shown in figures 36 and 37.

The information that is presented is the active filters in router, which show the TOS byte, traffic origin, transport protocol and destination port. Also included is a dump from packet arriving at the DiffServ client (figure 38). It can be seen that there is a difference between the values shown in figures 36 and 37 and in the dump (figure 38). That is because the value that the dump is showing is based on the DSCP field of the IP packet, while in the figures this value is depicted in the TOS format.

Filter 0

Bytes	48110
Destination	0.0.0.0/0
Destination ports	7078
Packets	1090
Transport Protocol	udp
Source	192.168.2.13
Type of Service Byte	0x2e

Figure 36: Audio stream filter

Filter 1

Bytes	579K
Destination	0.0.0.0/0
Destination ports	9078
Packets	473
Transport Protocol	udp
Source	192.168.2.13
Type of Service Byte	0x1a

Figure 37: Video Stream filter

```
IP (tos 0xb8, ttl  63, ...    192.168.2.17.7078 > 192.168.5.10.7078: [udp sum ok] UDP, length 22
IP (tos 0x68, ttl  63, ...      192.168.2.17.9078 > 192.168.5.10.9078: UDP, length 1412
IP (tos 0x0, ttl  64,  ...   192.168.5.10.7078 > 192.168.2.17.7078: [udp sum ok] UDP, length 22
```

Figure 38: Packet dump at the Diffserv client

However, the dump shows that the classifying rules are being correctly applied to the traffic that is coming from the WiMAX client to the DiffServ client.

This test also intended to assess if a connection's QoS requirements are being fulfilled and if the mapping is being done properly. In figure 39 are represented the AF classes. The represented AF class

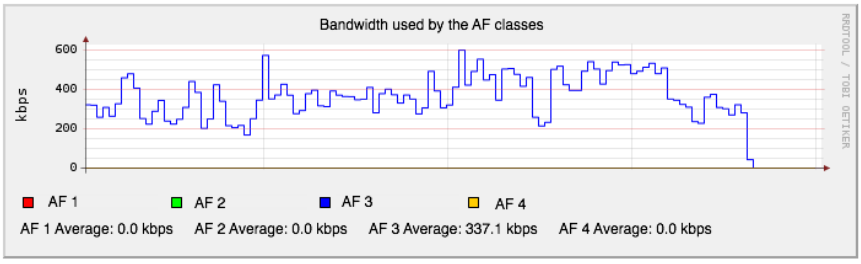

Figure 39: Bandwidth given to the AFx classes in the router

is the one that relates to video (mapping rtPS to an AF3 class).

In figure 40 is represented the background traffic that was generated during the test and the EF traffic (audio). The background traffic generation was initiated after the conference connection was launched. It

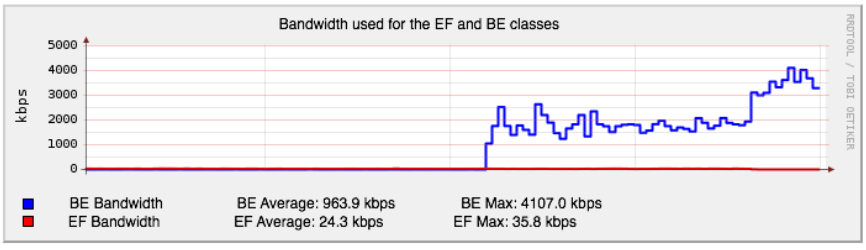

Figure 40: Background traffic generated while a multimedia session was taking place

can be seen that the video connection didn't suffer with the introduction of this background. During the

test was also verified that in terms of user interaction, the connection wasn't also affected. No block effect or delay were introduced. When comparing figures 39 and 40 it is possible to see that, when the video stopped, the Best Effort connection had more bandwidth available and therefore, the used bandwidth increased. The audio traffic does not play a significant role in this scenario, as the bandwidth used by it is typically very low, compared to the video or Best Effort traffic rates.

6.2.1.4 Threshold Fluctuation

Table 12 shows the reference values used in the admission control function for the subsequent tests (related with the admission control function). From here on, all the tests related with the Admission control function will have these values as reference.

Parameter	**Value**
Total Pool Capacity	24 Mbps
Class capacity	8 Mbps
Slice Size	1Mbps
Step	2
δ_N	1
δ_T	2

Table 12: Reference values for the admission control module

To test the threshold fluctuation, there were made four bandwidth requests and then, three of them were released. The objective in this test is to watch the variation of thresholds with the entrance of exit of new clients. Figure 41 shows the threshold fluctuation in the Admission control module according to these values. It is possible to see that the number of used flows by clients is always in the range [L,U],

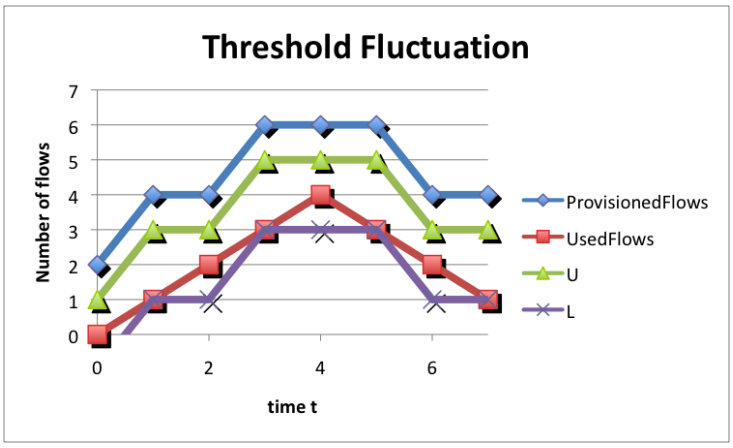

Figure 41: Test to the threshold fluctuation

with the number of provisioned connections being increased along with the threshold values.

6.2.1.5 Dynamic Stealing

Testing the dynamic stealing mechanism implies that a given class steals resources to another class. In this test this is what will happen, but also, the opposite, i.e., the stolen class will claim those resources back. One of the classes will be "neutral", as the other classes won't be able to steal bandwidth from it (UGS). The other two classes (rtPS and nrtPS) will each steal resources from one another, but at different moments in time. First, the rtPS will steal resources from the nrtPS class. Then, the number of used flows of the rtPS class will drop and the number of nrtPS clients will rise to a point where it will steal the resources back. The objective is to show the dynamic stealing mechanism working.

Sequence of events:

1. Six UGS clients enter the network, each with a 1Mbps flow
2. Ten rtPS clients enter the network, each with a 1Mbps flow
3. Six rtPS clients leave the network, each with a 1Mbps flow
4. Ten nrtPS clients enter the network, each with a 1Mbps flow

Figures 42 and 43 show the dynamic stealing mechanism. Figure 42 shows the evolution of the number

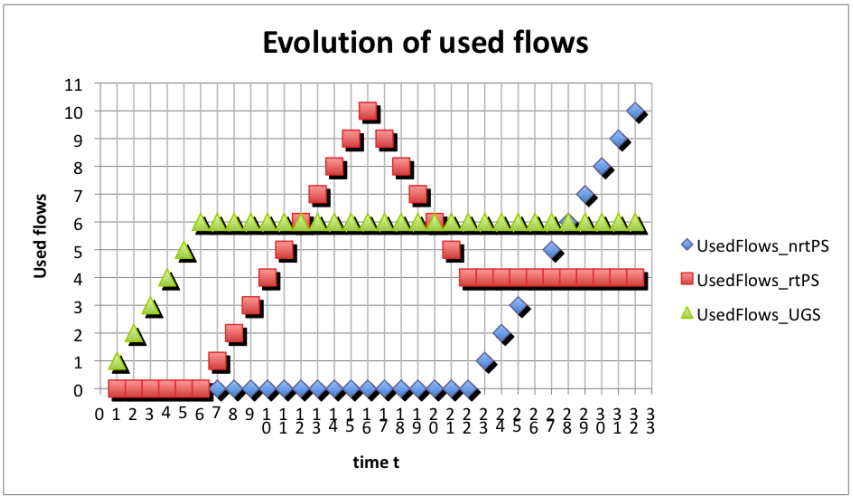

Figure 42: Dynamic Stealing - evolution of used flows

of used flows through time. Figure 43 is also relevant to evaluate, because it shows the bandwidth that is dedicated to each class. In a first moment, the bandwidth that is being given to each class is equal (8Mbps in this case). As resources of the rtPS class enter the network, and there is enough bandwidth in the nrtPS class, the value is transfered. That can be seen in figure 43 at time t=13. The number of used flows rises (refer to figure 42) and so does the bandwidth, dropping the bandwidth used by the nrtPS class. In the second moment, at t=27, the inverse happens. The number of flows used by the rtPS class decreases, which leaves space for the entrance of nrtPS clients. This time, the stealer class is the nrtPS and the stolen is the rtPS.

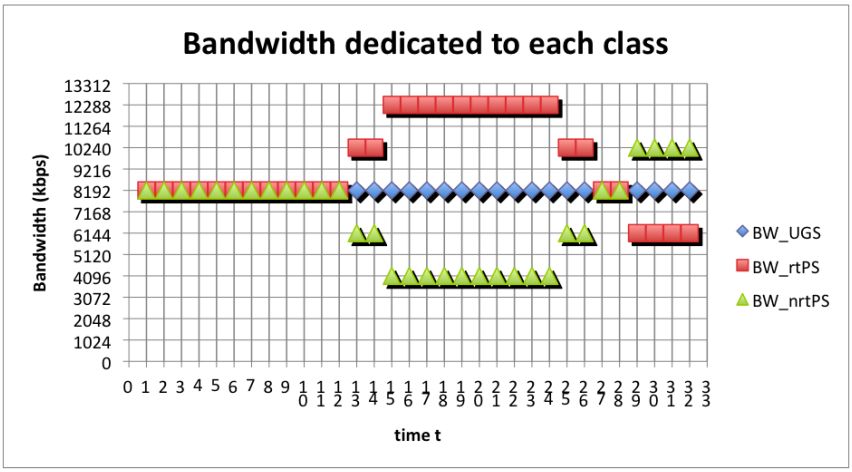

Figure 43: Dynamic Stealing - Bandwidth dedicated to each class

6.2.1.6 Bandwidth Degradation

Bandwidth degradation happens after the stealing of resources. This way, the test will first show the stealing of bandwidth from one class and then evolves to a state where the bandwidth is being degraded in the nrtPS class. The variables in this test will be the rtPS and nrtPS classes. The UGS class is considered neutral once again. The objective is to watch the evolution of bandwidth used, when the degradation process occurs. Note that the values considered for the test are those referred in table 12.

Sequence of events:

1. Six UGS connections enter the network (1 Mbps for each).
2. Four nrtPS connections request bandwidth, each with Max. Sustained Rate = 1Mbps and Minimum Sustained Rate = 512kbps
3. Twelve rtPS connections enter the network, each with a 1Mbps flow

Figures 44 and 45 show the evolution of the bandwidth usage in each of the classes. In a first moment ($t \in [1, 6]$) the UGS connections enter the network. In a second moment ($t \in [7, 10]$), the nrtPS connections enter the network. From that point on, only rtPS connections enter network. At t = 17, the rtPS class starts stealing resources to the nrtPS class, which guarantees more five rtPS connections. At t = 20, the rtPS class gives the last connection. The next two connections will need to degrade the bandwidth of the nrtPS connections to enter the network.

The total value that can be degraded is 2Mbps (512kbps from each of the nrtPS classes). To allow a first rtPS connection, the nrtPS degrades 256kbps from each of the classes, passing from a usage of 4096 to 3072 kbps and then, when a new rtPS connection enters the network, the nrtPS drops some more (to the 2048 kbps). This is the maximum that it can drop. At this point, all the nrtPS connections have dropped to a minimum and can not be further degraded.

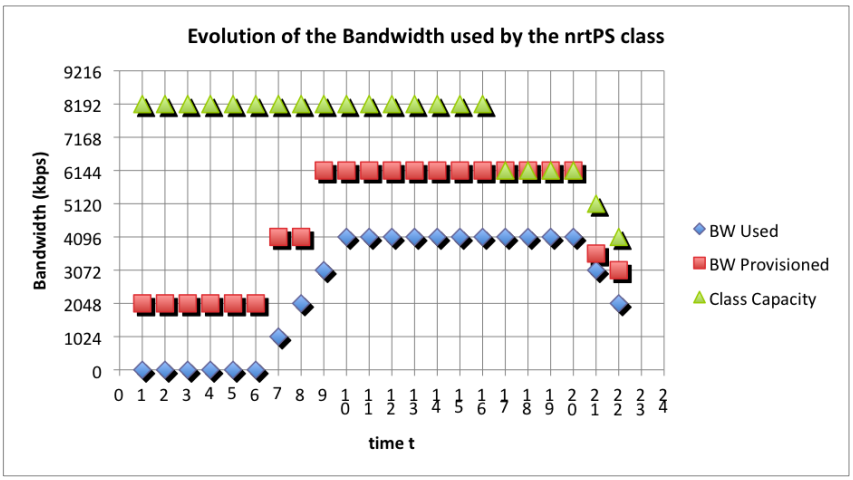

Figure 44: Degradation - Evolution of bandwidth used in the nrtPS class

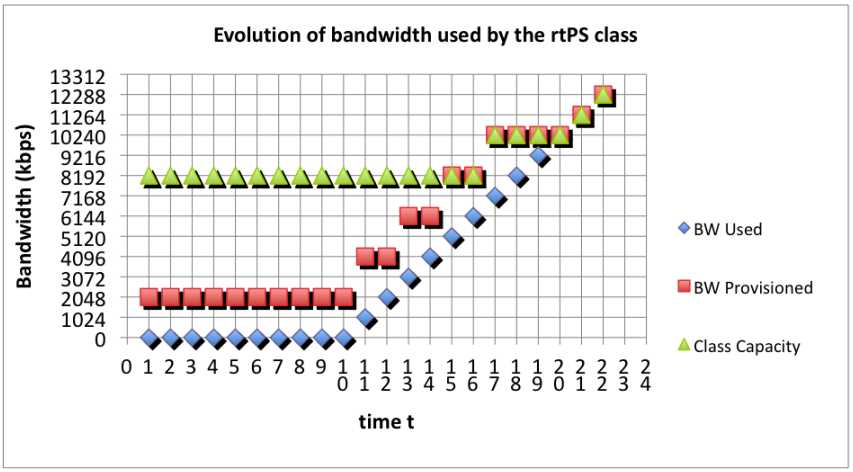

Figure 45: Degradation - Evolution of the bandwidth used in the rtPS class

Figure 46 shows the bandwidth usage of a connection in the process. In the figure it is possible to

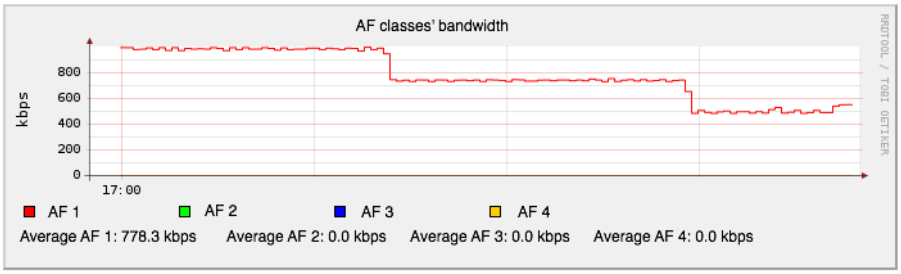

Figure 46: Degradation - bandwidth used by an nrtPS connection in the degradation process

see three steps. The first one is when the maximum sustained rate of the class is 1 Mbps. When the bandwidth that was initially given to the class is degraded, this value drops to 768 kbps and finally, the

bandwidth drops to the minimum allowed, which is 512 kbps.

6.2.2 Performance Tests

6.2.2.1 Time to provision and de-provision without TLM

The objective of this test is the measurement of the time that is necessary between detection and configuration of a connection. Table 13 presents the time that is spent in provisioning and deprovisioning dynamically a connection in the WiMAX Base Station. The experiments were performed as many times as necessary, so that the results have statistical relevance. Note that these results represent the time between the detection of a new request and the configuration of the associated connection in the Base Station.

Parameter	**Average (ms)**	**Standard Deviation**	**Confidence Interval (ms)**
Time to provision	12086	35.5	16.4
Time to de-provision	4038	31.1	14.4

Table 13: Values obtained without the Thread Launching Mechanism (95% C.I.)

As can be seen in table 13, the time to provision is significantly higher than the time to deprovision. This has one simple explanation. The number of sockets that need to be opened and closed for the operation to complete are directly proportional to the time spent in the process. In the provisioning process, first a new entry in the SNMP table needs to be created (1 socket), then the values are inserted (1 socket) and finally, the entry is marked as active (1 socket). This means that three sockets need to be opened and closed every time a connection is necessary. The time to provision is 12086 $\pm$ 16.4 ms, which is a very high value in terms of connection setup.

In the deprovisioning process, only one socket needs to be opened and closed. This happens because the entry only needs to be deleted from the table. This result is clearly shown in the table, where the provisioning process takes about three times longer than the deprovisioning process. The de-provisioning process total time is 4038 $\pm$14.4 ms.

6.2.2.2 Time to provision and de-provision using TLM

The objective of this test is the measurement of the time that is necessary between detection and configuration of a connection. Table 14 has the results relative to the time spent with the introduction of the TLM mechanism. These results were obtained in a multithreading-enabled processor. (2 GHz Intel Core 2 Duo - 667MHz bus). Note that these results represent the time between the detection of a new request and the configuration of the associated connection in the Base Station.

Parameter	Average (ms)	Standard Deviation	Confidence Interval (ms)
Time to provision	70.6	14.1	6.5
Time to de-provision	36.8	18.7	8.6

Table 14: Values obtained with the Thread Launching Mechanism enabled (95% C.I.)

The results obtained in table 14 show that the time spent in the process is still proportional to the number of sockets being opened/closed. Despite this fact, the time involved in both operations has decreased dramatically with the introduction of the TLM mechanism. In both cases, the time spent decreased from magnitude of seconds to values below 100 ms. The provisioning process now takes 70.6 $\pm$ 6.5 ms, while the de-provisioning process takes 36.8 $\pm$ 8.6 ms.

6.2.2.3 Time to change connection's characteristics

The objective of this test is the measurement of the time necessary to change a running connection's configuration. This case occurs when it is necessary to change a parameter in a connection (through degradation). Table 15 shows the results obtained for tests to the degradation process.

Parameter	Average (ms)	Standard Deviation	Confidence Interval (ms)
Time to degrade	57.7	8.8	5.5

Table 15: Time spent in the degradation process (95% C.I.)

The values represented in the table are the time that is necessary to make changes to connection's characteristics plus the time spent in processing. Compared to the time spent in deprovisioning (TLM enabled), the values are higher: 57.7 $\pm$ 5.5 ms.

This time assumes a higher value as it involves processing time in calculation of new values for the degraded classes and also because of the delay introduced by the communication between entities (depicted in Appendix C - Resource Degradation).

6.2.2.4 Influence of optimization strategies

To test the influence of bandwidth optimization solutions (dynamic stealing and degradation), the system was tested with the same input (i.e., client requests) for the three cases:

1. Base system, without dynamic stealing and Resource degradation
2. System with dynamic stealing, but without Resource degradation
3. System with dynamic stealing and Resource degradation enabled

The input values for the admission control function are those defined in table 12. The results presented in figures 47 and 48 represent two different scenarios. The objective is to watch the influence of the

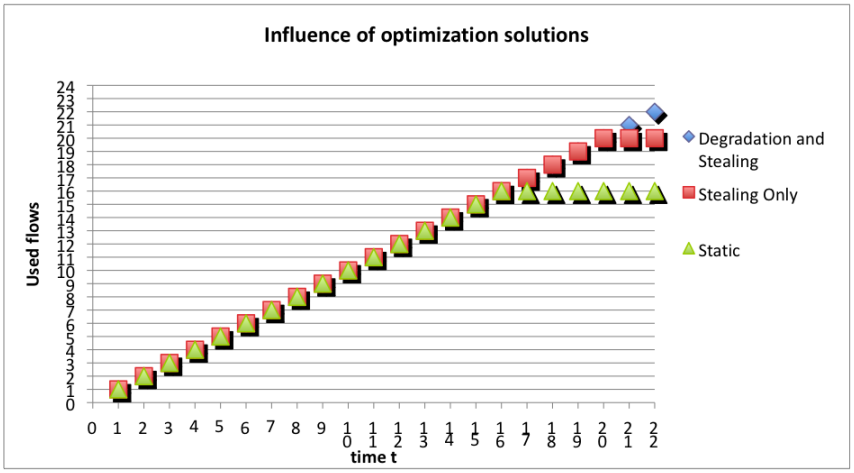

Figure 47: Influence of the optimization solutions in terms of used flows

introduced optimization strategies for two different scenarios. In the first scenario (figure 47), the sequence of events is the following:

1. Four nrtPS connections enter the network (Maximum Sustained Rate = 1Mbps and Minimum Sustained Rate = 512kbps).
2. Four UGS connections enter the network (1Mbps)
3. Fourteen rtPS connections enter the network (1Mbps)

This situation represents an abnormal distribution of the connections through classes, so it does not favor the Static allocation of resources, as can be seen in figure 47. This happens because, with that approach, each class only has 8Mbps statically assigned. Watching the use of the stealing mechanism, it can be seen that, for this scenario, the results are better than using statically assigned values (also depicted in figure 47). This happens because the connections were not distributed evenly through classes. The degradation of resources along with stealing proved, for this scenario, to be the best approach. However, one should not forget that the degradation implies that the QoS characteristics of the nrtPS clients are degraded in favor of other connections.

Figure 48 represents a different scenario. The sequence of events is the following:

1. Eight nrtPS connections enter the network (Maximum Sustained Rate = 1Mbps and Minimum Sustained Rate = 512kbps).
2. Eight UGS connections enter the network (1Mbps)
3. Eight rtPS connections enter the network (1Mbps)

This scenario intends to show an equal distribution of connections through classes. This scenario favors the Static approach, which fills exactly all the bandwidth given to each class, reaching the maximum capacity. On the other hand, the stealing mechanism is not favored with this distribution. It is possible

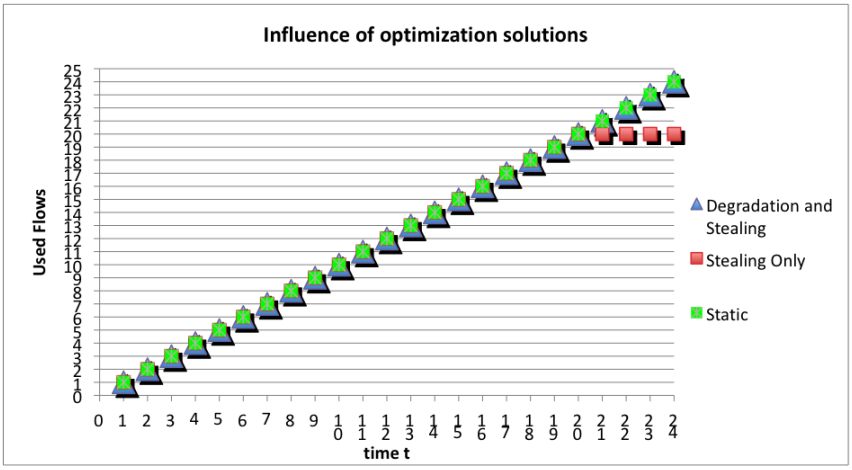

Figure 48: Influence of the optimization solutions in terms of used flows

to see in figure 48 that the number of flows allowed is actually lower. This happens because, when a class steals bandwidth, it will steal a value that is equal to it's step. So, in this case, the UGS and nrtPS had still 2 connections available each, while the rtPS connections were rejected. If we take a look at the degradation values, we see that, for this scenario it still performs well. It reaches the same levels of the static approach, but with degradation of the nrtPS classes. Although the number of used flows of the static and the degradation plus stealing mechanism are the same at this point, it should not be forgotten that the first approach is now full and has no spare connections, while the latter still has two (1Mbps) nrtPS connections plus two (1Mbps) UGS connections to give.

6.2.2.5 Influence of background traffic in a video-conference call

This test is an analysis to the behavior of traffic while two WiMAX clients are in a video-conference call. The objective is to establish a comparison between the behavior of the video, audio and Best Effort traffic with and without the dynamic provisioning of Service Flows. The procedure for this test is the following:

1. Establish two background traffic connections between Subscriber Station 01 and 02 (control sample)
2. Establish a video-conference call between SS 01 and 02 (audio and video)
3. During conference generate two background traffic connections
4. Hang-up the conference call
5. Connect modified proxy
6. Establish video-conference call
7. During conference generate two background traffic connections
8. Hang-up conference call
9. Generate two background traffic connections (second control sample)

For this test was considered a two minute time-window, with the period of the generated background traffic being 10 seconds. Furthermore, the audio connection was provisioned as a 128kbps UGS connection while the video connection was provisioned as a 512kbps rtPS connection.

Figures 49, 50 and 51 represent the results obtained during the test. The graphics presented in

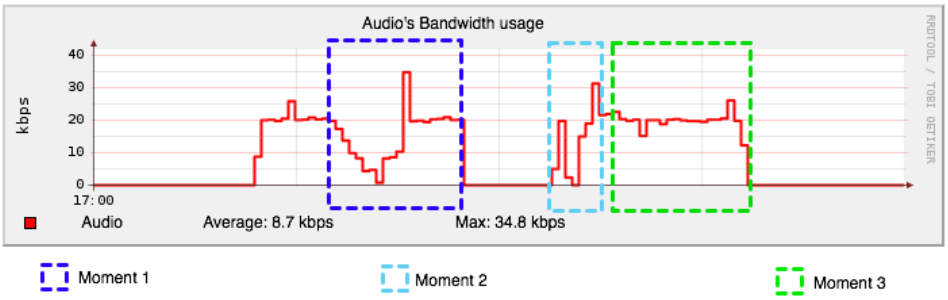

Figure 49: Audio usage through time

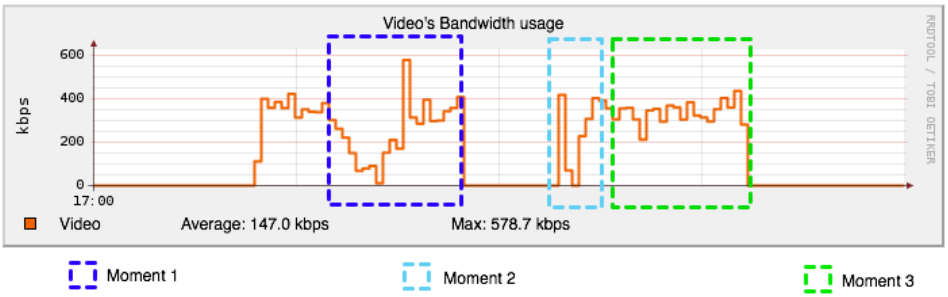

Figure 50: Video usage through time

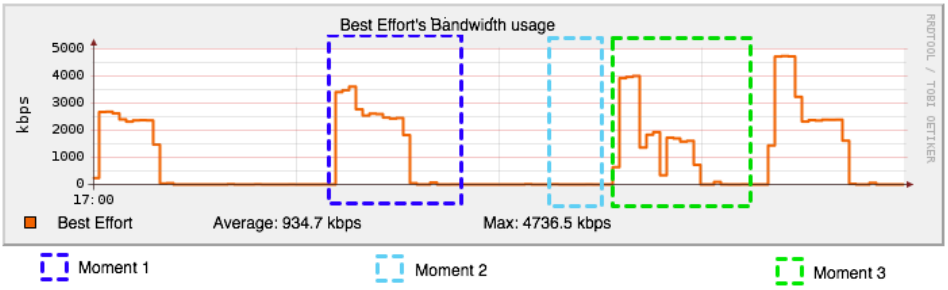

Figure 51: Simulation of background traffic

the figures where highlighted with the three key moments of the test. The first moment represents the beginning of the background traffic connections (represented in figure 51). If we take a look at figures 49 and 50, it is possible to see that the bandwidth given to them decays with the introduction of background traffic. When this traffic stops, the bandwidth given to the audio and video connections stabilizes.

In Moment 2 is represented the establishment of the conference call, now with the modified Proxy and Broker connected. Because new Service Flows are being provisioned in each Subscriber Station, the traffic is unstable for one or two seconds.

When the traffic stabilizes, new background traffic is generated in both clients. This represents moment 3. Unlike moment 1, the video and audio connections do not suffer from the introduction of background traffic. The values are stable and the best effort traffic is the one that suffers at this moment.

Finally, the control samples that were described in the procedure can be compared with the samples in moments 1 and 3. In figure 51 it is possible to see that in moment 3 is when the background traffic suffers the most. Compared to all other samples, this is, in average, the lowest. Regarding moment 1, the background traffic didn't suffer much and is similar to the control samples.

6.2.3 Discussion

The tests presented show that the prototype is performing as expected. There were presented functional tests that demonstrate the dynamic provisioning of both Base Station and router. These tests also demonstrate that the SIP Proxy communicates with the WiMAX QoS Broker, thus providing client's QoS requirements. In terms of the admission control function, there were also demonstrated the three associated mechanism: threshold fluctuation, dynamic stealing and degradation.

In terms of performance, the time spent in resource provisioning, de-provisioning and degradation achieved satisfying results, with the introduction of the Thread Launching Mechanism, with times under 100 ms.

The introduction of the threshold fluctuation, dynamic stealing and degradation model in the admission control function also showed that, in some situations the system can benefit from the introduction of such mechanisms. The scenarios tested compared symmetry and asymmetry of types of connections in use. For the latter, the use of dynamic stealing and degradation provided better results than a static approach, because the class with *connection bursts* could borrow resources from the other classes. In the scenario where classes had an equal distribution, the stealing mechanism proved to have a worse performance.

7 Conclusions

7.1 General Conclusions

This dissertation proposes a mechanism for Service Flow management in an 802.16d compliant network. The singularity in this mechanism is that is does it in a dynamic fashion. Based on requests from SIP-enabled applications, the values are extracted, processed and then provisioned in the access network, to provide clients the resources they need. Along with the dynamically provisioning of QoS connections comes also the dynamic de-provisioning of these resources. This allows optimization in the access network, as the de-provisioned connections can be given to other clients.

Along with the provisioning and de-provisioning of resources, an admission control model was introduced with the singularity of stealing resources between different classes of service, so that, when a given class is having *connection bursts*, the bandwidth may be re-distributed among classes. Additionally a model for degradation of less-prioritary resources was introduced. This allowed that new clients requesting higher priority services to be served, while degrading the lower priority connections.

The usage of the system in a heterogeneous environment was also accounted for. For this purpose, a mapping mechanism was proposed, with the intent to provide QoS-enabled paths throughout the network (with the integration with an IP-DiffServ enabled network).

Additionally, the functions of network element configuration and admission control were decoupled from the data forwarding functions through the creation of bandwidth brokers. These brokers are considered specific to network domain they belong to and are responsible for all the management regarding resource provisioning, policy installation, classifier provisioning and admission control. The singularity in these brokers is the fact that they use a modular approach, allowing the introduction of new functionalities easily.

In terms of tests, they were divided in two separate groups: functional and performance. Along with the tests came a base scenario that defined the entities that were considered for the tests. The functional tests aimed at showing the basic functionalities of the prototype using a test web page. This page contained information about both gateways (Base Station and DiffServ router).

The performance tests evaluated the prototype in different perspectives. One of the important points in the provisioning of resources is the time that is spent in the process, which led to results under 100 ms (these are results obtained with the TLM). These results could be considered quite satisfying as this time should not be noticed by a user in the establishment of a connection. Despite this fact, it was noticed in one of the tests that the provisioning of resources implied a drop of packets in the moment of provisioning, which may be justified by some limitation in the 802.16 Base Station.

Besides the time spent in provisioning, the admission control function was also evaluated in terms of bandwidth usage, which resulted in some gains when *connection bursts* occurred.

Also evaluated was the influence of background traffic in the network with and without the use of

prototype. The results showed that the introduction of the proposed mechanisms led to a clear distinction between high priority flows (like audio and video) from the background traffic.

The outcome of the project is not only the prototype itself, but also an API that allows configuration and gathering of information from the 802.16 equipment. This API involved a lot of work in the reverse engineering process, as the vendor did not provide any documentation on this matter. This way, it is now possible to configure the 802.16 Base Station without the use of the vendor's NMS. This API assumes itself as a tool for development and experimentation of a new set of tasks.

Additionally, the web interface used for evaluation may be used as a platform for testing and evaluation of other projects that use the 802.16 Base Station.

7.2 Future Work

Although the presented solution is working, thus reaching the goals that were proposed, further tests should be conducted, to evaluate the existence of inefficiencies. The only goal that was not reached was the support for IPv6 in both access and core network. This occurred due to device limitations in the 802.16 network. If the vendor's support for this technology is not embedded soon, the problem could be solved by the implementation of a NAT-PT mechanism [35], which currently lacks an implementation in Linux.

In terms of the 802.16 standard, with the introduction of mobility support in the 802.16e standard, the prototype could be extended to support mobility (802.16e standard compliant equipment), which would bring new challenges, mainly in terms of the admission control function.

Concerning the admission control function, an analytical model that evaluates the model itself should be developed, so that the efficiency of the model is analyzed in mathematical terms.

In terms of the core network, besides the IP-DiffServ core that was used in the evaluation of the prototype, the implementation could be extended to support other technologies (like IP-MPLS, IntServ or ATM for example).

In terms of the bandwidth brokers, it should be considered in a first step their physical separation and it should also be considered the implementation of a mechanism of dynamic queue adjustment in the DiffServ QoS Broker. Along with these enhancements, it should be considered the usage of a standardized signaling protocol, for network communication between brokers.

Additionally, a topic that was not subject of research in this dissertation but could prove to be a value-added feature is the inclusion of advanced packet scheduling mechanisms in the Base Station, to improve overall system performance. In terms of the 802.16 standard this is a research topic as it was left undefined by the standard specification.

A - Subscriber Station Protocol Stack

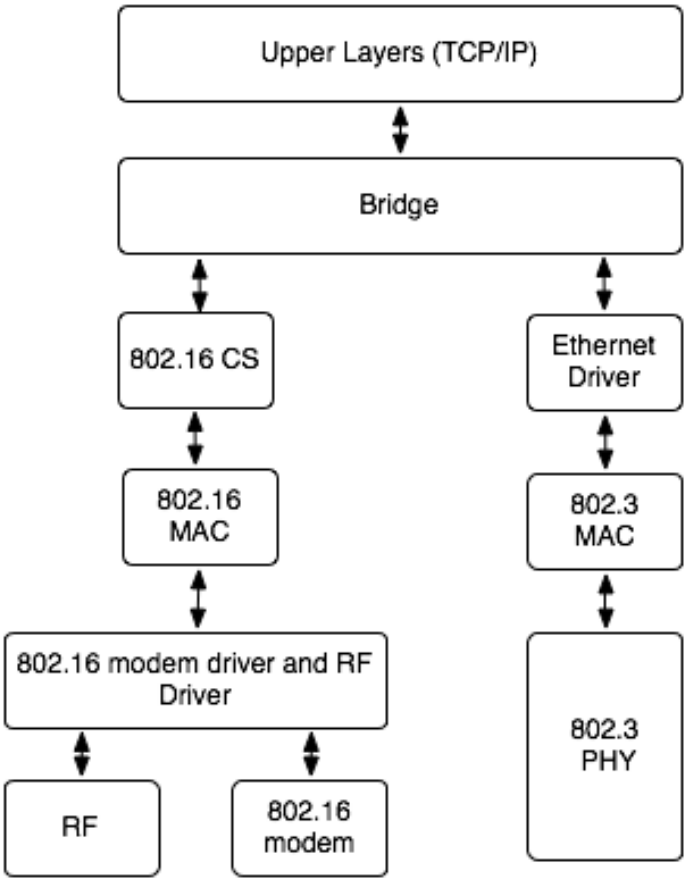

Figure 52: Protocol Stack of the Subscriber Stations

B - Modifications to *QoS em IPv6*

Modification in terms of network protocol

The initial system in [16], in terms of networking, defined the use of IPv6 as the network layer protocol. Because of the device restrictions that are mentioned in Appendix D, the network layer protocol had to be changed to IPv4. What also needed adjustments was the marking values of IP packets. This is due to the difference between the TOS and Traffic Class fields (both web server and router were changed).

Modification in terms of physical topology

Another aspect that needed to be changed was the physical network topology. Initially, the project was designed to work in a controlled environment, away from other network. What the current dissertation proposes is the use of [16] as the core network.

This way, the router needed to connect to another network through an interface and also have the hosts in different network segments. This was accomplished through the use of the IEEE 802.1Q standard. This enabled the physical configuration to change (the router had two network interfaces and now only needs one network interface), maintaining the logical configuration, i.e., the original hosts are still in different network (VLAN 5 and 6) and the access network is coming in the physical interface in trunk mode.

Modification in terms of META-Info definition

The initial project also had to suffer modifications in terms of the definition of the META-INFO files. In the initial specification, these files defined the contents' requirements in terms of QoS. For example, a content that had requirements of Expedited Forwarding behavior, would have a corresponding meta-file with the *String* EF on it.

What is proposed for this Thesis is that the definition of these files includes not only the definition of the Class of Service, but also the Maximum Bandwidth Value. This way, a content with requirements of EF and 8 Mbps of Maximum Bandwidth has associated a meta file like the contents defined in Code Example 2.

Code Example 2 - New format of the meta info file

```
EF
AS:8192
```

The format of the meta info file preserves the old definition (first line), thus giving backward compatibility with the initial project (provided that the implementation of the core network is in IPv6) and

adds the maximum bandwidth. The definition of the bandwidth is the same as used for the bandwidth modifier of the SDP standard [30].

This mechanism provides the necessary information to provide to the *Provision Bandwidth Event* on the WiMAX Broker, which will provision the necessary connection in the WiMAX domain.

In order to communicate with the WiMAX QoS Broker, there had to be made some adjustments in the *qos-control.pl* script.

This script used to analyze the meta-info files, looking for the Class of Service Definition. With the introduction of a new parameter, it had to be extended to support it. Additionally, the Web Server is now also responsible for extracting the following parameters:

- Source IP of requester
- Own IP Address
- Source Port on which traffic will be served

The gathering of these parameters is accomplished through specific Web-Server attributes (except the port, which is well known for each of the traffic classes).

All this information is sent to the Broker to provision a new connection for a specific client. Along with this information, the VLC controls that were initially given to the user had to be enhanced. This was necessary because it is necessary that the end of the transmission is detected. For this purpose, there were introduced *javascript* functions that allow this detection.

The *javascript* functions have embedded the same information that was used for the provisioning process and they are associated to the **stop** button. When this button is pressed, the control not only stops the transmission of data, but also sends a message to the Broker, informing it that the connection was stopped and there is no need to have it provisioned.

The introduction of the *javascript* functions in the Web-Server allow that the control of connection creation and deletion is on the Web-Server side, thus maintaining the weight of QoS-enabled connections control in the core.

Enhancements for an easier installation

Additionally, there were also created .deb packages to automate the installation of the project (these packages can be run in any debian-based system).

C - Communication Diagrams

SS network entry

The SS network entry is the event that handles the network entrance of a client in the WiMAX domain. The messages and modules that are involved in this process are represented in figure 53.

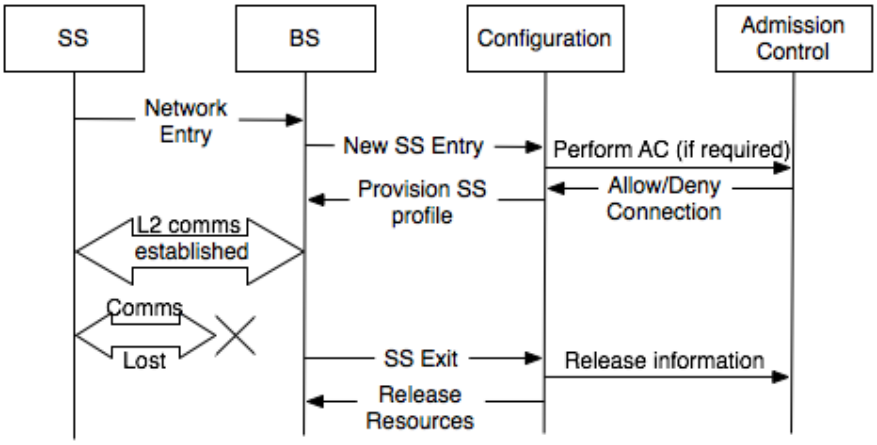

Figure 53: Communication Diagram representing the entry/exit of a WiMAX client

The SS starts its network entrance by exchanging messages with the WiMAX Base Station. These initial messages are part of the 802.16 protocol. When this communication is established, the BS sends a message to the Configuration Module, informing him of the new WiMAX client.

The information in this message is the MAC Address of the SS. The newly entered client is then processed in the Configuration Module and his profile is provisioned in the Base Station. This process will add specific Service Flows to this Subscriber Station, allowing the client to establish L2 communication connections. Remember that the process of establishing L2 communication is essential for the establishment of upper layer protocols. Only after, can the non-management traffic between client and Base Station circulate.

The Configuration Module will keep the configuration untouched on the Base Station until another event occurs. It is also represented the Subscriber Station Exit. This event occurs in two situations: 1) the client is leaving the network or 2) the client is receiving a poor signal from the Base Station and it starts to scan the spectrum looking for a better transmitter.

When any of the previous situations happen, the Base Station notifies the Configuration Module. The Module looks for the provisioning information associated with the Subscriber Station and orders the deletion of all resources associated with it. This includes all Service Flows provisioned (including BE) and associated Packet Classifiers.

Provisioning/Deprovisioning Bandwidth on request

The request for provisioning new resources may happen only through the use of the SIP protocol. Figure 54 shows the entities involved in the process of dynamic bandwidth request, as the messages exchanged.

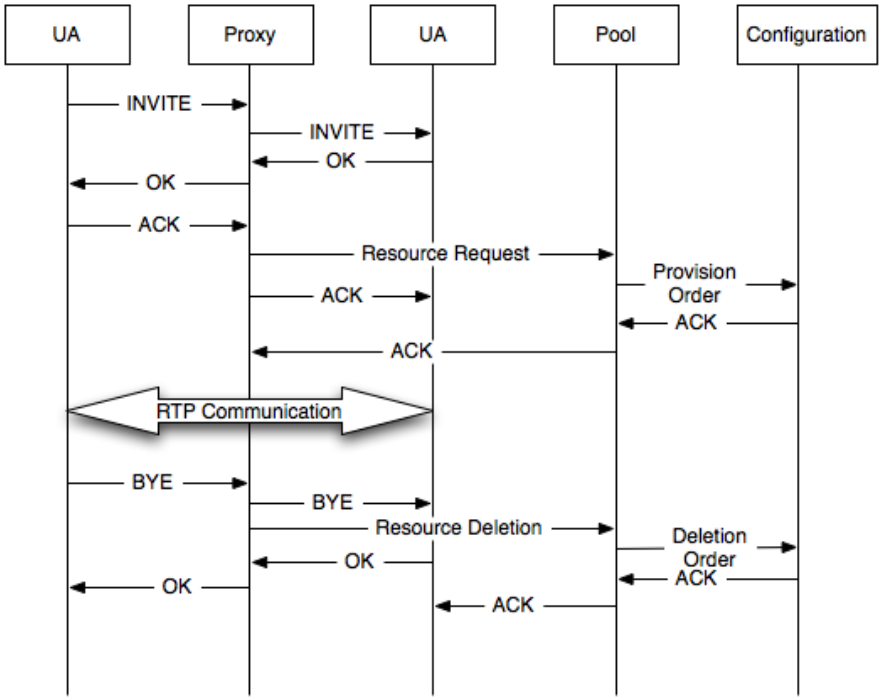

Figure 54: Communication Diagram representing the Provisioning of new resources

Some messages belong to the SIP protocol. This protocol is essential to the provisioning of resources. The messages that do not belong to the SIP protocol are the matter of interest. These messages are the ones exchanged between the Proxy, Pool and Configuration. The first one is the *Resource Request* message. This message is sent by the Proxy to the Pool and is considered the beginning of the Provisioning process. Note that this message is only sent by the Proxy after gathering all the relevant information from the UA's (that's the reason why it is only sent after the ACK message reaches the Proxy).

When the request reaches the Pool, it will process it. After processing, and assuming that there is enough space for resource provisioning in Base Station, the Pool sends a message to the Configuration, ordering the Provisioning of the resources.

When the message is received in Configuration there are two different possibilities (these are not represented in figure 54). The first one is the provisioning of resources in the Base Station, which will happen if the client belongs to the WiMAX domain. The other possibility is that the client does not belong to the WiMAX domain. In this case, a message is generated to the DiffServ QoS Broker, which

will provision the necessary filters for the connection.

After the provisioning of resources in the Base Station, the UAs will have their connections QoS-enabled. This way, the RTP communication will flow through these QoS enabled paths. When one of the UAs decides to leave the conversation, it signals it with a SIP BYE message. Once again, this message is intercepted by the Proxy. Following the interception of the BYE message, the Proxy sends a message to the Pool, informing that the connection is no longer in use. This is where the release of resources begins. The Pool receives this message and processes it, sending it to the Configuration module, for cleaning the previously established QoS flows.

The communication mechanism that was just described shows the signaling that occurs between SIP-enabled applications. Although there was mentioned previously (in section 4.4) that there were applications that needed the addition of a SIP Driver, the communication diagram doesn't suffer changes, as the communication process is the same. The only difference that may arise is in the type of communication that is sent between the clients (in figure 54 is represented the RTP communication, but for the case of legacy applications, other types of communication should be expected).

Resource degradation

The resource degradation is triggered when the Pool has exhausted all the resources and has an incoming request. This request is initiated by the SIP Proxy. Figure 55 shows the entities involved in this process and the messages they exchange.

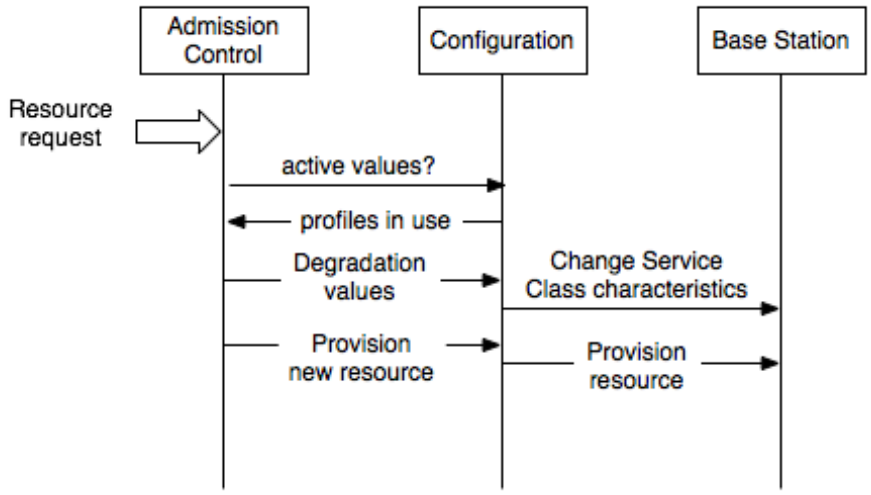

Figure 55: Communication Diagram representing the Resource degradation

As figure 55 shows, the request for new resource provisioning arrives at the Admission control module. If there are no resources available to give, the degradation process starts. The degradation model asks the configuration what are the currently active profiles and processes the values. If there is enough space

to allow the new resource, it will send to the configuration the information that it should change one or more Service Classes in order to allow the new resource. The configuration module then takes action and provisions these new changes in the Base Station.

D - WiMAX devices' limitations

The available hardware for the implementation of the prototype had some limitations in terms of functionality. The most important is the lack of support of IPv6 classification rules. This information can be found in the vendor's NMS documentation - refer to [34], page 283. It stats that the classification rules in IPv6 are not supported in the Subscriber Station, thus the access network must be:

1. IPv4 only with support of classification rules
2. IPv6 ready with no support of classification

The first point allows that traffic in the uplink direction to be classified an inserted into QoS enabled Service Flows (more than one), while the second point implies that there can't be any distinction of the traffic going in the uplink direction.

E - Detailed view of state machine actions

Figure 56 shows a detailed view of the evolution of the system from one level to another ($N_n \rightarrow N_{n+1}$). This means that there are new connections that need to be provisioned and that there is bandwidth available within the class this action refers to.

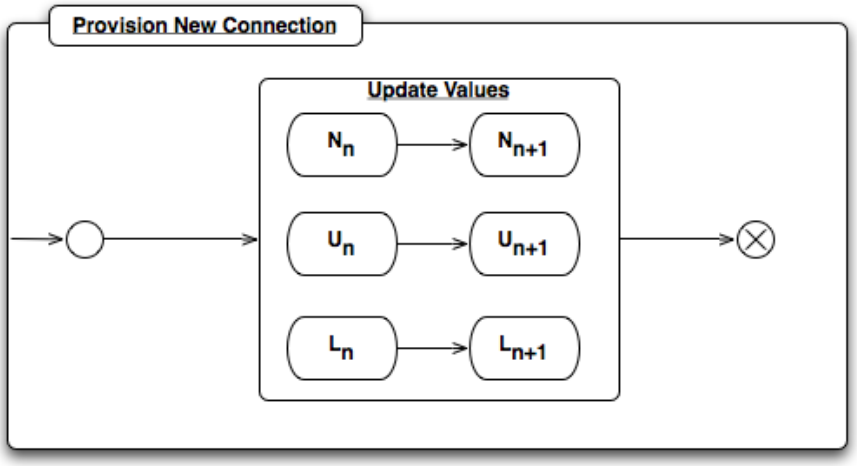

Figure 56: Detailed view for the state machine of the *Provision New Connection* action.

In figure 57 it is possible to see another detailed mechanism: Stealing Resources. To enter this state it is absolutely necessary that there are enough resources available in one of the other classes, to steal. This process is composed by two phases. The first one will update the values on the Stealer class, while the second will update the values on the Stolen class.

This mechanism is useful when at a given time, the number of connections of a given connection type increase and occupy more bandwidth than the other classes. Thus, what this mechanism represents is a transfer of resources from one class to another. This stealing of resources may be temporary or not.

Note that the stealing mechanism will only happen if the equation $B_{available}^{conntype} > B_{Step}^{conntype}$ stands.

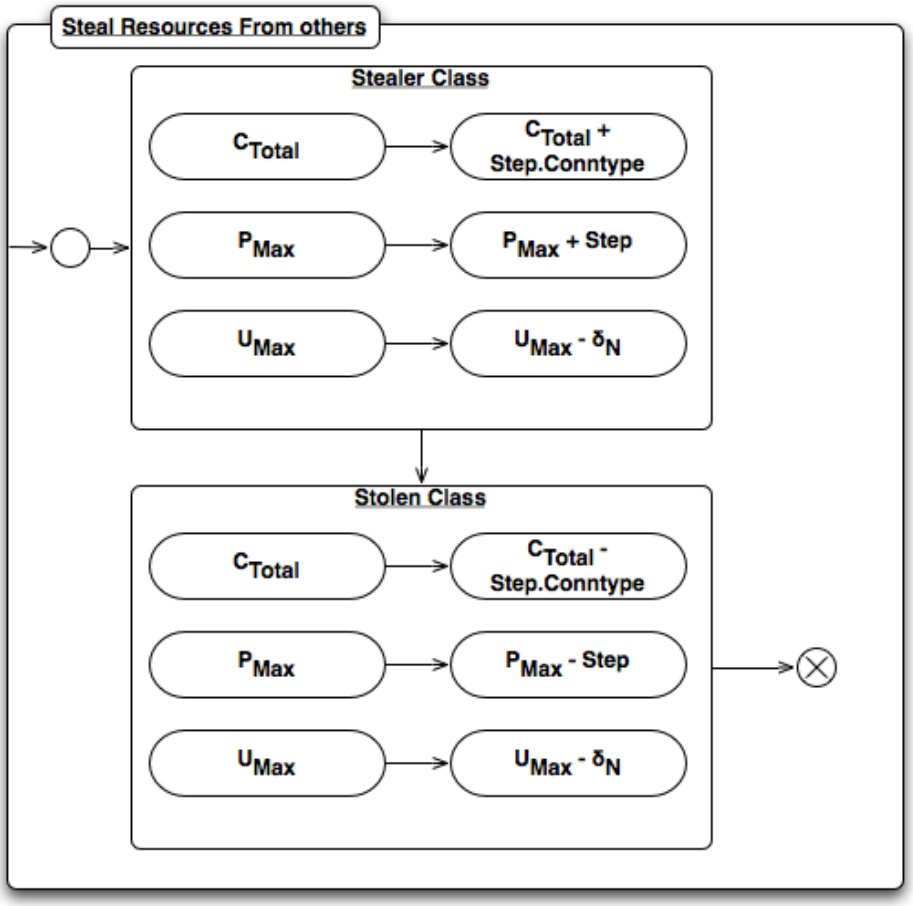

Figure 57: Detailed view for the state machine of the *Steal Resources from others* action.

F - Messages exchanged by the broker

Provision Bandwidth Event:

The messages sent to the Provision bandwidth event have the format depicted in figure 58.

Variable size	4 bytes	4 bytes	Variable size	4 bytes	Variable Size	Variable Size
UserName	SrcIP	DstIP	BW	Port	MediaType	Identifier

Figure 58: Provision Bandwidth Event message

The message format depicted represents the request and the response message. The UserName field corresponds to the user name that is triggering the bandwidth, the IP fields identify source and destination participants, the BW field identifies the value that is required for this session, mediaType identifies the type of traffic that is going to be sent (e.g. audio, video).

When the message depicted is a request, the value of the identifier field should assume the value "0". If a successful provisioning happens, the value returned in the identifier field is different from "0" and assumes the value of the pair (Service Flow ID, Classifier Rule Index).

Note that this message format is also used for the *Provision DiffServ Filter Event*. However, the fields Username and Identifier are ignored.

Degradation Event:

The degradation event messages are depicted in figures 59 and 60. To trigger this event, the broker

4 bytes	4 bytes
SC_ID_1	NrClasses
...	...
SC_ID_n	NrClasses

Figure 59: Degradation Event Message (response from broker)

4 bytes	8 bytes
SC_ID_1	NewMaxSR
...	...
SC_ID_n	NewMaxSR

Figure 60: Degradation Event Message (order from admission control)

first receives a request with a String code. The response to this request is represented in figure 59 and corresponds to the values of currently active connections associated to a specific Service class ID. This ID represents the profile identifier.

After computing the values for degradation, the Admission Control function sends a new message to the broker informing him of the new Maximum Sustained Rate values. The message is represented in

figure 60 and contains the Service Class Identifier (profile) and the new values for maximum sustained rate.

New SS Event:

For configuration of a Subscriber, the message format depicted in figure 61 is used.

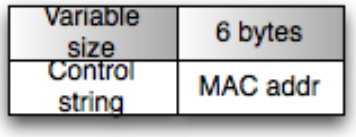

Figure 61: Provision Bandwidth Event message

The control string gives the information if the the SS Event is for the action of entry or exit of the network. Along with the control sequence, the MAC Address is also in this message and identifies the mac address of the Subscriber Station.

References

[1] IEEE 802.11 Working Group, *Wireless LAN Medium Access Control (MAC) and Physical (PHY) Layers*, January 2003

[2] Hancock, R.; Karagiannis, G.; Loughney, J.; Van den Bosch, S.; *Next Steps in Signaling (NSIS): Framework, RFC 4080*, June 2005

[3] WiMAX Forum, *WiMAX End-to-End Network Systems Architecture Stage 2-3 Release 1.1.0.* http://www.wimaxforum.org/technology/documents/ , July 2007

[4] ITU-T Recommendation E.800: *Terms and definitions related to quality of service and networks performance including dependability*, August 1994

[5] ITU-R Recommendation BT.500-11: *Methodology for the subjective assessment of the quality of television pictures*, June 2002

[6] ITU-T Recommendation P.800: *Methods for objective and subjective assessment of quality*, 1996

[7] ITU-T Recommendation P.862: *Perceptual evaluation of speech quality(PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*, February 2001

[8] Braden, R.; Clark,D.. *RFC 1663: Integrated Services in the Internet Architecture: an Overview*, July 1994

[9] Shenker, S.; Partridge, C.; Guerin,R.. *RFC 2212: Specification of Guaranteed Quality of Service*, September 1997

[10] Shenker, S.; Wroclawski, J.. *RFC 2215: General Characterization Parameters for Integrated Service Network Elements*, September 1997

[11] Nichols, K.; Blake, S.; Baker, F.; Black, D.. *RFC 2474: Definition of the Differentiated Services Field (DS Field)*, December 1998

[12] Blake, S.; Black, D.; Carlston, M.; Davies, E.; Wang, Z.; Weiss,W.. *RFC 2475: An architecture for Differentiated Services*, December 1998

[13] Wang, Zeng *Internet QoS - Architectures and Mechanisms for Quality of Service*, Morgan Kaufman Publishers, 2001

[14] IEEE 802.16 working group. *IEEE Std 802.16-2004 (Revision of IEEE Std 802.16-2001): IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems.* IEEE Press, 2004

[15] Almsberg,Wener; Le Boudec, Jean-Yves; Oechlin, Philippe. *Application Request IP over ATM (AREQUIPA) and its Use in the Web*, Global Information Infrastructure (GII) Evolution, pp. 252-260, IOS Press, 1996.

[16] Correira, Pedro; Silva, Ricardo : *Relatório final de TFC - QoS em IPv6*, IST, 2006

[17] Veltri, Luca; Salsano, Stefano; Papalilo, Donald: *QoS Support for SIP Based Applications in a Diffserv Networks*, October 2002

[18] IEEE 802.16 working group. *IEEE Std 802.16e-2005 and IEEE Std 802.16-2004/Cor 1-2005 (Ammendment and Corrigendum to IEEE 802.16-2004): IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems. Ammendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Lecensed Bands and Corrigendum 1.* IEEE, 2005

[19] Wang, H; Li, Wei; Agrawal, Dharma P. : *Dynamic Admission Control and QoS for 802.16 Wireless MAN*, IEEE 2005.

[20] Gakhar, Kamal; Achir, Mounir, Gravey, Annie: *Dynamic Resource Reservation in IEEE 802.16 Broadband Wireless Networks*, IEEE 2006.

[21] Chen, Jianfeng; Jiao, Wenhua; Guo, Qian: *Providing Integrated QoS Control for IEEE 802.16 Broadband Wireless Systems*, Vehicular Technology Conference, 2005. VTC-2005-Fall. 2005 IEEE 62nd.

[22] Mai, Yi-Ting; Yang, Chuan-Chuan; Lin, Yu-Hsuan: *Cross-Layer QoS Framework in the IEEE 802.16 Network*, ICACT 2007.

[23] Delicado, J; Orozco-Barbosa, L.; Delicado, F.; Cuenca, P.: *A QoS-aware protocol architecture for WiMAX*, IEEE CCECE/CCGEI 2006

[24] Angori, Enrico; Borcoci, Eugen; Mignanti, Silvano; Nardini, Cristina; Landi, Giada; Ciulli, Nicola; Sergio, Giacomo; Neves, Pedro. *Extending WiMAX technology to support End to End QoS guarantees*, May 2007

[25] Bohnert, Thomas Michael; Castrucci, Marco; Ciulli, Nicola; Landi, Giada; Marchetti, Ilaria; Nardini, Cristina. *Architectural Solution for QoS Management in a WiMAX Network*, BWCCA 2007

[26] http://www.euqos.eu/, online January 2008

[27] Howarth, M.P; Flegkas, P.; Pavlou, G; Wang, N.; Trimintzios, P; Griffin, D.; Griem, J.; Boucadair, M.; Morand, P.; Asgari, A.; Georgatsos, P.. *Provisioning for Interdomain Quality of Service: the MESCAL approach*, IEEE Communications Magazine, June 2005, pp. 129-137.

[28] Mykoniati, E.; Charalampous, C.; Georgatsos, P.; Damilatis, T.; Godersi, D.; Trimintzious, P.; Pavlou, G.; Griffin, D.; *Admission Control for Providing QoS in DiffServ IP networks: the TEQUILA Approach*, IEEE Communication Magazine, January 2003, pp. 38-44.

[29] Rosenberg, J.; Schulzrinne, H.; Camarillo, G.; Johnston, A.; Peterson, J.; Sparks, R; Handley, M.; Schooler, E.; *RFC 3261 - SIP: Session Initiation Protocol*, June 2002

[30] Handley, M; Jacobson, V.; Perkins, C.; *RFC 4566 - SDP: Session Description Protocol*, July 2006

[31] IEEE 802.16 working group.; *IEEE Standard for Local and metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems - Ammendment 1: Management Information Base.* IEEE Press, December 2005

[32] Fock, F.; Katz, J.; *SNMP4J - The Object Oriented SNMP API for Java Managers and Agents.* - Web-site: http://snmp4j.org/index.html, visited August 2008

[33] Net-SNMP Web-site: http://net-snmp.sourceforge.net/, visited August 2008

[34] Airspan Networks Inc.; *Netspan SR6.0/6.1 Users Guide - version 605-0000-845 Rev C*, September 2007

[35] Tsrtsis, G.; Srisuresh,P; *Network Address Translation - Protocol Translation (NAT-PT)*, February 2000