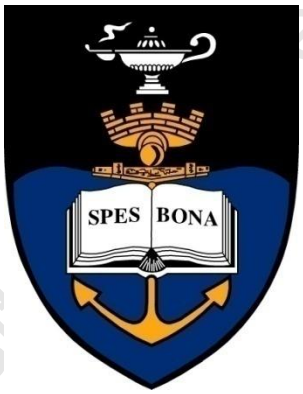
Implementation and Performance Evaluation of an NGN prototype using WiMax as an Access Technology

Bessie Malila



This thesis is submitted in partial fulfillment of the academic requirements
for the degree of
Master of Science in Electrical Engineering
in the Faculty of Engineering and The Built Environment
University of Cape Town
December 2012

As the candidate's supervisor, I have approved this dissertation for submission.

Name: Neco Ventura

Signed: _____

Date: _____

Declaration

I hereby declare that: (1) the above thesis is my own unaided work, both in conception and execution, and that apart from the normal guidance of my supervisor, I have received no assistance apart from that stated below; (2) neither the substance nor any part of the thesis has been submitted in the past, or is being, or is to be submitted for a degree in the University or any other University.

I am now presenting the thesis for examination for the Degree of MSc in Electrical Engineering. I also grant the University free license to reproduce the above thesis in whole or in part, for the purpose of research.

_____ _____

Bessie Malila Date

Abstract

Telecommunications networks have evolved to IP-based networks, commonly known as Next Generation Networks (NGN). The biggest challenge in providing high quality real-time multimedia applications is achieving a Quality of Service (QoS) consistent with user expectations. One of the key additional factors affecting QoS is the existence of different QoS mechanisms on the heterogeneous technologies used on NGN platforms. This research investigates the techniques used to achieve consistent QoS on network technologies that use different QoS techniques.

Numerous proposals for solving the end-to-end QoS problem in IP networks have adopted policy-based management, use of signalling protocols for communicating applications QoS requirements across different Network Elements and QoS provisioning in Network Elements. Such solutions are dependent on the use of traffic classification and knowledge of the QoS requirements of applications and services on the networks. This research identifies the practical difficulties involved in meeting the QoS requirements of network traffic between WiMax and an IP core network. In the work, a solution based on the concept of class-of-service mapping is proposed. In the proposed solution, QoS is implemented on the two networks and the concept of class-of-service mapping is used to integrate the two QoS systems. This essentially provides consistent QoS to applications as they traverse the two network domains and hence meet end-user QoS expectations. The work is evaluated through a NGN prototype to determine the capabilities of the networks to deliver real-time media that meets user expectations.

Acknowledgements

I would like to thank the following people for their assistance during the course of this project:

Mr. Neco Ventura, for his supervision and guidance throughout this project.

Mr. Vitalis Ozianyi, for his technical assistance and invaluable criticism.

The past and present members of the Communications Research Group at UCT, for their advice and feedback.

Table of Contents

List of Tables

List of Figures

Abbreviations

AF	Assured Forwarding
API	Application Programming Interface
APS	Application Service Provider
AS	Application Server
ASN	Access Service Network
BB	Bandwidth Broker
BE	Best Effort
BGF	Border Gateway Function
BGP	Border Gateway Protocol
BS	Base Station
CG	Continuous Grant
COPS	Common Open Policy Service
CPE	Customer Premise Equipment
CSN	Connectivity Service Network
DSCP	Diffserv Code Point
DSL	Digital Subscriber Line
EF	Expedited Forwarding
FDD	Frequency Division Duplex
FEC	Forward Error Correction
GMPLS	Generalized Multiprotocol Label Switching
HD-IPTV	High Definition IP Television
HSS	Home Subscriber Station
I-CSCF	Interrogating Call Session Control Function
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPTV	IP Television
ITU-T	International Telecommunications Union – Telecommunications Section
LTE	Long Term Evolution
MAC	Media Access Control

MPLS	Multiprotocol Label Switching
NACF	Network Access Control Function
NAP	Network Access Provider
NAT	Network Address Translation
NE	Network Element
NGN	Next Generation Network
nrtPS	non real time Polling Service
OFDM	Orthogonal Frequency Division Multiplexing
OMA	Open Mobile Alliance
OSPF	Open Shortest Path First
P-CSCF	Proxy Call Service Call Function
PDFE	Policy Division Functional Entity
PDU	Protocol Data Unit
PHB	Per Hop Behavior
PSTN	Public Switched Telephone Network
QAM	Quadrature Amplitude Modulation
QoE	Quality of Experience
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RACF	Resource Admission Control Function
RIP	Routing Information Protocol
RSVP	Resource Reservation Protocol
rtPS	Real Time polling Service
S-CSCF	Serving Call Session Control Function
SD-IPTV	Standard Definition IPTV
SDP	Session Description Protocol
SFID	Service Flow Identifier
SFM	Service Flow Manager
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SS	Subscriber Station

SU	Subscriber Unit
Tcng	Traffic Control Next Generation
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TDD	Time Division Duplex
TRC-FE	Transmission Resource Control Functional Entity
UDP	User Datagram Protocol
UE	User Equipment
UGS	Unsolicited Grant Service
UMTS	Universal Mobile Telecommunications System
VoD	Video on Demand
VoIP	Voice Over Internet Protocol
WiFi	Wireless Fidelity
WiMax	Worldwide Microwave Access
WLAN	Wireless Local Area Network

Glossary

3G: This is the third generation of mobile network technologies under the ITU IMT 2000 technologies, which allow higher bandwidth and support more solutions for transmitting data over wire networks.

Delay: This is a performance characteristic of a telecommunication network which specifies how long it takes a bit of data to travel across the network from one network node or end point to another. Customers are usually concerned about the total delay of a network but engineers usually specify maximum and average delay for a network.

IEEE 802.16: An IEEE telecommunications standard, which specifies an air interface of fixed broadband wireless access systems that support multimedia services.

IPTV: Internet Protocol Television is a television service transmitted over IP networks and is one of the multimedia services enabled by the Internet.

ITU-T: The International Telecommunications Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication standardisation sector (ITU-T) is a permanent organ of the ITU responsible for studying technical, tariffs and operating questions and issuing recommendations, with the aim of standardizing telecommunications internationally. QoS is one of the key topics under study by the body.

Jitter: Network jitter refers to packet delay variability over time across a telecommunication network. The value is expressed as the average deviation from the mean network delay. Also known as packet delay variation (PDV), jitter is an important factor in assessing the performance of IP networks.

LTE: A Long Term Evolution is a new cellular radio standard, which allows faster, more efficient transfer of data and enables the next generation of mobile data services.

MPLS: Multi-protocol Label Switching MPLS is a telecommunications technology used in NGN core networks including converged data and voice networks. MPLS works alongside existing and future routing technologies to provide high-speed data forwarding between Label Switched Routers and offer bandwidth reservation for traffic flows with different QoS requirements.

Network bandwidth: Network bandwidth refers to the data rate supported by a network connection or interface. Bandwidth represents the capacity of the connection. The maximum throughput of a channel therefore represents the bandwidth of the channel. Bandwidth tests provide the maximum throughput of a channel.

NGN: Next Generation Networks

Packet loss: Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Causes of packet loss include signal degradation over the transmission medium due to multi-path fading, packet drop because of channel congestion, corrupted packets rejected in-transit, faulty networking hardware, faulty network drivers or normal routing routines.

QoS: Quality of Service, in packet switched telecommunication networks, refers to resource reservation control mechanisms. The term refers also to the ability of a network to provide different priority to different applications, users, or data flows; or the ability to guarantee a certain level of performance to data flows. QoS metrics include throughput, delay, and jitter. Packet loss is the packet dropping probability.

QoE: Quality of Experience is a subjective measure of a customer's experiences with a service for example web browsing, phone call, TV broadcast or call to a Call Center). QoE looks at a service provider’s offering from the customer’s point of view. This different from QoS which objectively measures the service provided by the vendor and is mostly directed towards the media and not the customer, as a customer would not publicly mention that the jitter is too high or the delay is too long.

Throughput: In communication networks, network throughput is the average rate of successful data transfer through a communication path. Delivery of the data or message is

over a physical or logical link; or through a network node.

UMTS: Universal Mobile Telecommunications Service is a 3G standard, which supports data transmission rates of up to 2Mbps.

VoIP: Voice over Internet Protocol refers to the two-way transmission of voice messages over a packet-switched network.

WiMax: Worldwide Interoperability Microwave access is the name created by the WiMax Forum to describe a telecommunication protocol that provides fixed and fully mobile Internet access at up to 1Gbits/s speed. The WiMax Forum promotes conformity to the standard and interoperability of equipment from different manufacturers.

Chapter 1

1.0 Evolution of Telecommunications Networks

Traditional telecommunications networks were designed to carry different services on different technology platforms. This resulted in separate networks for telephony, telegraphy, fax and data services. As Internet and cellular network technologies emerged, network operators continued to build networks that operated independent of each other as well as parallel to existing communications infrastructures. Due to technological advancements, telecommunications networks have evolved into one transport network that is able to carry integrated services and applications, commonly known as the All-IP network or Next Generation Network (NGN).

According to the International Telecommunications Union – Telecommunications standardization sector (ITU-T) definition, “A Next Generation Network is a packet-based network able to provide services including Telecommunication Services and able to make use of multiple broadband, Quality of Service-enabled transport technologies in which service-related functions are independent from underlying transport-related technologies. It offers unrestricted access by users to different service providers. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users”, [1]. The NGN architecture therefore decouples the traditional telecommunications networks into access, core, control and application/services layers. This allows network operators to deploy the best technologies at each layer, thus unlocking the flexibility to choose more than one vendor for network deployment.

Figure 1 illustrates the evolution of the communications networks from traditional independent infrastructures to an IP-based infrastructure. NGN platforms enable end users to access services using a single device, compared to traditional networks where multiple devices are required to access services on the different networks.

The challenges to NGN implementation include ensuring interoperability of network equipment and internetworking between different network technologies and different operator domains. One of the key challenges faced by network operators is the ability of the converged heterogeneous networks to provide end-to-end Quality of Service (QoS)

guarantees to real-time multimedia applications that meet end-user expectations.

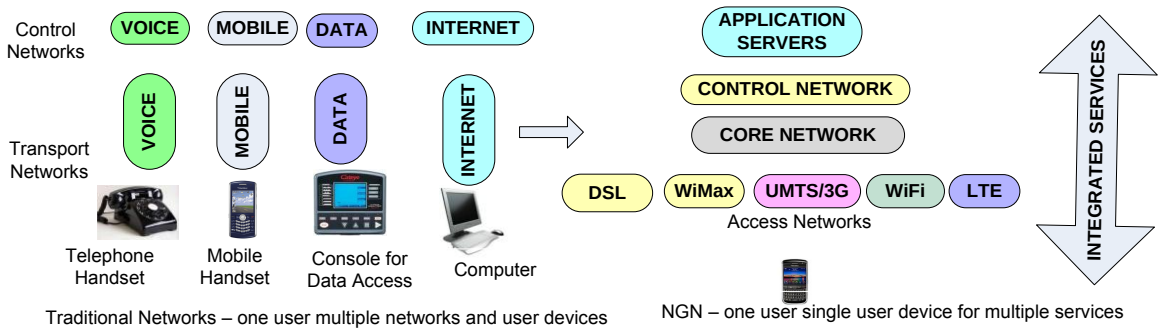


Figure 1: Evolution of Telecommunications networks

1.1 QoS in IP Networks

The Internet Protocol (IP), on which the NGN is based, uses packet switching technology to transfer information from one network node to the other. Consequently, NGN have inherited the QoS problems inherent in packet-switched networks. As the packets are transferred from one node to the other, they can be subjected to delay or they can be dropped if congestion is detected. Jitter or packet delay variation occurs when packets carrying the same type of information are subjected to different delay times on the IP network. In real-time applications like video and voice, large values of jitter degrade the user's perceived quality of the application, i.e. user quality-of-experience. Another problem in packet switched networks is the allocation of bandwidth to applications having different throughput requirements. These four metrics, delay, jitter, packet loss and throughput define a packet-based network's ability to support the transfer characteristics and requirements of real-time multimedia applications, i.e. the QoS provided by the network.

IP-based networks were originally designed to carry best effort traffic, e.g. email and web browsing. In addition to this, today's IP networks must also carry real-time audio, video and other multimedia traffic. These applications are sensitive to delay, jitter, packet loss and present different bandwidth requirements. The ITU-T defines the performance metrics that an NGN must have in order to meet the QoS requirements of applications and services. Specifications for NGN also require that these networks be QoS-enabled and be able to provide integrated voice, video and data services.

Network technologies that meet the ITU-T QoS specifications are now available. Figure 2 shows the four-layered hierarchy of the NGN. QoS-enabled wireless access technologies

include Worldwide Interoperability for Microwave Access (WiMax) [2], Wireless Fidelity (Wi-Fi) and Long Term Evolution (LTE), which evolved from 3G networks. Digital Subscriber Line (DSL) and fibre optic cable dominate wired networks and provide high capacity access links. In the core of the network, leading technologies are Internet Protocol (IP) and Multiprotocol Label Switching (MPLS). The research study in this thesis focuses on WiMax and IP networks.

Control layer technologies include the IP Multimedia System (IMS) and the Soft-Switch. The application or services layer is composed of content servers and an Application Programming Interfaces (API). The API provides an interface for independent application developers to develop additional content for network operators. Although the IMS and application servers are not part of this research study, they are discussed to provide a complete picture of a practical NGN platform. End users are expected to access voice, video and data services using desktop computers, mobile phones or laptops seamlessly across the different access networks. Network operators must satisfy end use Quality of Experience (QoE).

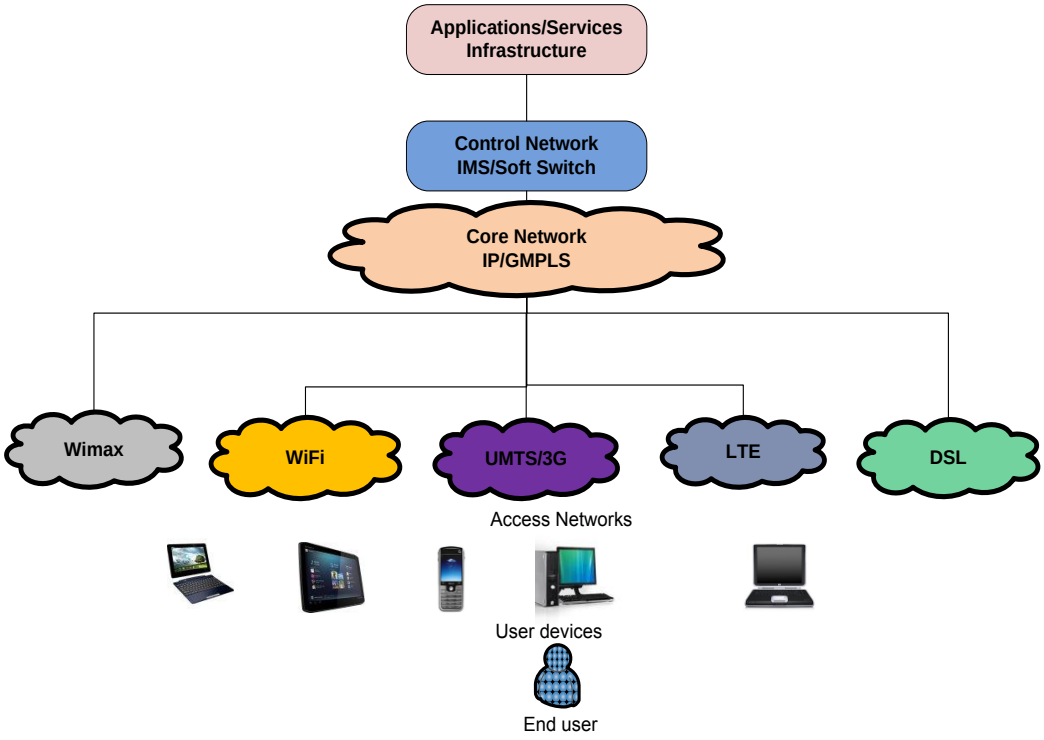


Figure 2: Next Generation Network Hierarchical Network Components

Efforts to address QoS issues in NGN have targeted individual access and core network technologies, resulting in different QoS systems for each of the technologies. Since end users are expected to access services and applications ubiquitously, inter-networking technologies with different QoS systems results in inconsistent QoS handling of traffic between the networks. Consistent QoS is therefore achieved when traffic flows are subjected to the same QoS treatment from source to destination irrespective of technologies or network domains traversed. This thesis focuses on achieving consistent QoS between a WiMax access network and an IP core network.

1.2 Problem Description

NGN are expected to carry real-time multimedia applications that impose stringent QoS requirements on the networks. Since NGN infrastructures are made up of QoS-enabled heterogeneous technologies, more than one network technology is involved in the end-to-end delivery of services and applications. Each technology has a QoS system, which uniquely handles the QoS requirements of services and applications on the network. While the IP technology provides a unified transport network for voice, video, data and other multimedia applications, integrating different QoS solutions is still a challenge. Successful delivery of a service from source to destination therefore requires that all networks involved meet the minimum QoS requirements of the service.

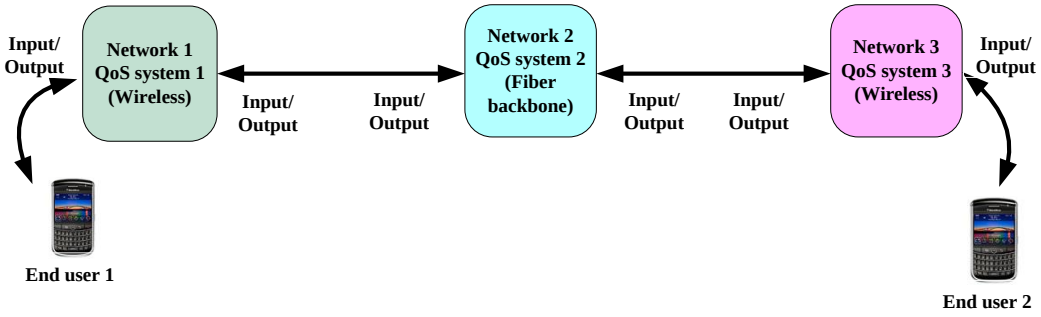


Figure 3: Network Domains with different QoS systems

Consider the communication system shown in figure 3 where real-time applications are transferred between two users using three different networks. Each network has its own

QoS architecture, which ensures packets are treated differently. QoS guarantees on each network are characterized by latency, packet loss, throughput and jitter. A number of practical difficulties arise:

- Each network has an admission control system, which controls the traffic admitted into the network, depending on the resources available in the network. Suppose network 1 is able to admit 1000 traffic flows without compromising the QoS of flows already in the network. Since the networks are interconnected, this traffic is pushed onto network 2. If network 2 cannot accept up to 1000 traffic flows without compromising the QoS of flows already on the network, the excess traffic is discarded at the entry point, resulting in packet loss on the network. If the admission control systems work independently, consistent QoS cannot therefore be achieved on the network.
- Scheduling in IP networks determines which packets are transmitted first on the output link. The scheduler in network 1 may send packets in a particular order depending on the delay, jitter and packet loss characteristics defined on the network for each application. Without knowledge of this schedule, the scheduler on network 2 may follow a different pattern depending on the QoS definitions on the network. The QoS characteristics of the applications are therefore lost along the communications channel.
- IP networks use buffers to store received packets waiting to be sent onto the transmission link or waiting to be processed within the system. When these buffers are full, the QoS systems must decide which packets to drop first, depending on the QoS configuration on the network. If the drop sequence is not consistent on all the networks, the packet loss rate is therefore not controlled; hence, this QoS characteristic is affected.

The need to have an integrated QoS system that addresses the end-to-end QoS requirements of services and applications in IP networks is therefore justified with the following advantages:

- With no proper QoS handling from end-user to end-user, network operators have resorted to providing bit-pipes whose performance is unpredictable when carrying

integrated services. In such networks, real-time applications like VoIP and IPTV starve other applications of limited network resources.

- The ITU-T specifies the QoS requirements for traffic in IP networks [3]. Enforcement has however, been largely left to vendors and network operators. In most cases, network operators have failed to integrate the QoS systems implemented in the different technologies, making it difficult for them to charge for services whose QoS requirements they cannot guarantee.
- In NGN, customers are expected to roam seamlessly from one access network to the other. To support the QoS requirements of applications, consistent QoS across the different networks makes it possible for network operators to meet user expectations.

The practical difficulties of end-to-end QoS management and the benefits of a consistent QoS system in heterogeneous NGN technologies suggest that a consistent QoS system is required.

1.3 Thesis Objectives

In IP networks, one of the biggest challenges faced by network operators is guaranteeing QoS for real-time media consistent with user expectations. In NGN, the principal additional factor is the heterogeneity of the technologies involved in the end-to-end delivery of media. These technologies implement different QoS techniques. This research study investigates the techniques used in NGN platforms to improve the end-to-end delivery of real time media consistent with user expectations, and provide proactively network QoS management to achieve this. In the literature, a number of end-to-end QoS solutions are presented. These solutions aim to meet end user QoS expectations by providing consistent QoS to real-time applications as they traverse different technologies and network domains.

- This research study aims to examine these solutions and uncover particular mechanisms required to achieve consistent QoS across technologies that implement different QoS systems. Equipment manufacturers have developed technologies like WiMax, MPLS and LTE that can deliver real-time media to user

satisfaction. These technologies however, use different QoS techniques to meet user expectations. In NGN, end-to-end delivery of a service involves more than one technology. While proposals in the literature include policy-based QoS management and QoS signalling, to date, there is no standard solution that ensures that the QoS requirements defined in one network or technology domain are achieved in a different or adjacent domain.

- This thesis investigates the key issues surrounding end-to-end QoS management in NGN platforms, specifically and how this is achieved between a WiMax network and an IP core network. A NGN prototype is implemented as proof-of-concept. A class of service mapping strategy for achieving end-to-end QoS control in NGN platforms, specifically WiMax access technology to IP core technology is proposed. The proposed mapping takes into consideration the integration of wireless and wired technologies in NGN architectures. The prototype implementation also provides a platform for the evaluation and proof-of-concept test for future research work related to NGN signalling protocols, QoS control mechanisms, security and multimedia applications.

The prototype network is used to evaluate the performance of converged broadband QoS-enabled NGN technologies. Link quality tests are carried out on the transport networks to evaluate the performance of the network and to ascertain the network's conformance to ITU-T network performance guidelines as outlined in [3]. These tests are used to obtain values of throughput, jitter, delay and packet loss; QoS metrics used to evaluate the performance of IP networks. Tests are also carried out on the prototype network to evaluate the network's ability to carry integrated real-time traffic, which is characteristic of NGN applications. In this thesis, three applications are used in the evaluation tests, i.e. video streaming, IPTV and data. The results obtained provide insight into the behaviour of real networks. Experience is also gained in implementation of NGN architectures.

The prototype network is implemented using open source software on the core, control and applications/services layers. This demonstrates the ability to use open source software in the implementation of low cost, broadband and QoS-enabled networks that can be used to provide real-time multimedia services. Performance measurement tests are also carried out using open source software. Research test beds based on open

source software can easily be adopted in developing countries with limited financial resources for research. This will go a long way in solving some of the challenges faced by developing countries as they the embrace new low-cost technologies.

1.4 Scope and Limitations

A number of limitations have been set to reduce the scope of the study. NGN platforms comprise several access network technologies, an IP or MPLS core, an IMS or Soft-Switch control network and content servers. This work involves the integration of WiMax, IP-router core network and the IMS. Other technologies are therefore beyond the scope of this thesis.

Work on the WiMax network is limited to the installation of a WiMax system consisting two subscriber stations and a micro base station. The network's performance is evaluated based on its conformance to ITU-T QoS specifications. Figure 4 shows the WiMax network architecture.

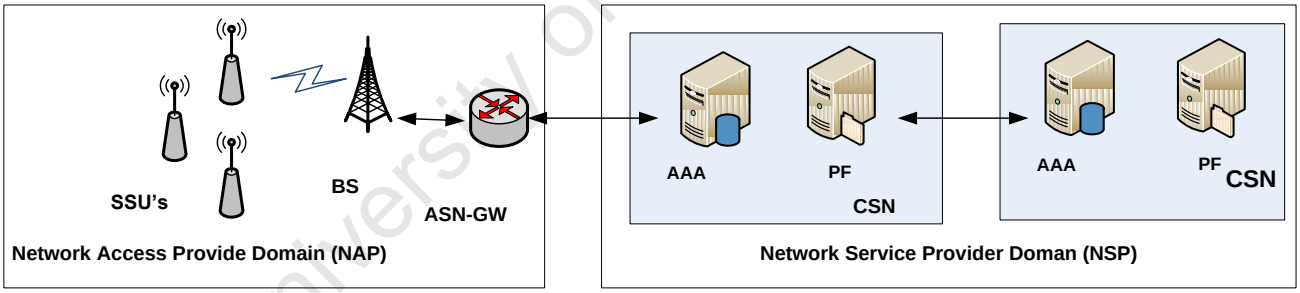


Figure 4**:** Point-to-multipoint WiMax network architecture

Notes:
BS – base station ASN-GW: Access Service Network-Gateway CSN: Connectivity Service Network
PF: Policy Function Server AAA: Authentication Authorization Accounting Server

The Access Service Network (ASN) consists of the radio Network Elements and the gateway node, which interconnects to the core network, and is owned by a single network operator, the Network Access Provider (NAP). The Connectivity Service Network (CSN) represents functions that provide IP connectivity services to Wimax subscribers and is owned by a Network Service Provider (NSP). The system used in this thesis consists of

the ASN only and obtains these services from the IP core network and the IMS network.

Although the network used in the experiments is the fixed access standard – IEEE 802.16d, the proposed QoS traffic classification also applies to the mobile standard i.e. IEEE 802.16e.

The WiMax technology implements QoS control both on the physical layer and on the MAC layer. The system implemented dynamically controls physical layer QoS. On the MAC layer, the system provides an option for the network operator to configure traffic classification on the micro-base station through a network management system. Three options are available for traffic classification, the transparent mode, 802.1p mode and the DiffServ Code Point (DSCP) mode. This research work is limited to the use of the DSCP option for traffic classification. The network is inter-connected to the core transport network via a fast Ethernet data interface on the micro base-station, and to the end-user equipment via an Ethernet interface on the subscriber station indoor unit.

The use of the IMS is limited to QoS signalling between application servers that interact with the IMS and the end user computers that connect onto the platform via the WiMax subscriber stations. In terms of QoS capabilities, the clients are available in two types [4]. Type 1, clients are clients with no QoS negotiation capabilities. These clients are used for generating data and Internet traffic. Type 2 clients can negotiate QoS parameters e.g. bandwidth through service signalling but are unaware of the QoS capabilities of the transport layer [4]. They use the Session Initiation Protocol (SIP) with Session Description Protocol (SDP) for QoS negotiation. Type 2 clients are used for evaluating multimedia applications that have stringent QoS requirements. The core and access networks therefore provide a path for call session set up and control signalling between type 2 clients, running as IMS clients and the application servers. The UCT IMS client is used for type 2 clients and this is the type used in the thesis.

On the core network, a number of open source modules are installed on the Linux machines to enable routing and QoS functionalities. The Quagga [5] open source routing software is used with only the Zebra kernel routing manager package and Routing Information Protocol (RIP) daemons activated. For QoS control, the DiffServ module is used for implementation of traffic classification. The Traffic Control Next Generation (tcng)

module is installed to enable classification of next generation network traffic. The Iptables module is installed to achieve definition of packet filtering rules. Security implementation using Iptables is beyond the scope of this thesis.

While traffic on NGN is expected to be video, voice, data, IPTV, gaming and other multimedia applications, the performance of the converged network is evaluated based on its capability to meet the QoS requirements of voice, video and data applications only. The DiffServ QoS model and the tcng modules support the implementation of sub-classes of QoS for example pure voice traffic and voice with silence suppression; video streaming and video conferencing; or various traffic classes of data like transactional services, email, web browsing or other data applications. In this thesis, only one type service is implemented for each of the three types of service. These are pure voice, video streaming and data traffic respectively.

Link quality tests are performed to evaluate the conformance of the network to ITU-T performance guidelines [1]. The results are expected to exceed expectations since tests are done in an indoor environment with no effects from the outside environment and no transmission delay due to distance. The tests will therefore evaluate the performance of the WiMax network in terms of its ability to adapt to suitable modulation and coding techniques and the routing functionalities of the open source routing software. Iperf, an open source software application is used for the evaluation of the network's performance. The Alvaricraft Network Management System, which is proprietary to the WiMax network manufacturer, is used to monitor and configure the WiMax network.

1.5 Thesis Outline

The remainder of this thesis is structured as follows:

Chapter 2 introduces some basic and key QoS concepts and issues in IP networks. It then reviews the QoS issues related to end-to-end QoS management in converged NGN architectures. A review of QoS in WiMax networks and IP routers is given. This is followed by a review of literature related to QoS mechanisms for NGN where WiMax is used as an access or backhaul network for other access technologies and IP core networks. The concept of Class of service mapping is presented as way of translating QoS parameters between WiMax and IP QoS models using the DiffServ QoS model as the common

element for integrating the two QoS systems.

Chapter 3 is separated into two main sections. Firstly, the concept of class of service mapping is introduced. Details of how the concept is used for translation of QoS parameters from one network domain to the other are given. The reasons for the use of this approach are also discussed. The second part of the chapter explains the actual QoS translation procedure between the WiMax network and the IP core routers.

Chapter 4 describes the evaluation framework used in this research. The design of the NGN prototype and QoS details needed to evaluate the proposed end-to-end QoS solution in a real-world network is described. It details how the different QoS systems described in chapter 3 are implemented.

Chapter 5 describes the tests performed for the performance evaluation of the NGN prototype based on the four key QoS metrics for IP networks, i.e. delay, jitter, packet loss and throughput. The first tests, link quality tests, ensure network conformance to ITU-T standards before traffic is loaded. The results are compared against those defined by the ITU-T. The second tests are carried out to evaluate the ability to carry real-time multimedia traffic in the form of data, video and IPTV video on demand.

Chapter 6 presents a set of conclusions drawn up from the evaluations. The chapter also contains remarks on a number of issues that were raised in the previous chapters. The chapter concludes with recommendations for future work and developments that should be done.

Chapter 2

Literature Review

2.0 QoS in IP networks

The Internet was originally designed for best effort traffic like web browsing and email. The emergence of real-time multimedia traffic like voice over IP and video conferencing has resulted in the need to provide QoS guarantee to traffic in IP networks. In communication networks, QoS is the ability to provide different priority to different traffic flows or users; or the ability to guarantee a certain level of performance to a data flow. In IP networks, this involves guaranteeing a required bit rate, delay, jitter, packet dropping probability and/or bit error rate [6]. QoS guarantees are important if the network capacity is insufficient, especially for real-time streaming multimedia applications such as voice over IP, online games and IPTV, since these in most cases require fixed bit rate and are sensitive to delay. QoS guarantees are also important in access networks with limited capacity, e.g. cellular data networks.

A network or protocol that supports QoS may agree on a traffic contract with the application and reserve network resources in nodes, for example during session establishment. During the session, the level of performance may be monitored by checking the data rate, delay, jitter and packet loss. The network can then dynamically control scheduling priorities in the network nodes. At the end of the session, the reserved network resources are released.

QoS therefore involves the subjection of network traffic to scrutiny and control. The concept, in IP networks involves the use of tools and protocols designed to aid the provision of defined and predictable data transfer characteristics. A QoS-enabled network must be able to meet the QoS requirements of different types of traffic. The network must also be able to handle applications requiring different QoS treatment within the same traffic class. The performance of a network is determined by the inherent delay, jitter, packet loss and throughput.

Throughput: Throughput refers to the amount of data or packets transferred on a communication channel or the data processed per unit time. The most common approach to QoS has been to increase network capacity by using fibre optic links and Gigabit Ethernet Network Elements. However, access technologies like WiMax do not have the same throughput as these networks. Inter-connecting the different technologies as is required in NGN platforms results in speed mismatches. Buffering would be required and this leads to the need to apply QoS techniques such as queuing and packet prioritisation. Increasing network capacity does not therefore solve end-to-end QoS problems, although it reduces the stress on QoS. Furthermore, different applications have varying data rate requirements. Video streaming, for example requires higher and guaranteed throughput, but lower delay and jitter requirements. VoIP on the other hand requires lower throughput but has strict jitter and delay requirements. QoS therefore involves the assignment of the required application bit rate or throughput on a network node.

Delay or Latency: This QoS parameter indicates the average time a packet takes to traverse the space between a source and a destination. The space may be comprised of the physical distance a packet must travel, in addition to the network routing and switching elements that cause additional delay. There are four broad types of delay in IP networks: propagation delay, transmission delay, codec delay (for video and audio applications) and de-jitter-buffer delay [3]. Codec delay is associated with the time it takes a voice or video signal to be processed into the appropriate codec. Different codecs are associated with different delays. For voice, a codec of G.729 is associated with 15ms while G.711 is associated with 3.875ms [7]. Propagation delay is the time it takes a packet to travel along a transmission medium. This includes the delay a packet experiences in network nodes along the path of the packet. De-jitter buffer delay arises when a de-jitter buffer, designed to reduce jitter on a network, holds packets for too long resulting in excessive delay.

Large values of delay cause echoes and talk overlap in voice traffic and may result in packet loss. The ITU-T specifies the delay tolerance of NGN applications. A delay of up to 150ms is specified as acceptable for voice traffic [3]. Any delay greater than 400ms is not acceptable for most communications.

Jitter: In IP networks, information is transferred in the form of packets. Consecutive

packets may not carry information from the same source. Packets from the same source are expected to experience the same delay as they are transferred from the source to the destination. Jitter is a result of packets from the same source experiencing different values of delay as they move along the communications channel to the destination. Large jitter values may cause packets to arrive in the wrong sequence. This causes jerky video and stutter/pop in audio applications [7]. Jitter is tolerable in applications like ftp and email because systems are able to store individual packets until all packets have been received and re-ordered. Jitter therefore affects real-time applications whose packets cannot be stored in buffers to delay playback at the receiver. In IP networks, jitter buffers are used to compensate for network jitter by buffering received packets and playing them out as a steady stream, but this also results in increased end-to-end delay [8]. Packets that arrive when the jitter buffer is full are discarded, resulting in packet loss.

Packet loss: During information transfer from source to destination, some or all of the information packets can be lost. The value of this QoS metric is given as a percentage of transmitted packets that never reach the intended destination. The primary cause of partial packet loss in IP networks is congestion in routers. When too many packets are simultaneously sent to a router, it discards some packets, assuming that the application that sent the packets will retransmit [7]. Complete packet loss is usually a result of a complete break in the communications channel and is not the subject of study in this thesis. A QoS-enabled network must ensure that in the event of partial packet loss, packets belonging to applications with stringent reliability requirements are given priority and hence must be the last to be discarded. Voice and video traffic is tolerant of packet loss as long as the packet loss does not occur in bursts, which result in a large number of information carrying packets being lost. The ITU-T recommends less than 1% packet loss for voice and less than 0.1% for video streaming applications [6]. Retransmission of voice traffic is not desirable and is impossible in real-time conversations.

End-to-end QoS: In IP networks, voice, video and data traffic is transported on the same communications channel, each traffic type presenting different QoS requirements to the network. In NGN, more than one network technology may be involved in the delivery of a service from source to destination. End-to-end QoS ensures that each service or application experiences consistent QoS treatment as it transverses networks with different

QoS systems. The QoS systems must be able to distinguish between the QoS requirements of the different applications and services. The ITU-T recommendation Y.1541 [3] defines network performance levels that are codified into performance objectives. The objectives are matched with delay, jitter, packet loss and data rate requirements of key NGN applications. End-to-end QoS provisioning in IP networks therefore involves ensuring that networks are able to meet application QoS requirements to the satisfaction of the end user.

Traffic classification in IP networks simplifies traffic management by grouping together network traffic into a limited number of classes based on the source or destination address, source or destination port number, or traffic type. Each traffic class is characterised by a set of predefined application QoS requirements in terms of delay, throughput, jitter and packet loss. In WiMax networks, the Caller Identification Number (CID) is a function of source and destination address, source and destination port address as well as traffic type. Traffic type is therefore used in the proposed QoS architecture.

Class-of-service (CoS) mapping provides an abstract way of providing consistent QoS guarantees to traffic traversing network domains that use different QoS systems. Network-specific QoS classes are mapped to a predefined set of QoS classes whose QoS parameters meet the QoS requirements of applications to be assigned to that class. The QoS specifications for classes in adjacent QoS systems must be the same. As an example, if the QoS parameter specifications for real-time video are guaranteed constant bit rate of 1Mbps and packet loss rate 1%, these values must be the same in the mapped QoS classes of the networks being inter-networked. The process is simple when the two domains belong to the same network operator. If however, the two domains belong to different network operators, there is need for trust between the operators since there would need to share sensitive network information like IP addresses.

In this thesis, it is assumed that the access and core networks belong to the same network operator. CoS mapping is implemented between the DiffServ modules in the WiMax ASN-GW and the core network edge router. The next section discusses the importance of QoS in NGN.

2.1 Importance of QoS in NGN

QoS is a key element in the delivery of service in NGN. This section discusses why network operators must address QoS when implementing NGN platforms and services.

QoS in NGN enables network operators to guarantee delivery of services to end-users at acceptable levels. Acceptability of NGN services, i.e. real-time applications like voice and IPTV, to an end-user depends on user perception. The ITU-T has created standards defining end-user "Quality of Experience" to address QoS as perceived by the end user [55]. Details of the relationship between QoS and QoE are given in appendix A. Network operators therefore use QoS as a basis for meeting end-user QoE. Satisfying end-user QoE results in a stable customer base and this can translate to guaranteed revenues.

Network operators are able to assign network resources according to QoS requirements of applications. Video and VoIP services, for example, are treated as premium services requiring low latency and jitter; web browsing and email are treated as best effort traffic and resources are assigned as and when they are available. This way, users with premium services are charged a higher fee for network usage.

Network operators can also go into Service Level Agreements (SLA) with subscribers or other network operators based on QoS requirements of traffic from the subscriber or network operator. Every service level agreement entered into, between a network operator and a subscriber or other operators, must be met with guarantees provided by the network operator to deliver the service according to the specified QoS parameters. The SLA's are used as the basis for charging for services [9]. If a network operator fails to meet the application QoS requirements and specifications, the subscriber may be eligible for a refund on the service resulting in loss of revenue. Continued failure to meet the QoS requirements can lead to users switching to other network operators, resulting in loss of future revenues.

QoS is also used to control usage of network resources [9]. A network operator keeps a database with end-user subscription information. Before a user is allowed to send data on the network, the database is checked to see if the user is registered on the network and to check the services the user can access on the network. End-users are only able to access services and network resources to which they are subscribed.

QoS enables network operators to use network resources optimally. Applications have different QoS requirements. QoS allocates network resources to applications according to their requirements and predefined specifications. QoS in private networks enables networks to handle different business application requirements and efficiently utilize the Wide Area Network (WAN) connections.

2.2 QoS provisioning in IP Networks

The primary goals of QoS in IP networks include the provision of dedicated throughput, improved packet loss characteristics and controlled jitter and latency, required by real-time and interactive applications [1]. QoS also ensures that when providing priority to one or more traffic flows, other flows do not fail. End-to-end or edge-to-edge QoS refers to a network's ability to deliver services needed by specific network traffic between two defined end-points of a network [6]. QoS in IP networks is generally provided at three basic levels. These are best effort, differentiated services and guaranteed services. Best effort service is characterized by a general lack of guarantees for traffic delivery. This is the service originally designed for the Internet. Best effort traffic includes file transfer, email and web browsing. The differentiated service provides preferential treatment of specific traffic. Delivery is not guaranteed, but some traffic can be assigned higher data rates, lower delay or experience lower packet loss rate. Examples of traffic that receives such treatment includes transactional services and audio and video streaming. Guaranteed service, also known as hard QoS, reserves network resources for specified traffic, which in most cases is real-time traffic. Examples are IPTV, video conferencing and VoIP. These services directly affect user experience, since they are real time and interactive.

2.2.1 QoS Provisioning Options

QoS provisioning in IP networks can be achieved in one of three ways:

1. QoS provisioning in Network Elements (NE): This involves the implementation of traffic classification, queuing disciplines, service scheduling and traffic shaping in network nodes.
2. QoS signalling: This involves the coordination of end-to-end QoS between Network Elements and network domains. A signalling protocol is used to transmit QoS

requirements in Network Elements along the path of an application and to reserve appropriate resources.

3. Policy Based Network Management: The use of existing network operator's policies and user information to define the level of QoS for an end-user or application.

Figure 5 shows a general overview of the three QoS mechanisms. Option 1 involves the implementation of QoS in routers and other network nodes. Option 2 involves the use of signalling protocols on the end-to-end path of an application to send QoS requirements to network nodes and network domains. Option 3 involves the use of policy-based management to control admission of traffic flows onto the networks and use of network resources. The following section describes the three techniques in more detail.

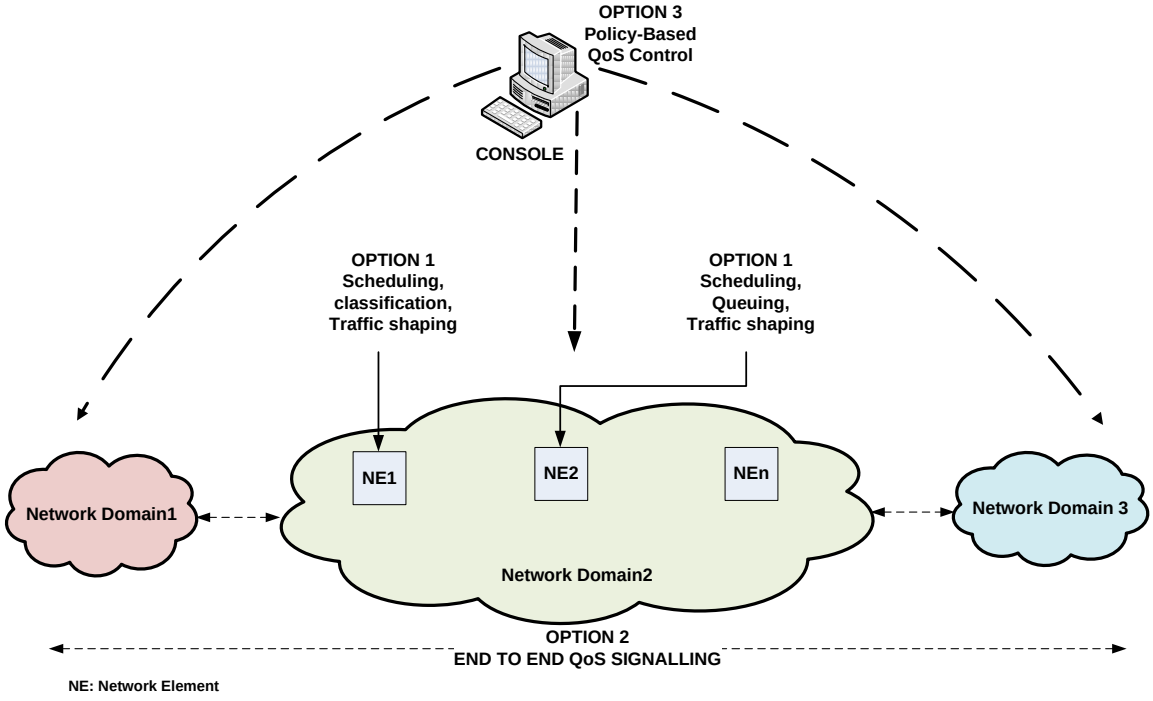


Figure 5: QoS control in IP networks using DiffServ mechanisms

2.2.2 QoS Control in Network Elements

QoS control in Network Elements involves implementation of QoS in routers. Each device along the path of an application must be able to guarantee the QoS required by the application. The Integrated Services (IntServ) QoS model, used in conjunction with the Resource Reservation Protocol (RSVP) signalling protocol [10] was the first standardized

QoS implementation in Internet routers. The IntServ model could not scale with large networks and was superseded by the Differentiated Services (DiffServ) QoS model [11]. DiffServ is able to provide QoS in IP networks through traffic classification and conditioning within a network node. It is possible, for example to provide on average low latency to voice and video traffic while at the same time providing best effort guarantees to non-critical services like web browsing. Traffic is classified according to source or destination IP address or according to traffic type and is assigned to a specific traffic class. Traffic is conditioned by being subjected to rate limiting, traffic policing or shaping. Figure 6 shows the DiffServ traffic conditioning components. NE_1 through NE_n represent Network Elements.

Two types of network nodes or routers form a DiffServ domain: boundary or edge routers, and interior routers [11]. Boundary nodes interconnect the domain to other DiffServ or non-DiffServ domains while interior nodes interconnect nodes within the same DiffServ domain. Within a domain, the QoS requirements of a group of services or applications can be guaranteed. Network nodes in one domain are configured to perform consistent QoS handling of traffic flows. Figure 6 shows the DiffServ traffic-conditioning block in a network node, typically a router. A network operator configures the DiffServ nodes. Within a DiffServ domain, traffic is classified and treated according to a service level agreement between the network operator and the customer or peer network operator.

Traffic classification is a function of the edge or border router. Traffic is directed to a logical output stream based on the IP addresses or DSCP value of the packet. The meter measures the temporal properties of the traffic stream. The instantaneous state of the meter is used to determine the operation on the packet by the marker or shaper/dropper. Packet dropping is the process of dropping packets based on pre-specified rules. Traffic shaping involves delaying packets of certain streams so that the stream conforms to defined traffic profiles.

The Internet Engineering Task Force (IETF) defines two groups of Per-hop Behaviours (PHB) for the DiffServ QoS model; the Expedited Forwarding (EF) PHB and the Assured Forwarding (AF) PHB [12]. The EF PHB is used to build low loss, low latency, low jitter, assured bandwidth resources and end-to-end service equivalent to a leased line, within a DiffServ domain. The AF PHB group defines four classes of service with each class

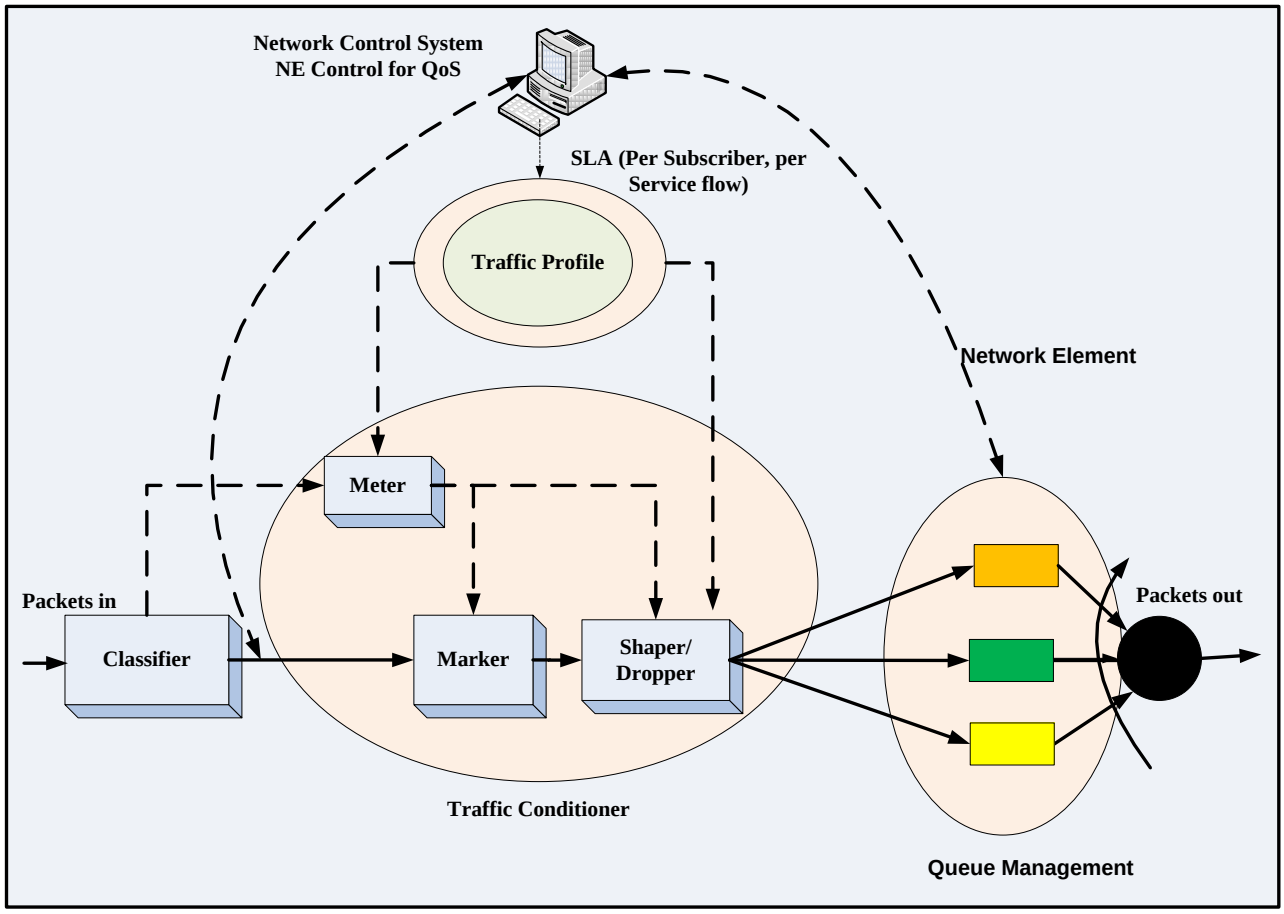


Figure 6: Components of the DiffServ traffic-conditioning block

allocated a portion of the available resources. Within each class, packets are marked with one of three possible drop precedencies. The drop precedence value marks the relative importance of the traffic stream within the AF class. In case of congestion, packets with lower drop precedence are protected from being discarded.

The DiffServ QoS model provides up to 2^6 (64) possible traffic classes. The IETF recommends the use of only one DSCP value for the EF class – (101110), while the recommended DSCP value for the AF classes is as shown in table 1 [13]. The AF PHB group is designed to cater for the varying QoS requirements for traffic like video streaming. Traffic in the EF PHB group includes VoIP or Video Conferencing, i.e. real-time traffic.

TABLE 1: RECOMMENDED DIFFSERV AF CODE POINT BINARY VALUES

Drop Precedence	Class 1	Class 2	Class 3	Class 4
Low	001010	010010	011010	100010
Medium	001100	010100	011100	100100
High	001110	010110	011110	100110

The absence of end-to-end signalling in the DiffServ QoS model has led to the development of other options for end-to-end QoS provisioning. Another problem of the model is the failure to scale under heavy traffic loads [14]. Moreover, the DiffServ model does not guarantee the performance of individual flows in a BA. The DiffServ QoS mechanism has also been criticized for its inability to provide consistent QoS handling in peering DiffServ domains because the behaviour of DiffServ routers in different DiffServ domains is unpredictable and may be different [14].

2.2.3 End-to-end QoS signalling

End-to-end QoS signalling is used for conveying applications QoS requirements, reservation of network resources across a network or discovering the best path for traffic flows. A signalling protocol is used between routers and switches to transmit data rate, delay, jitter and packet loss requirements of applications. Exchange of signalling information before data transfer also ensures controlled admission of traffic flows onto the network. Without admission control, a network can receive more traffic than it was designed to carry and compromise the QoS of the whole network. In addition to conveying application QoS requirements and network information, QoS signalling also makes it possible to provide network security through authentication. Only known and registered users are able to use network resources.

To accommodate the heterogeneity of transport network technologies and enable inter-networking on NGN, the ITU-T defined a Resource and Admission Control Function (RACF) layer in the ITU-T REC Y.2111 [4] for supporting QoS signalling. The RACF intermediates between transport technologies and control layer entities. The RACF hides service and transport network details from each other and manages QoS resources within transport networks. The RACF consists of a Policy Decision Functional Entity (PD-FE) and

a Transport Resource Control Functional Entity (TRC-FE). Figure 7 shows the architecture of the RACF [4].

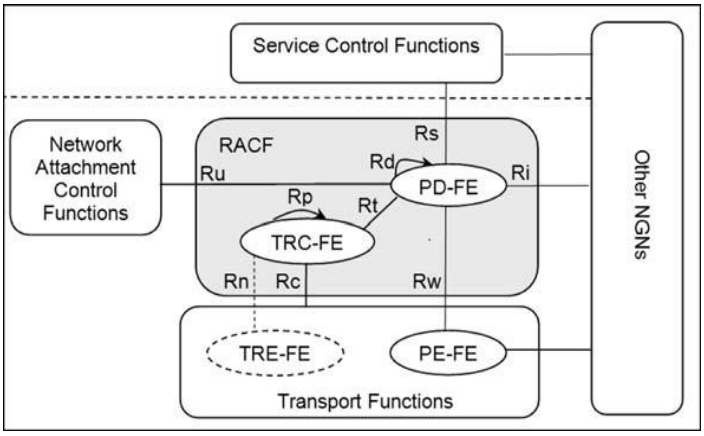


Figure 7: ITU-T RACF QoS modules and interfaces

Notes:

PD-FE: Policy-Decision Functional Entity
RACF: Resource Admission Control Function
TRC-FE: Transport Resource Control Functional Entity
PE-FE: Policy Enforcement – Functional Entity
TRE-FE: Transport Resource enforcement Functional Entity
R_n R_c, R_w R_u, R_p, R_s, R_d, R_t, R_i, R_s : interfaces

The PD-FE is an application aware entity. It translates resource requests from applications in the higher layers into class-of-service definitions based on applications QoS requirements, independent of the underlying transport technology. The PD-FE performs admission control by sending messages to the TRC-FE to check if the network can meet the QoS requirements of the application without compromising services already running on the network. The Network Attachment Control Function (NACF) module stores user related information. Before the admission of an application, the PD-FE checks this module for user subscription information like authentication and authorization.

The RACF architecture uses the Common Open Policy Service (COPS-PR), H.248, Diameter and Simple Network Management Protocol (SNMP) protocols for QoS signalling [4]. Within a network domain, a single RACF system is used. Where more than one network domain is involved, the RACF modules of the peer domains exchange the relevant information required for end-to-end QoS signalling. Although the ITU-T defines the RACF for NGN, most of the NGN technologies have in-built capabilities to support the

signalling capabilities defined by the IETF, i.e. RSVP and DiffServ.

2.2.4 QoS using Policy Based Network Management

In Policy-based network management, a set of predefined rules called policies [15] are used in Network Elements to manage QoS. A framework based on the COPS protocol, consists of a policy decision point, a policy enforcement point and a policy repository. The policy decision point acts in response to changes on the network conditions, and uses the rules in the policy repository to enforce the policies in the Network Elements through the policy enforcement points. In IP networks, the Policy Decision Point (PDP) can be a server controlled by the network operator and the repository can be a database that contains policy statements describing user profile, the type of traffic and their resource requirements and available network resources. A policy enforcement point can be a router that implements the policies as the traffic moves through the network.

The IP multimedia subsystem (IMS), developed by the 3rd Generation Partnership Project (3GPP) [16], has been adopted for the ITU-T NGN control layer. The IMS was originally designed by the 3GPP standards body to deliver IP multimedia services in mobile networks. It has since been adopted for use in NGN for delivering multimedia services in IP-based networks. The IMS core network is a set of SIP servers called Call Session Control Functions (CSCF) linked by standard interfaces. Figure 8 shows the architecture of the Open Source IMS core [17].

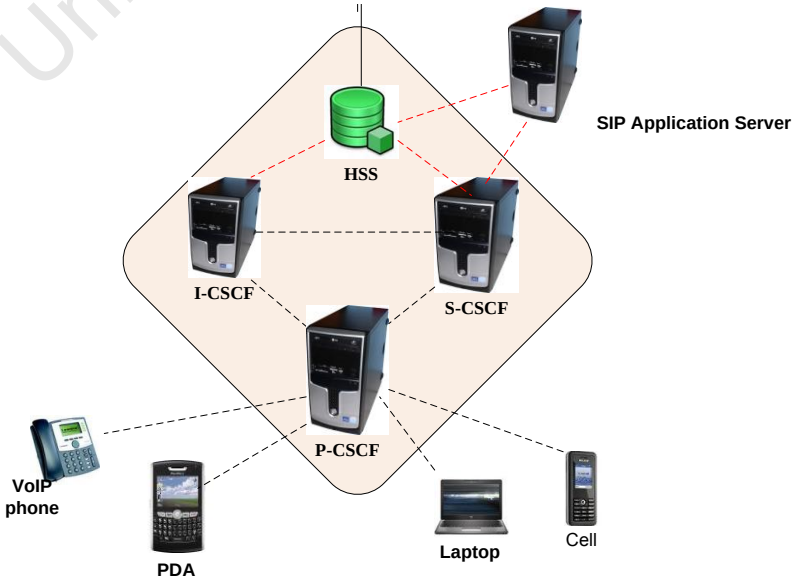


Figure 8: Open Source IMS (OSIMS) core

Home Subscriber Station (HSS): The central database that supports the IMS network entities that handle calls and sessions. It stores user subscription-related information. It also performs user authentication and authorization. As well as information related to the location of a user.

Proxy-Call Session Control Function (P-CSCF): The first point of contact for IMS terminals. It is located in the visited network. It can also be located in the home network in cases where the visited network is not IMS compliant. A session border controller is used in some networks for this function.

Serving Call Session Control Function (S-CSCF): This is the central server on the signaling plane, which acts as both as SIP server, and session control server. It is always located in the home network. The S-CSCF uses the Diameter protocol to upload and download user information to and from the HSS respectively. The S-CSCF inspects all SIP messages from the user. It also provides routing services and enforcement of network operator policies. The server also handles SIP registrations which bind a user to a location.

Interrogating Call Session Control Function (I-CSCF): This is a SIP proxy located at the edge of an administrative domain. It acts as a proxy for users in their home network. Functions of the server include contacting the HSS to retrieve user location and determine which S-CSCF is assigned to the user. In earlier releases of the IMS the I-CSCF is used to hide the topology of the internal network from the outside world by encrypting SIP messages. This function is however removed in releases 7 and later and moved to the Interconnection Border Control Function (IBCF). The IBCF is used to do Network Address Translation (NAT) and firewall functions (pin-holing) between peer networks.

Application servers (AS): AS's host and execute services for the IMS. These can be located in the home network of a user or in a third party Application Service Provider (ASP) network. Where the application servers are collocated in the IMS core network, they communicate directly with the IMS Network Elements using the SIP signaling protocol. Where the AS's are located in APS networks, an open API is required to enable independently developed applications to communicate with the IMS elements. Such API's include those developed under the Parlay/Open Mobile Alliance (OMA) and web2.0 [16].

Interconnecting gateways: Gateways and gateway controllers interconnect the IMS to

different networks including Public Switched Telephone Networks (PSTN) to enable the exchange of session information between end-users, the IMS and transport networks using common interfaces and protocols.

IMS End user Devices: Access and core networks allow SIP devices (IMS clients) to establish IP connectivity and connect to the IMS network. The transport networks also enable the exchange of SIP messages between the end-users and the IMS core network. Clients access the IMS using any IP-based transport layer technology. Media gateways allow the devices to place or receive calls to and from PSTN or any Circuit Switched (CS) network. Once IP connectivity is established, the clients are responsible for their own IMS interactions independent of the transport network.

In the IMS, COPS is used in conjunction with the Session Initiation Protocol (SIP) to achieve call and session control while enforcing a set of policies to achieve QoS. A policy control function located in the IMS Call Session Control Function (CSCF) acts as the PDP. The policy enforcement points are transport layer routers. QoS is achieved by making decisions to change traffic management policies, for example changing the traffic rate in the routers, priority queuing or DiffServ Code Point (DSCP) values to alter the delay or packet loss.

Work is currently underway in various standardization bodies in developing policy-based network management frameworks for achieving QoS network transport technologies [18]. The aim is to develop a general architecture that caters for the heterogeneous transport network environment of NGN platforms.

2.3 Inter-domain QoS

Another approach to end-to-end QoS in IP networks is that of using independent networks to monitor several network domains on behalf of a user and direct traffic to the network that best meets the QoS requirements of the application. The Bandwidth Broker [19] and Overlay networks [20] are such solutions. A Bandwidth Broker (BB) manages the QoS within a network domain and makes decisions based on the knowledge of network resource availability within the domain. It communicates with peer BB's and negotiate service level specifications for inter-domain traffic. An overlay network is built on top of other networks and the nodes are connected via virtual or logical links that correspond to a path. Overlay networks provide QoS guarantees for multimedia traffic by real-time

monitoring and routing traffic through paths that meet the QoS requirements of the applications. Overlay networks can run independently of underlying networks, making the solutions more attractive for end-to-end QoS provisioning to network operators. Implementation of both the BB and overlay networks does not require changes in the existing network infrastructure.

To understand end-to-end QoS in NGN, in addition to discussing QoS issues in IP core networks, QoS in access networks should be covered. The following section therefore describes the QoS techniques in IEEE 802.16 networks. A description of how the standard is extended to enable integration with IP-based networks is also given.

2.4 QoS in IEEE 802.16 networks

The IEEE 802.16 standard defines a broadband Wireless Metropolitan Area Network (Wireless MAN) that is QoS-enabled on the physical and MAC layers [21]. Commonly known as WiMax, the IEEE 802.16 technology has become an alternative access technology to Digital Subscriber Line (DSL) and cable modem because of the low investment cost and ease of deployment associated with wireless networks. Unlike WLAN technologies, WiMax networks provide QoS. Compared to 3G mobile networks, WiMax is capable of delivering high data rates suitable for applications like video streaming and video teleconferencing [22]. The original fixed access IEEE 802.16 standard has been enhanced to add mobility to the standard with later family members of the standard i.e. IEEE 802.16n, being designed to meet data rates required for 4G networks [23]. Mobile 802.16 networks bridge the gap between very high data rate wireless local area networks and very high mobility cellular systems [22].

2.4.1 QoS on the physical layer

On the physical layer, the IEEE 802.16 technology makes use of several QoS techniques known to reduce interference, increase throughput and use available frequency efficiently. The techniques result in reduced latency, jitter and packet loss on the networks. Some of the physical layer features that are instrumental in giving the technology the power to deliver robust performance in a broad range of channel environments are flexible channel widths, adaptive burst profiles and forward error correction with concatenated Reed-Solomon and convolutional encoding [22].

The standard also uses Frequency Division Duplex (FDD) and Time Division Duplex (TDD) [23] techniques to provide QoS on the air interface. Networks configured to work in FDD mode allow the transmission and reception of signals on different sub-bands. Separation of transmit and receive channels results in reduced interference. This also allows flexible bandwidth allocation and higher throughput.

IEEE 802.16 networks transmit data over the air interface in frames and each frame is divided into uplink and downlink sub-frames. The sub-frames are further divided into slots. A Guard slot separates the uplink and downlink sub-frames. When configured to work in TDD mode, a system dynamically allocates bandwidth to the uplink or downlink by shifting time slots between two sub-frames depending on user bandwidth requirements. This is very important especially for Internet traffic, which is dominant in IP networks. Internet traffic is asymmetric, consisting of short durations of uplink traffic, when users send requests to the Internet, and long durations during download.

FEC builds redundancy in wireless systems by repeating information bits. Missing or errored bits are corrected at the receiving end. Without FEC, in the event of errors in the information-carrying bits, the sender must resend a complete frame, which includes both errored and error-free bits. This results in increased traffic on the link causing higher latency and inability to meet the QoS requirements of services and applications. The OFDM technique also makes it possible to use bit-interleaving [24]. This means bits carrying the same information, are transmitted using sub-carrier frequencies, which are on different frequencies. As a result, multiple copies of bits are carried on several sub-carriers; as a result, copies of any bits weakened on the air interface during transmission can still be received at the destination. The system makes use of Quadrature Phase Shift Keying (QPSK) and Quadrature Amplitude Modulation (QAM) [24]. QAM enables the systems to deliver higher throughput, but is more susceptible to interference and is therefore unsuitable for long links. Use of QPSK makes it possible for the systems to have a longer coverage range between the base stations and subscriber stations but with a resultant lower throughput. Therefore; there is a trade-off between throughput and range with higher throughput more achievable at shorter ranges. The networks therefore use adaptive modulation techniques for better QoS, adapting to QPSK modulation in case of long links with a resultant lower throughput [24] switching to QAM in case of short links. Under adverse channel conditions in shorter links, the system can also apply QPSK

technique on the channel to maintain QoS. Figure 9 shows the changes in modulation as link distance increases. While BPSK enables use of long links, the throughput achieved is lower.

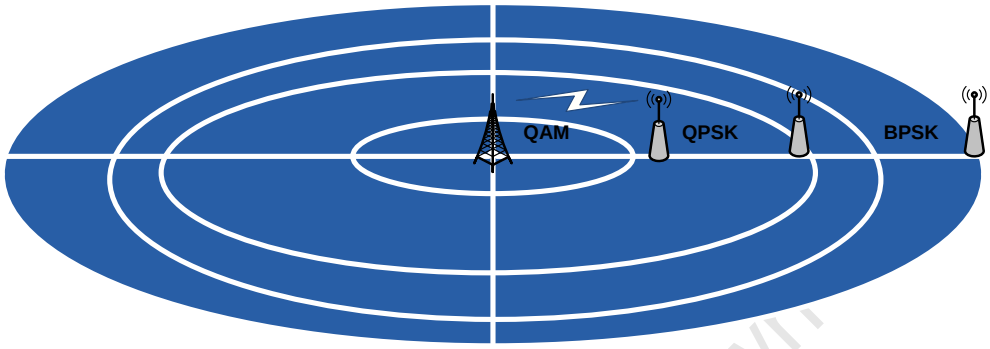


Figure 9: Effects of different modulation techniques on QoS

Using QAM for short distances achieves high throughput but is unsuitable for long distances as it results in errors in the transmissions between the base station and the subscriber station.

2.4.2 QoS Control on the MAC Layer

A key distinguishing feature between IEEE 802.16 and IEEE 802.11 and 3G wireless technologies is that it is connection oriented. Packet forwarding is therefore faster and QoS is offered per connection. Two sub-layers of the 802.16 MAC layer are responsible for QoS, the Convergence Sub-layer (CS) and the Common Part Sub-layer (CPS). The CS handles traffic flows from higher layer ATM, Ethernet and IP-based transport networks. This layer maps traffic flows from the transport networks to CPS. The CPS then fragments and segments the service data units into protocol data units (PDU).

Figure 10 shows the 802.16 MAC layer QoS mechanism [25]. Each connection from a subscriber station is assigned a connection identity (CID) and a service flow identity (SFID). The SFID's are used to identify traffic flows with the same QoS parameters and place them into one of five classes of service; Unsolicited Grant Service (UGS), real-time polling service (rt-PS), non-real time polling service (nrtPS) or best effort (BE) [25].

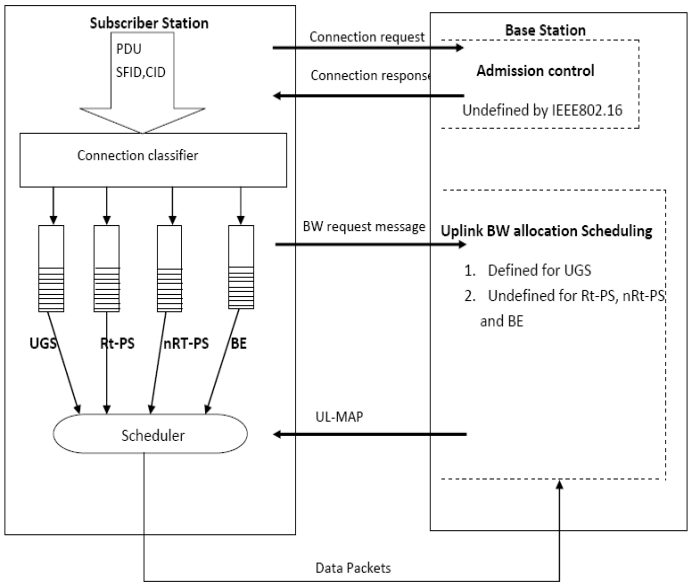


Figure 10: IEEE802.16 QoS implementation on the MAC layer

Notes:

PDU: Protocol Data Unit
CID: Connection ID
UDS: Unsolicited Grant Access
Rt-PS: Real time polling service
nRt-PS: non-Real-Time Polling Service
BE: Best Effort Polling

In addition to these classes, the extended real-time polling service (ertPS) is also defined in Mobile WiMax and is intended for voice with silence suppression. Table 2 shows the 802.16 QoS classes, examples of services for each class and QoS requirements of the traffic in the class.

TABLE 2: IEEE 802.16 QOS CLASSES

Class	Service Type	Requirements
UGS	VoIP (no silence suppression)	Low latency and jitter, fixed size data packets, constant bit rate
rt-PS	Video/Audio streaming	Real time traffic that is bursty in nature, variable size data packets
nrt-PS	File transfers (FTP)	Non-real time traffic, variable size data grants, guaranteed bandwidth, delay tolerant.
ert-PS	Voice with silence suppression	Enhancements to the standard to allow mobility.
BE	Email, web browsing	No QoS guarantees. Uses available network resources

Notes:

UGS: Unsolicited Grant Service
Nrt-PS: Non-real time Polling Service
BE: Best Effort
Rt-PS: Real time Polling Service
Ert-PS: Extended real time Polling Service

Allocation of bandwidth resources to applications on the 802.16 networks is through bandwidth request and grant messages between the BS and the SS. For the UGS, a SS is granted bandwidth implicitly at connection set up. For polling services, either a SS sends a bandwidth request message incrementally or sends its aggregate requirements for the connection in response to received polling message from the BS. Best effort service is used for applications with no QoS requirements. In this case, the SS issues a bandwidth request message in a contention period.

2.4.3 Extending IEEE 802.16 QoS to NGN

The IEEE 802.16 standard has been commercialized under the name WiMax by the industry alliance, the WiMax Forum. The mission of the Forum is to promote and certify compatibility and interoperability of broadband wireless products based on the standard. The Network Working Group (NWG) within the forum was formed to develop technical specifications beyond those defined in the scope of the 802.16 standard and enable the technology to interwork with IP-based networks [24]. To date there is no standardized mechanism for interworking the 802.16 QoS to IP networks.

The IEEE 802.16 standard defines QoS on the Physical and MAC layers but it does not specify how the QoS metrics can be translated when interworking the technology with other networks. The WiMax Forum NWG however, recommends the use of DiffServ QoS classes for translation of the 802.16 QoS classes to QoS classes of IP-based networks [25]. This ensures consistent QoS in a heterogeneous network environment. The IMS provides QoS on NGN. The NWG focus is therefore on the development of a WiMax architecture that will enable interworking the technology with the IMS. Although not yet completed, the WiMax Forum has incorporated most of the concepts of the IMS into the WiMax architecture [18].

On the IMS, QoS provisioning is policy-based and clients access services depending on their applications' QoS requirements and network resource availability. The Home Subscriber Server (HSS) on the IMS stores the subscriber QoS profiles and associated policies used for making the decisions to allow or deny services. The behaviour of traffic flows once admitted on a network is monitored through the policy decision/enforcement framework of the IMS to ensure that they adhere to the requested QoS. To date,

interworking between WiMax and the IMS is limited to the WiMax network providing a transport path between the IMS core and IMS clients located in end-user networks.

2.5 Work related to end-to-end QoS in NGN

In section 2.4 QoS provisioning techniques in IP and WiMax networks were discussed. Consistent QoS on the NGN is achieved when applications are subjected to the same QoS treatment across the heterogeneous transport technologies. End-to-end QoS has been achieved using signaling protocols, policy-based techniques and creation of domains in which service flows are subjected to the same QoS treatment, e.g. DiffServ and MPLS [27] networks. This section therefore reviews work related to end-to-end QoS in IP-based networks based on these techniques. The implementation of test beds for validation of research work in NGN is also discussed. The section concludes with a discussion on the use of class of service mapping for achieving end-to-end QoS in NGN.

2.5.1 Achieving end-to-end QoS using signaling protocols

To achieve end-to-end QoS signalling in heterogeneous network environments, extensions to RSVP protocol have been proposed [28]. Extensions to the protocol have also been proposed to enable QoS interworking between UMTS and WLAN technologies [29]. The RSVP-Traffic Engineering (RSVP-TE) extension has been proposed for use in Multiprotocol Label Switched (MPLS) networks for end-to-end QoS signalling [30]. Some of the drawbacks of the use of the protocol include signalling overhead and the inability of applications to signal for QoS requirements.

An end-to-end QoS model using the NSIS protocol [31], currently under development within the IETF, was also proposed. The QoS model, used on the project known as the WiMax Extension for Isolated Research Data networks (WEIRD), was applied to heterogeneous networks using WiMax as an access network [32]. The proposed solution involves installation of NSIS modules on network nodes across the domains involved in the end-to-end delivery of the service/application. As highlighted earlier, network operators are reluctant to accept any solution that involves installation of modules on network nodes on existing networks. The implementation uses DiffServ for traffic classification. While the DiffServ QoS model has drawbacks, as highlighted by the authors, it remains the standard for traffic classification defined by the ITU-T for IP networks.

Further work on the WEIRD project [32] makes use of SIP [33] in conjunction with the NSIS protocol for end-to-end QoS signalling. The proposed integration addresses the fact that on the NGN architecture, IMS is the standard control layer network and SIP is used for QoS signalling between terminals, IMS elements and application servers.

Performance evaluation of WiMax using real-time multimedia applications has also been carried out on a test-bed [34] and SIP and NSIS as signalling protocols were used for applications QoS signalling. A problem identified is the delay between acquisition of data and interpretation of the data by the end-user. Further work on the project is aimed at reducing the signalling delay to make the application more real-time like.

2.5.2 Achieving end-to-end QoS by implementing QoS in Network Elements

The DiffServ QoS model is widely used in providing QoS in IP routers. On its own, the DiffServ has a number of drawbacks [35]. Implementation on individual network nodes results in network nodes treating packets independently. Where more than one network domain is involved uniform, packet handling is lost, resulting in end-to-end peering problems. The model has therefore been used in conjunction with other techniques to achieve consistent QoS in heterogeneous network environments.

An approach for achieving end-to-end QoS based on the concept of Network Planes (NP) and Parallel Internets (PI) was proposed in [35]. Commonly referred to as the AGAVE (A lightweight Approach for viable end-to-end IP-based QoS services) project, the platform is based on a business model that consists of the customer, a Service Provider and an IP network provider. A proposal is also given for interworking the architecture with the IMS. In the architecture, QoS is achieved using DiffServ, multi-protocol routing services, e.g. Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP); dedicated label switched paths and RSVP traffic extension (RSVP-TE) [27] for signalling. IP tunnelling and overlay routing is used for traffic with less stringent QoS requirements. For reliability, IP/MPLS fast rerouting is used.

The AGAVE project provides a good basis for testing end-to-end QoS control for the NGN architecture using WiMax as a radio access technology. In this implementation, access network QoS classes are mapped on a one to one basis with those of an IP network domain. If the core network experiences congestion, traffic from the wireless network is

more likely to experience further delay and packet loss due to retransmissions caused by TCP, which is used for congestion control in IP networks.

2.5.3 Achieving end-to-end QoS by using policy-based QoS techniques

Policy-based QoS has been proposed for an integrated 3G radio access network and the IMS [36]. 3G has evolved to LTE [37], one of the NGN access network technologies. This thesis provides an analysis of QoS on NGN platforms, using WiMax inter-connected with an IP core network and the IMS, thus providing a platform with network technologies of the NGN.

QoS in a heterogeneous wireless network environment using policy-based techniques has also been proposed for integrated UMTS and Wireless LAN systems [38]. The proposed architecture has a number of drawbacks. It does not use standard service level specifications. Network security is a concern when more than one network operator is involved. The process of QoS negotiation is also slowed down when there is a chain of network domains involved in the QoS negotiation process. A QoS control system using the IMS can solve some of these problems since the IMS provides services independent of the underlying transport technologies. The evolution of the UMTS technology to LTE and the availability of broadband QoS-enabled WLAN technologies also addresses some of these issues.

The Management of Networks and Services in Diversified Radio Environments (MONASIDRE) project [39] uses the COPS protocol to implement QoS in heterogeneous network environments. DiffServ is used for QoS translation between the different network domains. SIP is used in conjunction with COPS for call and session control to achieve QoS and service control in the multimedia environment. The work is based on simulations. Simulations have the drawback of requiring further work when implementing the solution in practical networks. Further work is proposed to evaluate the performance of the proposed framework in large network environments.

The technologies used in the MONASIDRE project are evolving to broadband QoS-enabled networks that can be used on NGN platforms. The IMS will therefore make it possible for end-user to access services using any of the access technologies. Work is already underway in several standardization bodies (ITU-T, 3GPP, ETSI, WiMax Forum and Cable LABS) related to interworking transport technologies with the RACF and IMS

[18]. Policy-based QoS in NGN using the IMS is likely to be adopted. The next section presents work related to NGN research test-bed implementations.

2.5.4 NGN Research Networks and Implementations

Research test-beds create an environment for NGN interoperability testing and benchmarking. They also provide an environment for validation of research results in a practical network environment. Some of the research test-beds established for this purpose include the Fraunhofer Institute Fokus/Technical University of Berlin in Germany for research in NGN applications and technologies [40]. A number of new projects have also been initiated for prototyping Future Internet Infrastructures that will enable research work in Internet Security, cross-domain networking, distributed service architectures and infrastructure virtualization [41].

Large-scale test bed implementations for testing NGN services and applications have also been implemented in Europe and these include the Pan-European Laboratory Infrastructure Implementation (PII) [43] and the Fraunhofer Institute FOKUS [44] research test bed in Germany. The later has resulted in the implementation of test-beds including the University of Cape Town IMS client (UCT IMS client) [45]. Earlier work on the implementation of NGN research is also explained in [42] with a focus on testing network interoperability.

The AKARI project [46] in Japan provides the future of research test beds as the NGN evolves to the New Generation Networks (NWGN). Work on the design of the test bed, which is designed for carrying out proof-of-concept tests and prototyping, is ongoing until 2010, with first field trials of the technologies expected in 2015.

While advanced test bed implementations provide direction for research and development in new technologies, prototypes based on NGN technologies still play a key role, given the fact that network operators are still faced with integration and interworking challenges in the roll out of NGN platforms. While research work in developed countries is now focused on applications development, there is very little or no research in NGN technologies in developing countries. Implementation of open-source research NGN test beds is expected to assist in knowledge acquisition. This will also speed up the rollout of NGN in these countries that are still lagging in the development of basic Internet infrastructures.

This thesis therefore focuses on the implementation of an NGN prototype based on NGN

technologies with most of the implementations done using open source software. The work specifically focuses on the implementation end-to-end QoS using the concept of class of service mapping between WiMax and an IP core network on the access and core layers respectively. The next section therefore discusses the concept of class-of-service mapping in heterogeneous network environments.

2.5.5 Achieving end-to-end QoS by using class of service mapping

The previous sections described various techniques used to achieve end-to-end QoS in IP networks. These include the use of signalling protocols, implementation of QoS in Network Elements in defined QoS domains and use of policy-based network management techniques. In order for applications to experience consistent QoS in heterogeneous networks, they must be subjected to the same QoS treatment across the different networks. This section therefore discusses the concept of class of service mapping in IP networks using the DiffServ QoS model for traffic classification. An interrelationship between QoS classes of different transport technologies is deduced.

A class of service defines service classes for traffic with defined QoS characteristics. For example, web browsing and file transfer traffic may be placed in one class of service and video traffic in another class. Traffic can be marked with its class of service membership using DSCP or 802.16p. End-to-end QoS guarantees that the applications in defined classes will receive QoS handling defined for that particular class of service, for example, guaranteed throughput, delay or packet loss. Implementation of QoS requires per device class of service configuration or some way of signalling the QoS requirements of the applications. Class of service mapping involves the translation of the QoS classes of one network domain to those of another domain in order to achieve consistent QoS between the two domains.

The ITU-T defines eight QoS classes of service for traffic in IP networks [3]. Different transport technologies also define QoS classes of service based on the ITU-T QoS classes. Table 3 shows the QoS classes of WiMax, UMTS and MPLS networks. The table illustrates how DiffServ QoS classes are mapped to IP QoS classes, which are in turn used in these networks to classify traffic. The first two columns present traffic classes in the DiffServ QoS model as defined by the IETF. The DiffServ classes of service are mapped to the Type of Service (TOS) bits in IPv4 or Traffic class in IPv6. The UMTS

classes of service are mapped to DiffServ QoS classes and traffic classification is based on the ITU-T class-of-service definitions outlined in ITU-T Y.1541 [3].

TABLE 3: TRANSLATING QOS CLASSES USING DIFFSERV

DiffServ PHB & DiffServ (3Most Significant Bits (MSB) of IP Type Of Service bits (TOS)) - IETF		IP Precedence bits (3MSB Of TOS bits) –ITU-T		MPLS Experimental IP bits values in Shim header - MPLS	WiMax - IEEE802.16	UMTS - 3GPP
BE	**000000**	**000**	**Routine traffic**	**000**	**Best Effort**	**Background**
AF1	**001000 (Class of Service)**	001	Priority	001	**Ert-VR**	**Interactive**
AF11	001010					
AF12	001100					
AF13	001110					
AF2	**010000 (Class of Service)**	010	Immediate	010		
AF21	010010					
AF22	010100					
AF23	010110					
AF3	**011000 (Class of Service)**	011	Flash	011		
AF31	011010					
AF32	011100					
AF33	011110					
AF4	**100000 (Class of Service)**	100	Flash over drive	---	**Nrt-PS**	**Streaming**
AF41	100010					
AF42	100100					
AF43	100110					
EF	**101000 (Class of Service)**	101	Critical	101	**Rt-VR**	**Conversational**
	101110				**UGS**	
Network Control	**110000**	110	reserved	---	**---**	**---**
Network Control	**111000**	111	reserved	111	---	---

The recommendation classifies IP traffic into eight QoS classes. The ITU-T also maps the IP classes into DiffServ QoS classes in amendment of the ITU-T Y.1541 [3]. Class-of-service mapping has been proposed in [47] to achieve end-to-end QoS in a UMTS/IP core environment. The UMTS classes of service are mapped to DiffServ QoS classes in an IP core network. MPLS networks also define QoS classes that can be mapped to DiffServ QoS classes whenever MPLS networks are inter-networked with IP-based routers [48] to achieve end-to-end QoS. In [49] a description of the performance evaluation of IP/MPLS networks where class of service mapping is used to achieve end-to-end QoS is given. In WiMax networks, traffic classification is also based on the ITU-T traffic classes defined in the ITU-T Rec Y.1541 [3]. When interworking the networks with IP-based networks, the WiMax Forum recommends the use of DiffServ QoS classes for translation of the WiMax QoS classes to IP QoS classes.

This thesis therefore focuses on the translation of QoS classes between a WiMax access network and IP core network on an NGN prototype test-bed to achieve consistent QoS

between the two networks. The platform also makes use of the IMS as a control layer network.

2.6 ITU-T traffic classification at the IP network level

The proposed QoS translation between the access and the core networks is based on the use of QoS classes to translate DiffServ QoS classes between the two networks. Traffic classification enables networks to group together traffic with similar QoS requirements. Class of service mapping enables translation of QoS parameters defined in one network domain, to a set of QoS parameters in another domain. Table 4 shows ITU-T-defined IP classes of service and some typical applications [3]. The recommendation defines eight QoS classes, two of which are provisional and are not shown in the table for clarity.

TABLE 4: IP QOS CLASSES AND EXAMPLES OF APPLICATIONS

IP	Examples of applications
0	Real-time, jitter sensitive, high interaction (VoIP, Video Tele-conferencing (VTC))
1	Real-time, jitter sensitive, interactive (VoIP, VTC).
2	Transaction data, highly interactive (Signalling)
3	Transaction data, interactive
4	Low loss only (short transactions, bulk data, video streaming)
5	Traditional applications of default IP networks e.g. web browsing and email

The table illustrates the complex QoS requirements of the various applications that are expected to use the same transport channels on NGN.

TABLE 5: IP TO DIFFSERV QOS MAPPING

IP	DiffServ QoS classes
0, 1	Expedited Forwarding (EF) - Dedicated Bandwidth
2, 3, 4	Assured Forwarding (AF)
5	Best Effort (BE) - Default

In an amendment of the ITU-T REC 1541 [50], the IP classes of services are mapped to the DiffServ classes of service as shown in Table 5, to enable classification of multimedia traffic in IP networks.

2.7 Discussion

This chapter presented the concepts of QoS in IP networks. The chapter started with definitions of the QoS metrics in IP networks: delay, jitter, packet loss and throughput. This

was followed by a discussion of the importance of QoS in NGN. Challenges faced by network operators in the QoS implementation in NGN were presented. A number of mechanisms that are used to address these challenges were discussed. Since this work focuses on QoS in a NGN prototype using WiMax as an access network, a detailed description of the IEEE802.16 QoS techniques was given.

Work related to QoS in heterogeneous network environments was also analysed. The analysis focused on UMTS, WiMax and WLAN networks inter-networked with IP-based networks. The implementations however, consist of networks that will not be access technologies for NGN platforms. The work studied also lacked a complete NGN environment with all the network components of NGN architecture. Some examples of test-beds set up for validation of researched work related to NGN were also discussed. These implementations are based on access and application layer, access and core layer technologies, or focusing on the control and applications/services layers. The chapter ends with a discussion of how QoS classes can be translated from one network domain to another using the DiffServ QoS model and IP classes of service. The next chapter presents a discussion of the architecture of NGN platforms.

Chapter 3

End-to-end QoS architecture for a NGN system

3.0 Introduction

One of the challenges faced by network operators when rolling out NGN's was highlighted in chapter 1 as the inability to guarantee consistent QoS across heterogeneous transport networks that use diverse QoS techniques. Chapter 2 reviewed literature related to QoS in packet-based networks. Key concepts relating to QoS in IP networks were also outlined. Most important to the QoS architecture proposed in this thesis are the QoS parameters and the concepts of traffic classification and class-of-service mapping. The chapter also presented a detailed analysis of QoS mechanisms in IEEE 802.16 and IP core networks. Previous work aimed at addressing end-to-end QoS challenges in NGN systems was also given.

The work done in this thesis involves three modular units. Firstly, the design of a QoS architecture that addresses the end-to-end QoS problem between two network domains, secondly, implementation of a QoS-enabled IP core network, and lastly implementation of the proposed QoS system on a prototype NGN system. The first and second units form the proposal central to this thesis research. This chapter therefore presents the proposed QoS architecture designed to address the end-to-end QoS problem in an NGN system that utilizes a WiMax access network and an IP core network. The chapter ends by presenting the design and architecture of a QoS-enabled NGN system. It is important that the architecture be explained so that the functionality of the proposed QoS architecture is clearly understood. Later chapters evaluate the performance of the end-to-end QoS architecture on a NGN prototype.

3.1 End-to-end QoS architecture Design

Before the proposed QoS architecture is discussed, it is important to re-address some of the challenges in providing end-to-end QoS guarantees in interconnected NGN transport networks. NGN systems are composed of heterogeneous QoS-enabled technologies. While the current NGN technologies are able to meet the QoS requirements of real time

multimedia applications, the problem faced by network operators is how to ensure consistent end-to-end QoS guarantees to applications traversing the different network technologies.

Policy-based QoS management (PBM) solutions allow applications to dynamically request QoS requirements before data can be transferred between end users. This solution requires implementation of policy decision and policy enforcement modules in the transport networks. Most network operators are not willing to implement these modules in their existing networks, due to disruption of service. QoS signalling protocols like NSIS [34] and RSVP [10] have been used to send applications QoS requirements in different networks. This also requires implementation of new modules in network nodes. DiffServ is the current default QoS model in IP networks and most IP-based network nodes have a DiffServ module that can be activated as and when the model needs to be used. The model allows static traffic classification and conditioning in network nodes.

The proposed QoS architecture therefore uses DiffServ as the underlying QoS model for solving the end-to-end QoS problem in WiMax and core IP networks. The reason for this is that, as a first step towards addressing the end-to-end QoS problem between WiMax and IP core networks, the WiMax Forum specifies that the DiffServ QoS model may be used to interwork the WiMax QoS system to that of adjacent IP core networks. The forum does not however specify how this can be done.

This thesis therefore proposes a QoS architecture that uses the DiffServ model to translate QoS specifications between a WiMax network and an IP core network. The architecture uses the concepts of traffic classification and class-of-service mapping to define the transfer of QoS specifications between QoS systems in different network domains. A special class-of-service mapping where best effort traffic from the wireless network is mapped to the AF QoS class, thus giving higher priority to this network in the core network. Figure 11 shows the use of the DiffServ model on the edge nodes of a WiMax are IP core network. Consistent QoS between the two networks is achieved when identical QoS classes are mapped to each other between on the edge nodes.

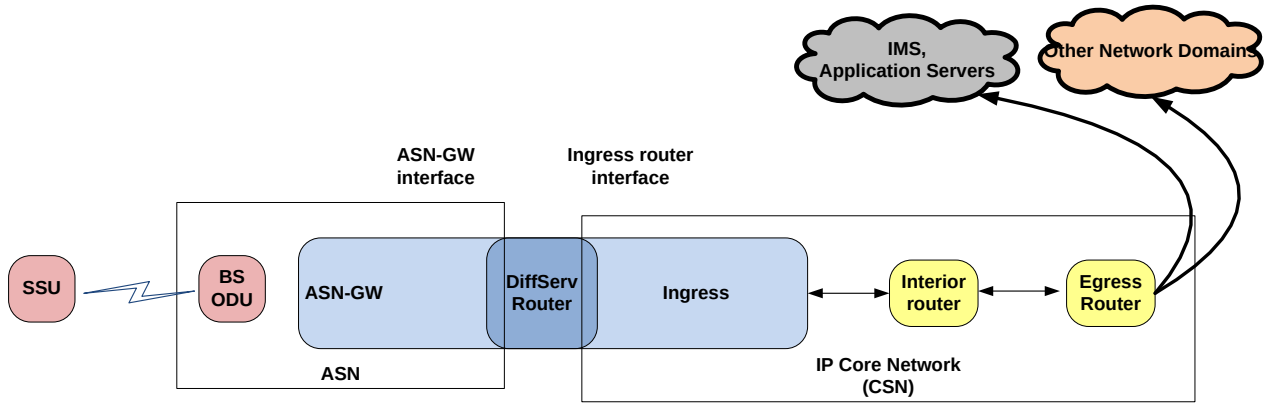


Figure 11: WiMax/IP QoS translation using DiffServ QoS classes

NGN transport technologies have unique QoS systems that are capable of providing QoS guarantees to various real time media. Each QoS system therefore has a concrete set of QoS classes, with each class defining the applications QoS specifications, i.e. delay, jitter, throughput and packet loss, assigned to it. It is also assumed that the networks' QoS systems include call admission control frameworks that are needed to control resource usage in each network, as well as end-to-end QoS signalling protocols. DiffServ therefore ensures transfer of the same QoS specifications from node to node and also from one network domain to the other.

Traffic flows are statically classified in the individual network domains into DiffServ QoS classes, then mapping the traffic classes to each other. In the access network, WiMax QoS classes are mapped to DiffServ QoS classes. A DiffServ module installed in the ASN-GW allows WiMax traffic classes to be administratively placed into DiffServ traffic classes via a network management interface. DiffServ Code Point (DSCP) values assigned to each traffic class identify the DiffServ class into which a traffic flow is placed and hence the QoS characteristics of the traffic.

In the core network, traffic classification is based on the ITU-T traffic classification for IP networks. Traffic is statically assigned into the QoS classes via an API. The QoS specifications, i.e. delay, throughput, jitter and packet loss, defined for traffic in each of the DiffServ classes must be the same in both the WiMax and core networks. This is achieved by assigning the same DSCP in the core network edge router and in the WiMax network

ASN-GW.

In fixed wireless networks, traffic can be subjected to packet loss due to fading or low bandwidth conditions due to admission control. In general, applications using TCP/IP may not be able to make the reason for packet loss. The mechanisms invoked by TCP/IP may therefore result in requesting for retransmissions on the air interface, when it is not necessary. This causes further degradation of service on the air interface due to increase traffic. The proposed QoS architecture therefore maps BE traffic from WiMax and other wireless networks into the AF_4 of the DiffServ model. In the event of congestion in the core network, BE traffic from the wireless network is not immediately dropped due to the higher priority assigned. The QoS on the air interface does not therefore worsen due to retransmission requests from the core networks when packets are dropped due to congestion in the core network. Under such conditions, only BE traffic from wired access networks is therefore assigned to the BE class in the core network.

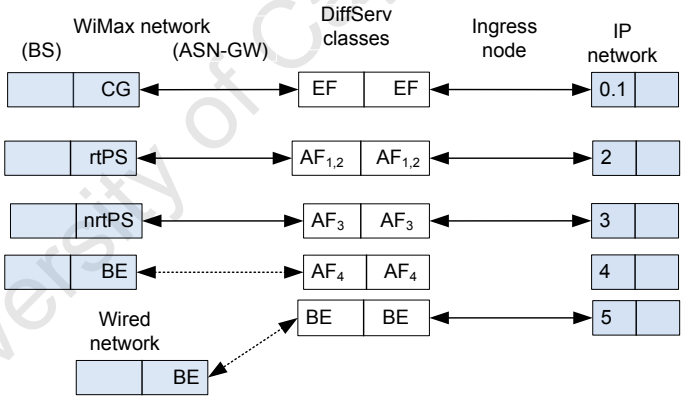


Figure 12: Proposed traffic class mapping between the access and core networks

Figure 12 above shows the proposed traffic classification and class-of-service mapping between the access and core networks. The DiffServ model is used in both networks for traffic classification. The traffic classes of the WiMax and core networks are assigned to DiffServ QoS class. By using the same DSCP in the ASN-GW and the core network ingress router, the same QoS specifications are respected in both networks.

Table 6 below shows the proposed traffic classification selected wireless and wired networks. BE traffic from wireless networks is assigned to the AF_4 while that from wired access networks is assigned to BE. Since IP core networks may also interconnect to

GMPLS network domains, BE traffic from these networks is also treated like BE traffic coming from wired network.

TABLE 6: PROPOSED QOS CLASS MAPPING FOR IP NETWORKS

DiffServ	IP	WiMax	GMPLS	UMTS	Other wired access networks
EF	0,1	CG	101	Conversation	EF
$AF_{1,2}$	2	rt-PS	001	Streaming	$AF_{1,2}$
AF_3	3	nRt-PS	010	Interactive	AF_3
AF_4	4	BE	011	BE	AF_4
AF4	4	-	-	-	AF_4
BE	5	-	000	-	BE

The interaction of WiMax and core network QoS models at the interconnection point of the networks is illustrated in figure 13. The MAC CS in the WiMax BS accepts WiMax traffic classes and forwards them to the ASN-GW where a DiffServ module reclassifies the traffic flows into DiffServ QoS classes. The traffic flows are placed onto the ASN-GW outgoing interface tagged with a DSCP value. For consistent QoS between the two nodes, the code point values must be the same. Once defined at the ingress router, the traffic is treated according to the IP QoS priority classes defined in the core network domain.

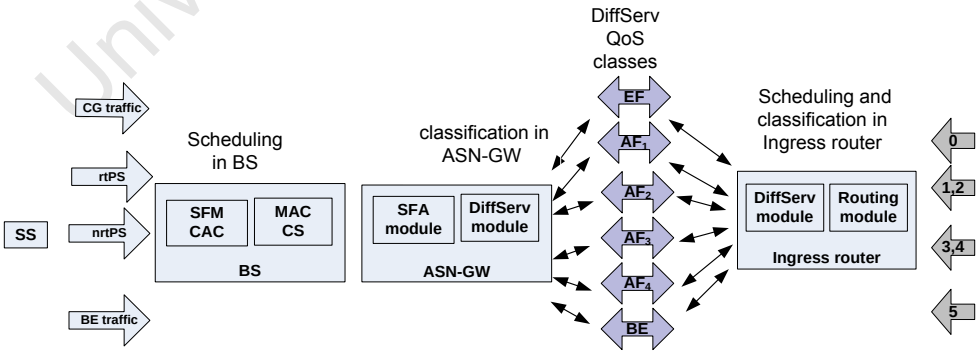


Figure 13**:** WiMax/IP/DiffServ QoS translation between network interfaces

The proposed end-to-end QoS architecture is evaluated on a NGN prototype set up by interconnecting NGN technologies used for research purposes in the CRG lab. The following section is going to cover the description of the architecture of the NGN prototype.

3.2 Design Considerations of the NGN prototype

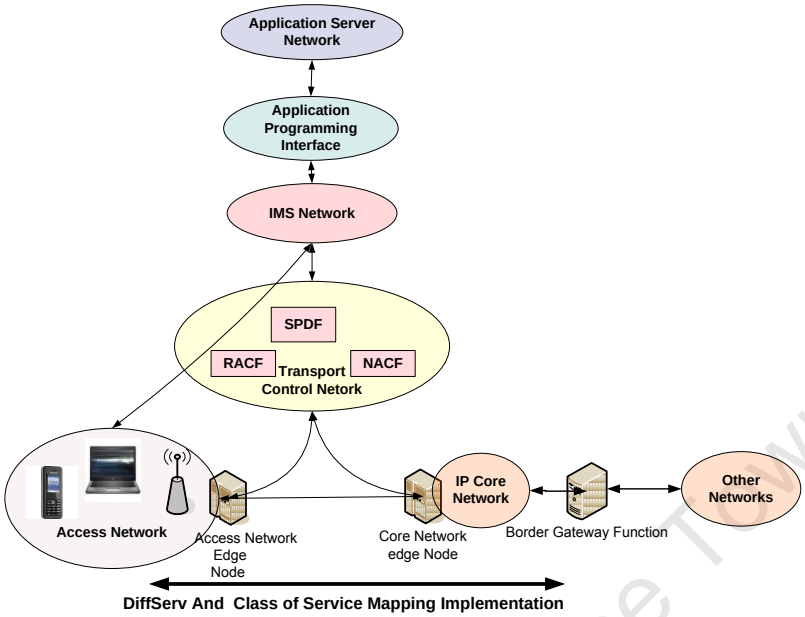


Figure 14: Typical NGN system with QoS-enabled networks

A NGN system consists of access, core, control and application/service networks. The NGN system must also have a QoS architecture that includes QoS provisioning, QoS control and QoS management. Figure 14 shows the architecture of a typical NGN system. The transport network consists of end-user equipment, access networks and a core network. The Border Gateway Function (BGF) on the transport layer enables NGN end users to have access to applications and services offered by third party Application Service Providers (ASP's). Edge nodes on the access and core networks provide internetworking functionality between different transport networks. The transport control network intermediates between the transport and control networks, enabling different transport technologies to communicate with the control network. The resource and admission control function (RACF), network attachment control function (NACF) and the related Services Policy Decision Functions (SPDF) support dynamic verification of resource availability and configuration of policy enforcement functional elements (PE-FE) on the transport networks. The IMS ensures end-to-end QoS for IP multimedia packet flows on the NGN by identifying session flows in the transport networks and prioritizing the routing of the packets.

3.2.1 The Application/Services Network

The IMS architecture incorporates application servers, which determine how services are invoked, the signalling and media required and how services on the network interact with each other on the NGN. The IMS defines three types of application servers, i.e. SIP application servers, open services architecture (OSA) servers and customized applications for mobile networks using enhanced logic service environment (CAMEL) servers. The NGN prototype presented in this thesis uses SIP application servers. An IPTV server is used in the performance evaluation of the NGN prototype to access a VoD application.

3.2.2 The control network – (IMS)

The control layer on NGN can be either Softswitch or IMS. The architecture presented in this thesis uses the IMS. The IMS solves the problems related to the delivery of IP services in NGN by identifying and separating session signalling information and media flows. Session signalling through the IMS is for the purpose of authentication, authorization and accounting. Figure 15 shows the flow of signalling information and media between the IMS, IP connectivity network (IP-CAN), User Equipment (EU) and the media and application servers. In the NGN prototype, the IMS provides the session set up and QoS signalling capabilities lacking in DiffServ. Since the IMS separates signalling and media, the proposed QoS model provides an underlying QoS model mode transfer of QoS specifications between the two transport networks.

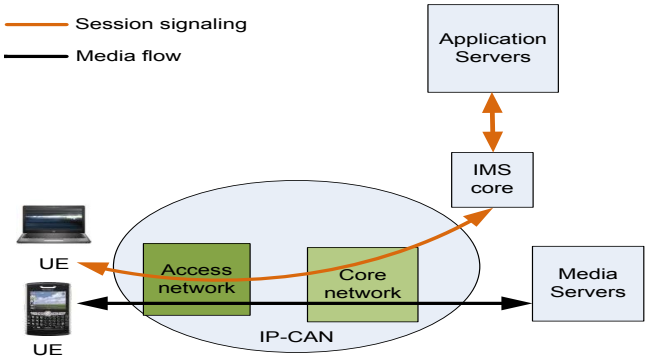


Figure 15: The IMS on an NGN system

3.2.3 The Access Network

Figure 16 shows the WiMax Access Network. The Access Service Network (ASN) represents the complete set of functions needed to provide radio access to subscriber stations. The Connectivity Service Network (CSN) provides IP services to the subscriber stations. In this thesis, the IMS and core networks represent the CSN. The IMS network provides AAA and policy functions while IP core network provides IP connectivity. On NGN systems, the ASN-GW on the ASN interconnects the ASN to the IP network through the IP edge node.

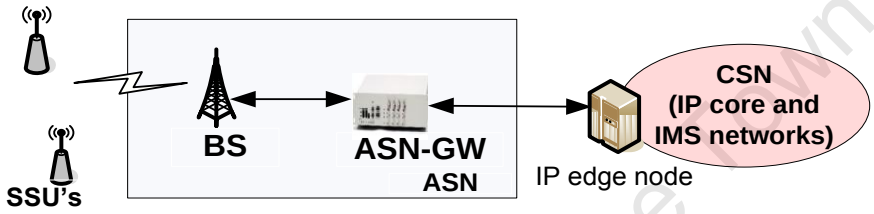


Figure 16: Interconnection between WiMax Base Station and CSN

The WiMax Access Network used in this thesis operates in point-to-multipoint configuration, Time Division Duplex (TDD) mode. In this mode, the Base Station (BS) controls connection establishment and resource allocation between itself and the subscriber stations on the network. QoS mechanisms are therefore central to the base station. Scheduling algorithms implemented in the BS guarantee the bandwidth required by each subscriber station and efficient usage of the wireless links. Admission control algorithms prevent saturation of the wireless resources and hence violation of QoS requirements of applications by restricting the number of SS simultaneously present on the network.

The BS periodically allocates bandwidth to SS's, allowing them to send their bandwidth requests and establish connections. Each connection to the BS is identified by a Connection Identifier (CID) and has specific QoS parameters, i.e. minimum reserved rate, maximum allowable latency, tolerated jitter and unsolicited polling interval and the request/transmit policy for the particular services transported on the connection. Each service flow from user equipment (UE) is associated with a connection. Service flows with the same QoS characteristics are therefore assigned to the same connection. On the

ASN, the BS is therefore able to Queue and schedule packets according to the QoS requirements requested and ensuring that the QoS requirements are satisfied. The BS therefore allows prioritizing packets transmission and hence reduces latency and jitter.

The WiMax air interface QoS mechanism can be illustrated by an example in which two services, identified by service flow identified as SFID1 and SFID2 originate from a UE. SFID1 is delay sensitive and SFID2 is best effort traffic. SFID1, through scheduling algorithms is assigned to a connection with low delay characteristics and SFID2 is assigned to a connection with BE characteristics. The low delay flows therefore use the low delay connection between the BS and SS, while the best effort flow utilizes the BE connection. The QoS system on the WiMax radio access network makes the technology a suitable access network technology for use on NGN systems and has already been adopted by the ITU-T as one of the key technologies necessary for the delivery of real-time multimedia applications like IPTV and video conferencing.

The WiMax network therefore presents a robust QoS system on the air interface capable of delivering real time multimedia applications. The proposed QoS architecture therefore provides a mechanism for translating the QoS performance achieved on the network to an IP core network.

3.2.4 The core network

The core network in NGN connects all the access layer networks to the control networks. The network must support network access control, packet routing and transfer functions, mobility management for mobile access networks and resource and network management. Core networks must also meet the QoS requirements of real-time multimedia traffic like IPTV and video conferencing in terms of throughput, delay, jitter and packet loss. Router processing power and link transmission delay are key in meeting throughput and delay requirements in the core network. Resource management to guarantee QoS involves buffer management, queuing and scheduling of packets, packet filtering, traffic classification, marking, policing and shaping. Core networks are therefore comprised of high-speed links, Gigabit switches and routers with high processing power and properly defined QoS capabilities.

The ITU-T specifies the use of IP or MPLS routers on the core network. The core network used in this thesis is based on open-source IP routers, specifically Linux routers. Open-

source Linux routers were selected because of the lower costs and richer feature sets compared to proprietary routers. Open source routers also provide high flexibility in customizing solutions. Customers are able to maintain total control of the need for new hardware and features independent of the vendor.

While open source router technology may deliver flexibility and affordable pricing, there are drawbacks associated with using such technology. Some hardware compatibility issues may arise which can stall implementation. In addition to this, the routers may not offer the levels of flexibility and usability that companies have come to expect from proprietary solutions. Another drawback is that, while the open source community provides free support, there is need for in-house skills to provide timely support, and it is not easy for companies to maintain a pool of such skills.

In this thesis, Linux machines run as QoS-enabled routers. Installation of routing and traffic control modules enables the Linux kernel to perform routing and traffic control. Activation of a DiffServ module in the kernel also enables the routers to perform traffic classification. This enables the routers to support QoS translation between the access and core network domains and hence achieve consistent QoS between the two networks. Use of Fast-Ethernet network interface cards on the Linux routers ensures the high throughput required in NGN core networks. The short distances between the routers ensure low negligible transmission delay.

Advanced routing and traffic control features in Linux routers make it possible to provide QoS to NGN multimedia applications. These features include Netfilter with Iptables for firewall capabilities, traffic control next generation (tcng) for QoS implementation [51] and routing software like Quagga [5], for dynamic routing in the IP core network. Iptables is a user space program that defines rules and commands in conjunction with the Netfilter kernel module, which evolved from Ipfwadm and ipchains, used in earlier versions of Linux [51]. It simplifies the process of configuring filtering rules in Linux. Netfilter supports packet filtering, Network Address Translation (NAT) and packet mangling i.e. marking of packets in the kernel using mechanisms such as DiffServ or 802.1p. Netfilter is also an improvement from the NetLink functionality in the Linux kernel.

The tcng module extends traffic control in Linux to a user interface, making the configuration of the system more flexible and solving some of the problems related to traffic control in Linux [51]. These modules enable Linux routers to handle real time

multimedia services that are characteristic of NGN traffic.

The Quagga routing package [5], is implemented on the Linux machines to enable dynamic routing on the core network. Quagga is an open source routing software that provides TCP/IP routing services. The package has three dynamic routing protocols: Routing Information Protocol (RIP), Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP). Dynamic routing on the core network enables automatic route discovery for packets to and from the client machines. Servers on the CRG LAN host DNS and DHCP services. The core network is a single DiffServ domain [52] with three routers.

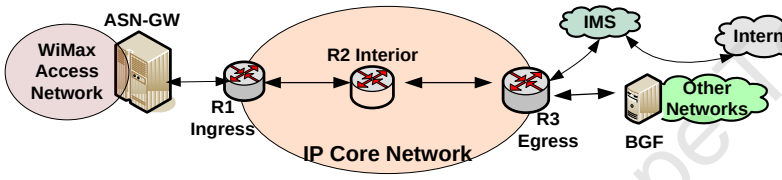


Figure 17: The core network based on a 3-router DiffServ domain architecture

Figure 17 shows a DiffServ domain consisting of three nodes, i.e. the ingress, egress and interior routers. The ingress router R1 at the edge of the network connects the domain to the access network. R3 is the egress router and it connects the DiffServ domain to other network domains through the border gateway function (BGF). The router is also the interconnection point for the access network to the IMS and Internet.

Implementation of DiffServ in Linux in conjunction with the tcng module provides a full set of traffic conditioning modules enabling network administrators to setup any type of DiffServ domain. DiffServ modules in Linux include a marker, classifier, service handlers for Expedited Forwarding (EF) and Assured Forwarding (AF) and several queuing disciplines. Edge routers perform traffic marking and classification. Traffic is marked according to the source IP address, traffic type or load. Once packets are marked, the core routers perform per-class traffic management.

Figure 18 shows the DiffServ traffic conditioning technique in a Linux-based ingress router [52]. A marker writes specific DiffServ Code Points (DSCP) into the IP header of the packets. Interior routers use the DSCP value in making decisions on how to treat the marked packets. The classifier uses DSCP to place traffic into appropriate queuing

disciplines. Service handlers enable placement of traffic into the different DiffServ per-hop-behaviours. Network control traffic is placed in the Network control service handler.

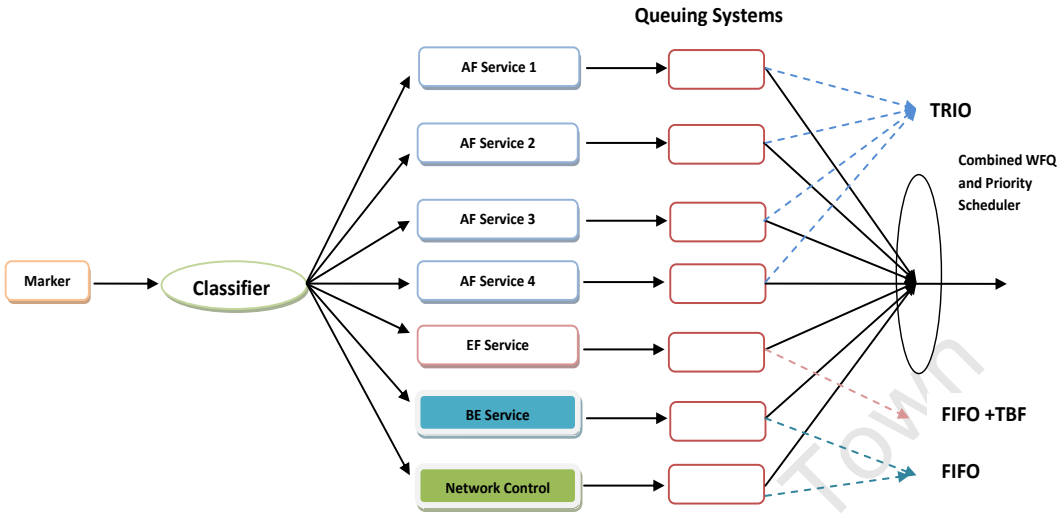


Figure 18: DiffServ traffic conditioning in an ingress router

Notes:

AF – Assured Forwarding
EF – Expedited Forwarding
BE - Best Effort
In and **O**ut
FIFO - First In First Out
TBF – Token Bucket Filter
WFQ – Weighted Fair Queuing
TRIO - **T**hree state **R**ED (Random Early Detection) with

Schedulers arrange or rearrange the queued packets for output. A scheduler ensures that each traffic class receives the appropriate bandwidth capacity and does not exceed the data rate assigned to the traffic flow. The ingress router handles micro-flows of traffic, aggregates the traffic into a group of flows with the same QoS requirements and forwards the traffic to the interior routers.

The egress router is implemented at the egress of the DiffServ domain. It ensures that the behaviour of the traffic leaving the domain complies with the behaviour expected by the next network domain. The router ensures this by applying traffic conditioning mechanisms like shaping and dropping.

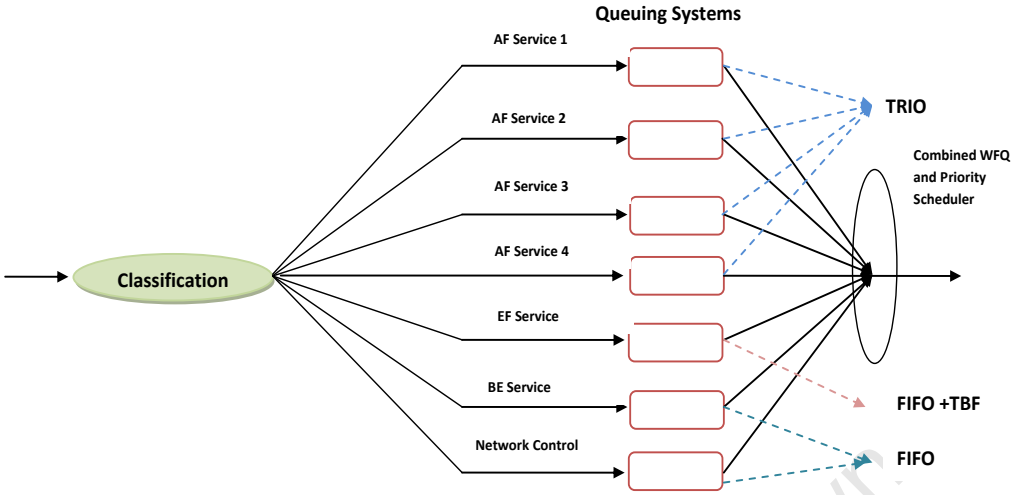


Figure 19: DiffServ implementation in interior router

Notes:
AF – Assured Forwarding
EF – Expedited Forwarding
BE - Best Effort
TBF – Token Bucket Filter
WFQ – Weighted Fair Queuing
TRIO - **T**hree state **R**ED (Random Early Detection) with **I**n and **O**ut
FIFO - First In First Out

Figure 19 shows the traffic handling in an interior router based on DiffServ. In the interior routers, traffic flows are aggregated into any one of the DiffServ QoS classes depending on the QoS requirements. Traffic therefore belongs to either the AF, EF or BE QoS class. Unlike the edge routers, the interior routers do not handle micro-flows of traffic. Minimal traffic conditioning is done to detect any mis-configuration by the ingress router [52]. Service handlers are removed from the architecture of the interior routers, thus simplifying traffic forwarding inside the domain.

The Linux routers therefore create a DiffServ domain with a QoS system that is different from that of the access network. The proposed QoS architecture enables the QoS specifications satisfied on the access network to be maintained in the core network, thus achieving consistent QoS between the two networks.

3.3 Discussion

This chapter introduced end-to-end QoS as one of the key challenges faced by network operators when rolling out NGN systems. The details of the proposed QoS architecture were discussed. It was discussed how consistent QoS can be achieved between networks

with different QoS systems, by applying the concepts of traffic classification and class-of-service mapping, which are important characteristics of the QoS architecture proposed in this thesis. Compared to other end-to-end QoS solutions, the proposed QoS architecture is simple and effective for its purpose within NGN systems. The focus of this thesis is centred on ensuring end-to-end QoS guarantees for NGN applications traversing network domains with different QoS systems. With simple static traffic classification and class-of service mapping, the proposed QoS architecture provides seamless service continuity to users of NGN services.

The important properties of the proposed QoS architecture may be summarised as follows:

- WiMax QoS classes are statically translated to DiffServ QoS classes in the ASN-GW of the WiMax network. DSCP values are used to translate the applications QoS requirements defined in the WiMax QoS specifications to DiffServ QoS classes.
- The differential QoS treatment achieved in the access network is therefore transferred to the DiffServ QoS model.
- The QoS requirements of traffic leaving the WiMax network are therefore the same as those defined on the air interface.
- Upon entering the core network, the proposed QoS model ensures the DSCP assigned to the received traffic are not changed. This is achieved by setting the rule in the DSCP module of the router such that the DSCP of the traffic remain the same.
- The core network, configured as a DiffServ domain, therefore satisfies the QoS requirements of the applications, as specified in the access network.
- To cater for the unpredictable behaviour of TCP/IP in case of network congestion, in IP networks, the proposed QoS architecture places BE traffic from wireless networks in the AF behaviour aggregate of the DiffServ model. This way, BE traffic from the already compromised air interface is not dropped first in the case of congestion in the core network; together with other BE traffic from wired access networks.

Since the proposed QoS architecture is evaluated on a NGN prototype network, the chapter concludes with the description of the architecture of a NGN system. Special attention is given to the access and core networks, which are involved in the evaluation of the proposed QoS architecture.

The following chapters address the evaluation of this work. The performance metrics of the NGN prototype are compared with those specified by the ITU-T.

Chapter 4

Evaluation framework of the NGN prototype

4.0 Introduction

This chapter discusses the design of the platform used to evaluate the performance of the proposed QoS architecture. With the requirements of the evaluation framework in mind, the implemented network topology, components and modules necessary to build the core network are introduced and discussed. At the onset, the objectives of the test bed are discussed. This is followed by a description of the test bed requirements and the reasons for the choice of the evaluation platform. A standard WiMax network is used in the test bed. The equipment is based on proprietary software. Therefore, no software modifications can be done. The work described in this chapter is therefore limited to service provisioning on the access network, which includes traffic classification and class of service mapping via a Network Management System (NMS). Work on the control and application server networks is also limited to interconnecting the network components to the core network. The rest of the chapter describes the installation of routing and QoS modules required for the Linux machines to emulate routers.

4.1 Objectives of the NGN test bed

For network operators to provide guarantees that an IP network will meet the performance requirements of NGN services and applications, the network QoS metrics should comply with performance standards of IP networks. The QoS metrics are specified in the ITU-T recommendations Y.1541 [3] for general next generation network services, and those for IPTV and video on demand services are specified in ITU-T G.1080 [6]. If they are not, the network should be engineered to meet these performance objectives. To ascertain the QoS capabilities of the prototype, and ensure that the network can meet the QoS requirements of the multimedia applications, performance evaluation tests must be carried out on the access, core and end-to-end segments of the network. The objectives of the performance evaluation tests are listed below.

1. To determine the inherent values of delay, jitter, packet loss and throughput of the access and core networks. These values provide an indication of the QoS capabilities of the networks and their ability to meet the minimum QoS requirements for NGN

access and core networks. The proposed QoS architecture uses the QoS systems of both the access and core networks. It is therefore critical to establish that the access and core network QoS systems meet the minimum QoS requirements of the NGN prototype network.

2. To determine the end-to-end values of delay, jitter, throughput and packet loss of the internetworked access and core networks. The end-to-end QoS values give an indication of the performance of the proposed QoS solution within the NGN prototype. The obtained values must also be within those specified for a NGN.
3. End user hosts must be able to access applications in the form of IPTV video on demand, video streaming and data files in the form ftp files. These applications have specific values of delay, jitter, packet loss and throughput that a network must met for their successful delivery. The evaluation tests establish the ability of the NGN prototype to deliver these applications between the end user hosts and media servers at the application layer.

4.2 Topology and QoS requirements of the evaluation test bed

To achieve the objectives of the evaluation test bed, the network topology needs to be clearly defined. The NGN prototype should be composed of an access and core network, a control network and an applications/services network where applications are hosted. End user hosts must also be connected to the WiMax subscriber station to providing users an interface to the NGN system. The access and core networks should be QoS-enabled. The WiMax network used in this thesis has in-built QoS modules, which allow translation of the WiMax QoS classes to higher layer QoS systems. The DiffServ module is one such module that is used to translate WiMax QoS classes to DiffServ QoS classes, used at the IP layer. The core network is based on Linux machines, which emulate routers. A QoS system must be installed in each of the routers to build the QoS-enabled core network. Interconnection of the two networks therefore presents platform for implementation of the proposed QoS architecture, which requires integration of the access and core network QoS systems.

End users should be able to access media in the applications/services network from any of the subscriber stations in the access network. The core network routers provide the routing capability and IP connectivity required to access the Internet via the CRG Lab

LAN. The network must also be able to deliver NGN services, in the form of IPTV video on demand, video streaming and data without degradation in the QoS requirements of any of the applications on the access and core.

Table 7 below [3] shows the performance requirements for selected multimedia applications, namely IPTV, video streaming and VoIP specified by the ITU-T. Delay, jitter, packet loss and throughput are the QoS metrics that describe packet transfer characteristics for IP networks.

TABLE 7: QOS PERFORMANCE OBJECTIVES FOR SELECTED MULTIMEDIA APPLICATIONS

QoS Metrics	Applications		
	IPTV	Video Streaming	VoIP
Throughput	≥10Mbps	≥2Mbps	≥64Kbps[1]
Delay	≤100ms	≤400ms	≤150ms
Jitter	≤50ms	≤50ms	≤10ms[2]
Packet Loss	≤0.01%	≤0.1%	<1%

Notes:
[1]Depends on the codec used.
[2]Not the recommended value by ITU-T but most network operators specify this maximum value on their networks.

The minimum packet transfer rate, i.e. throughput, must satisfy the minimum throughput requirement for IPTV. The subscriber stations used on the NGN prototype must therefore have a minimum throughput of 10Mbps for the network to be able to deliver the IPTV application to end users. The maximum values of delay and packet loss must also be lower than that defined for IPTV since the application imposes the least delay requirement on the network. Large values of jitter affect VoIP traffic and result in the audio message being difficult to comprehend. The maximum tolerable value of jitter for the network must therefore be less than 10ms. The NGN prototype must therefore satisfy the QoS requirements of all the applications on the network, despite their varying QoS requirements.

The ITU-T also defines performance objectives for IP networks aimed at providing performance guidelines to network operators. Table 8 [6] shows the performance objectives for traffic assigned in the six IP QoS classes. The performance objectives describe acceptable values of delay, jitter and packet loss for applications in the various IP classes of services – classes 0 to 5 in the table for NGN systems. The experiments

performed on the NGN prototype involve obtaining values of the QoS metrics and comparing them to these performance objectives. The values of the QoS metrics take into account the QoS requirements of the traffic in each of the IP QoS classes.

TABLE 8: ITU-T QOS CLASS AND PERFORMANCE OBJECTIVES FOR IP NETWORKS

Network QoS Metrics	Class0	Class1	Class2	Class3	Class4	Class5 Unspecified (U)
IPTD[1]	100ms	400ms	100ms	400ms	1s	U
IPDV[2]	50ms	50ms	U	U	U	U
IPLR[3]	$1x10^{-3}$	$1x10^{-3}$	$1x10^{-3}$	$1x10^{-3}$	$1x10^{-3}$	U

Notes:

1. IPTD – IP time delay (Network Delay) 2 IPDV – IP delay variation (Network Jitter)
2. IPLR – IP loss ratio (Packet Loss) 4 U - Unspecified

According to the ITU-T Rec. Y.1541, throughput cannot be specified because the metric is dependent on a number of variables. These include delay, available bandwidth, signal-to-noise ratio and hardware limitations. Values of throughput obtained in the tests conducted on the NGN prototype do not therefore take into consideration these variables. Similar values can only be obtained if the experimental conditions are similar. The value of the throughput obtained on the network can therefore be used to determine whether the network throughput is sufficient to deliver the applications under consideration. An indication of the throughput values for IPTV video on demand, video streaming and VoIP are discussed in the previous section.

The NGN system must be tested prior to any services being run on the network. These tests, referred to as link quality tests, give an indication of the QoS capabilities of a network. The delay and packet loss must be lower than those of IPTV since the application imposes the minimum requirements of the metrics. The measured value of jitter must be less than 10ms to enable the network to deliver the VoIP application. Each of the subscriber station must have a minimum throughput of 10Mbps to enable delivery the IPTV application. Iperf, an open source network-testing tool is used in this thesis to obtain in real time, the actual values of these metrics on the NGN prototype. The results obtained are used to determine if the NGN prototype meets the performance requirements of selected multimedia services i.e. IPTV video on demand (IPTV VoD) and video streaming. The results obtained also provide valuable information to researchers on the performance of NGN platforms using WiMax, DiffServ and IMS technologies. Chapter 5 discusses the tests carried out to determine the values of delay, jitter, packet loss and

throughput of the network and the ability of the network to deliver integrated multimedia services.

4.3 Choice of Platform

The network segments of the NGN prototype is a mix of hardware-based and software-based options. The cost of hardware has always been identified as a limiting factor in hardware-based research test beds; however, recently we have seen a decline in equipment costs. Proprietary routers like Cisco are however still relatively expensive. Linux machines, configured to emulate routers, are therefore used in the core network of the NGN prototype. Routing and QoS capabilities are enabled on the Linux machines by installing open source routing and QoS software modules. The routing and QoS software packages used are freely available on the Internet, thus bringing down the cost of the routers to that of the hardware only. Another reason for using Linux routers on the NGN prototype is that the routers can be customized to work as MPLS routers by installing MPLS modules. This flexibility is not available in proprietary routers, which can only be either IP or MPLS.

The core network of the NGN prototype is based on IP routers, which can also be configured to emulate MPLS routers as discussed above. IP routers were used because MPLS routers communicate with Network Elements that are also MPLS-based. The WiMax system used on the NGN prototype as access network is Ethernet-based and does not support MPLS. Interconnecting an MPLS core network to a WiMax network would be complex and require an MPLS-enabled ASN-GW.

The access network is based on a point-to-multipoint WiMax micro-base station system manufactured by Alvarion. Since WiMax roll out is still underway, use of the system on the test bed provides a real life scenario for evaluating the performance of the technology.

The UCT IMS client [45], which works with the Fraunhofer FOKUS Open Source IMS core [40], is also implemented on Linux machines. The network is a result of elements created by researchers in the UCT CRG lab. The platform has the flexibility to allow researchers to download and freely modify the IMS client to suit their own research requirements. The platform is used in this thesis as a control layer for the NGN prototype.

SIP-based application servers connected to the IMS core are used as application layer Network Elements of the prototype. IMS clients are installed on user machines to emulate

IMS user agents. The user agents register on the IMS to be able to access services in SIP-based media servers. User machines without IMS clients can also access media servers as client-server entities and no IMS services like authentication are used in connection set up. Session set up delay between IMS clients and application servers on the test bed is one of the QoS characteristics that is affected by the inherent end-to-end delay of the access and core networks. It is therefore used to evaluate the performance of the network.

The framework of the NGN prototype consists of all the network components of a NGN system. Figure 20 shows the NGN prototype system used in this thesis. This diagram will be used as reference to the NGN prototype throughout chapters 4 and 5. The WiMax network is an Ethernet-based IP access network. The network is therefore equipped with a DiffServ QoS module. The proposed QoS architecture maps WiMax QoS classes to those of the IP-based core network to achieve consistent QoS between the two networks. The architecture only uses the DSCP specified in DiffServ modules already installed in all IP-based Network Elements. QoS models based on signalling protocols, e.g. NSIS are often used in the literature. These require installation of QoS modules in the transport Network Elements. Policy-based QoS models require the installation of policy servers, policy decision and policy enforcement functions both in the access and core networks. The proposed QoS architecture is implemented on the NGN prototype without the need to put the routers out of service. Network performance is therefore improved without additional hardware costs.

UE_1 to UE_3 in figure 20 represent the end users on the NGN prototype. The access and core networks represent the transport networks required in NGN systems between the end users and applications and services networks. CAT-5 cable was used to interconnect the Network Elements but in real-world cases, high capacity microwave links or fibre optic links are used. On the access network, translation of WiMax QoS classes to DiffServ QoS is performed statically on the WiMax Micro-BS element via the NMS. Since the access network is an Ethernet-based IP network, the micro-base station indoor unit handles the IP functionality of the access network. The DiffServ module is therefore resident in this network element. The NMS provides an interface for defining service QoS profiles and assignment of DSCP to services. The core network routers are accessed via a single

monitor and keyboard through a KVM switch.

The following section describes how the two QoS systems are bounded together to provide consistent QoS between the two transport networks.

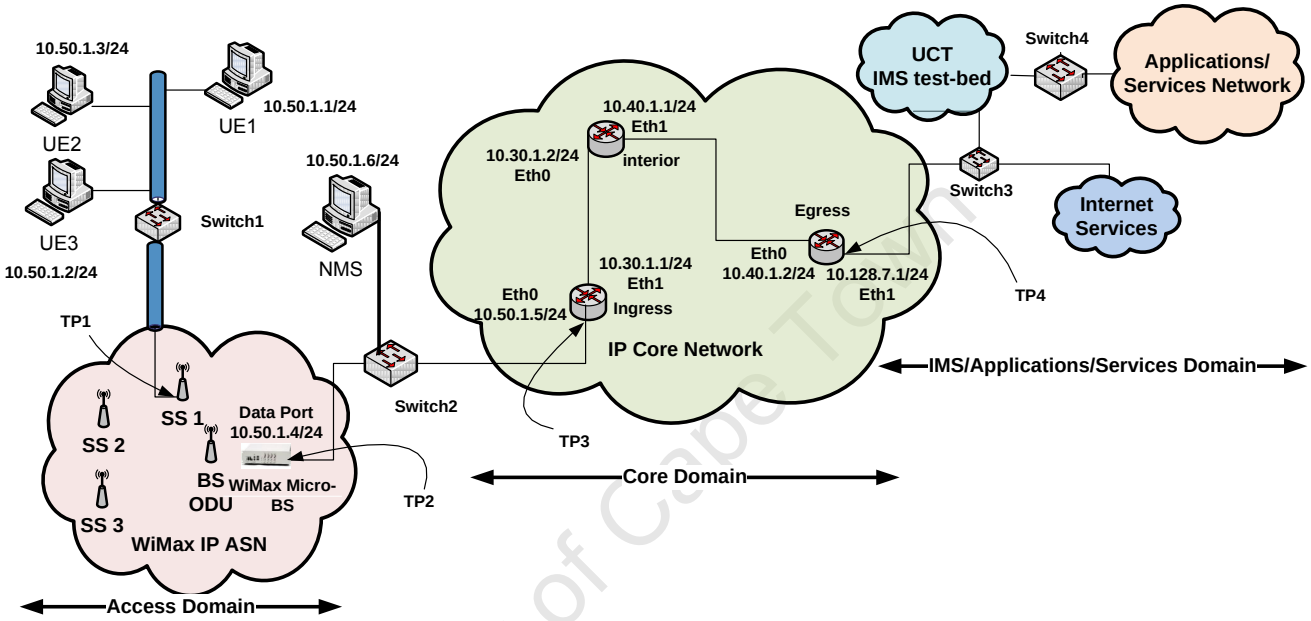


Figure 20: Evaluation framework showing the interconnected NGN technologies

Notes:

SU: Subscriber Unit BS: Base Station ASN: Access Service Network IP: Internet Protocol
IMS: IP Multimedia Subsystem UCT: University of Cape Town NMS: Network Management System UE: User Equipment
ODU: Outdoor unit TP: Test Point

4.4 Integration of the access and core network QoS systems

This section discusses the work done on the access and core networks to integrate the access and core network QoS systems. The implementation of the proposed QoS architecture follows the processes outlined in section 3.2. Traffic classification and class of service mapping on the WiMax network is implemented in the micro-base station indoor unit and on the core network, this is implemented in the ingress router.

The WiMax network used is a proprietary solution manufactured by Alvarion and is Ethernet-based. The core network is based on Linux machines emulating routers by

installing routing and QoS modules. The work done in the core network is discussed in more detail in the following section. The technical details of the WiMax network set up are given in appendix B.

The core network consists of Linux routers with traffic control capabilities enabled. Three machines are configured as ingress, interior and egress routers, forming a DiffServ domain as discussed in chapter 3. The ingress router connects to the access network and the egress router to the services networks, i.e. IMS, Internet and application servers. Each of the machines is equipped with two network interface cards assigned with IP addresses in different subnets to enable them to forward traffic between the interface cards. This is illustrated in figure 21. The ingress router Ethernet interface 0, Eth0, is in the same subnet as the user equipment and the access Network Elements, i.e. user machines. The following paragraphs describe the routing, QoS and firewall modules installed in the routers.

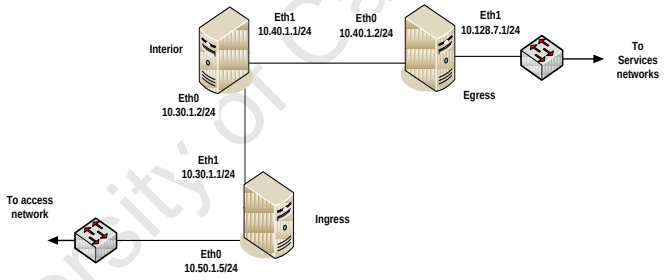


Figure 21: Core Network Implementation

4.4.1 Quagga routing software

The Quagga routing software [5] is installed on all machines that perform routing. The software provides implementation of three dynamic routing protocols; the Border Gateway Protocol (BGP), Open Shortest Path First (OSPF) and Routing Information Protocol (RIP). It can be used on Unix-like platforms, e.g. FreeBSD, Linux, Solaris and NetBSD. The Quagga architecture consists of a core daemon called zebra, which acts as an abstraction layer to the underlying Linux kernel. It presents the Zserv API over a Linux or TCP stream to Quagga clients. The Zserv clients implement a routing protocol and communicate routing updates to the zebra daemon [5].

Zebra provides TCP/IP based routing services with support for dynamic routing protocols.

Zebra has an interactive user interface for each routing protocol and supports common client commands. Due to this design, new routing protocol daemons can be easily added. The Zebra library can also be used as a program's client user interface. The Quagga routing suite on the core network routers provides a reliable and efficient routing system.

In this thesis, RIP and Zebra daemons are activated while OSPF and BGP daemons are disabled. RIP is chosen over BGP and OSPF due to the small size of the network. The RIP has low processing over-head compared to other two protocols. Moreover, it is felt that the high performance of the other protocols would not present any advantage on the network. If additional routers were connected, consideration could be given to activate either BGP or OSPF for dynamic determination for routes the bigger network. The Avahi daemon, well known to take up to 50% of the CPU processing time in Linux machines, was disabled to improve the performance of the routers since evaluation tests included video applications which also consume CPU processing time of the routers. Details of the commands used to configure IP addresses and IP forwarding in the routers are given in appendix C.

4.4.2 Traffic control next generation (tcng)

Tcng provides a configuration language that can be used to easily configure traffic control systems in Linux. The tcng language is similar to programming languages like C, Perl, or Java and this makes it simple to implement if one is familiar with any one of these languages. The tcng software consists of two major components: the traffic control compiler (tcc) and the traffic control simulator (tcsim). Tcng translates configuration scripts into a multitude of output formats used to configure traffic control subsystems. The tcng command line tool therefore provides a more user-friendly interface for performing traffic configuration operations on the Linux kernel.

The Linux version used in the core routers supports both ‘tcc’ and ‘tcsim’, tcng tools. The version also supports dsmark, the DiffServ marking mechanism. Details of how to enable tcng and all the tools required to support DiffServ QoS are outlined in [53].

The following section discusses how traffic classification and class of service mapping is achieved on the NGN prototype.

4.5 Traffic classification and class of service mapping

The proposed QoS architecture maps WiMax QoS classes to IP QoS classes in the core network. DiffServ is the common model between the two QoS systems. To implement the QoS architecture, it was necessary to create service classes in the access network by creating service and QoS profiles in the WiMax micro-base station indoor unit via the NMS (refer to figure 20). The system offers VLAN ID, 802.1p and DiffServ as classification parameters and the latter was selected since it is at the centre of this work. IP data was selected as the traffic type, since the applications selected for use on the network fall into this category. One of the parameters of the QoS profiles is the DSCP values. They define how the DiffServ module handles the traffic in terms of QoS requirements.

The tc (traffic conditioner) tool is used to set up DiffServ parameters in Linux routers. Details of how to set up DiffServ and create traffic classes in Linux are given in appendix C. The proposed QoS architecture focuses on traffic classification and class of service mapping at the network boundaries of the access and core networks. Details of the implementation of a DiffServ QoS domain in Linux routers are widely available. An example is the detailed description given in [54].

TABLE 9: CLASSIFICATION OF APPLICATIONS INTO DIFFERENT QOS CLASSES

Service Profile	WiMax	Core Network	DIFFSERV Equivalent Classes	Equivalent IP QoS Classes
IPTV VoD	CG	46	EF	1
Video Streaming	RT	10	AF_{11}	2
BE Data	BE	34	AF_{41}	4

The values assigned to the traffic used in this thesis are shown in Table 9. On the WiMax network the DiffServ QoS classes used were assigned the numerical values 46, 10 and 34 for EF, AF_{11} and AF_{41}. On the core network, the same numerical values were assigned to the predefined traffic flows carrying the three services. The specific statements used on the access and core networks to create the proposed class of service mapping are given below. Full details of the implementation on both networks are given in appendices C and D respectively.

Core network:

 # tc class change dev eth0 parent 1:0 classid 1:1 dsmark \ mask 0 value 0x46 **(IPTV VoD)**
 # tc class change dev eth0 parent 1:0 classid 1:2 dsmark \ mask 0 value 0x10 **(Video streaming)**

#tc class change dev eth0 parent 1:0 classid 1:3 dsmark \ mask 0 value 0x34 **(Data service)**

Access network:

- *Priority marking mode: DSCP* **(IPTV VoD)**
- *Priority marking value: 46*
- *Priority marking mode: DSCP* **(Video streaming)**
- *Priority marking value: 10*
- *Priority marking mode: DSCP* **(Data service)**
- *Priority marking value: 34*

4.6 Discussion

This chapter discussed how the architecture to support end-to-end QoS was implemented in the NGN prototype. The design of the test bed resembles a real-world QoS-enabled NGN system. The WiMax network presented a QoS-enabled NGN access network required to build up the proposed QoS architecture. The core network is based on Linux machines emulating IP routers. QoS and DiffServ modules were installed and enabled in the routers to create a QoS-enabled network domain. The IMS and application server networks developed by other researchers in the UCT CRG lab were interconnected onto the core network to present a complete NGN system. Application servers are used in the NGN prototype to host media accessed by end user machines.

To evaluate the performance of the proposed QoS architecture, performance metrics of the NGN system, i.e. delay, throughput, packet loss and jitter are measured and compared to those defined by the ITU-T for NGN systems. To determine the capability of the NGN prototype to deliver real-time NGN applications, IPTV video on demand, video streaming and data services are run on the network. Performance objectives of these applications, as defined by the ITU-T, are given.

The proposed QoS architecture uses the concepts of traffic classification and class of service mapping. The last section discusses how these concepts are used to classify traffic and map services on the architecture. The following chapter presents the results of the experiments performed on the NGN prototype. An analysis of the results obtained on each of the experiments is also given.

Chapter 5

Evaluation Results and Analysis

5.0 Introduction

This chapter presents the performance evaluation of the proposed QoS architecture within the NGN system presented in chapter 4. Each aspect of the work is evaluated individually and is presented in separate sections. Values of delay, jitter, throughput and packet loss are explored quantitatively and used to evaluate the performance of the proposed QoS architecture within the NGN prototype. The first section evaluates the performance of the access and core networks separately. The second section evaluates the performance of the proposed QoS architecture within the NGN prototype. The final section presents experienced performed on the test bed to evaluate the capability of the network to deliver real-time applications.

The NGN prototype presents a platform where the performance evaluation tests of the proposed QoS architecture can be controlled and all experiments repeated with consistency. Quantitative and qualitative results are obtained and compared to those defined by the ITU-T for NGN systems and applications. Qualitative analysis of the performance of the NGN prototype is based on subjective QoS descriptions of applications e.g. *"staticky, warbley, muffled, clipped"* for audio and *"blurry, jerky, blocky, busy, blotchy"* for video [25]

The following section presents a description of link quality tests used to determine the base line values of the QoS performance metrics. The objective of the experiments is to ensure that the NGN prototype meets the performance requirements of a NGN system as defined by the ITU-T.

5.1 Link quality tests

This set of experiments establishes the actual values of the delay, jitter, packet loss and throughput of the access, core and end-to-end networks. The values obtained are compared with those defined by the ITU-T for a NGN system. Iperf, a network performance measuring tool described in the previous chapter is used to obtain the values in real-time. Iperf can create TCP and UDP data streams and measure throughput, jitter

and packet loss experienced by the traffic. Values of delay are obtained using the 'ping' test tool. Two hosts running Iperf determine the boundaries of the link whose quality is being measured. In the experimental test bed, test points TP1 to TP4 (referred to figure 20 in chapter 4) are used as the boundary test points. Results of the values appear as output after running the Iperf test commands.

Values of jitter and packet loss are obtained using Iperf UDP tests. Network throughput on a particular segment is measured using Iperf TCP tests. Two hosts are connected to the boundaries of the network with one host set as a client and the other as a server. By default, an Iperf client connects to an Iperf server on port 5001. The throughput displayed is from a client to a server.

The following sections describe the tests carried out on the individual access and core networks. The results obtained are also given. An analysis of the results obtained for each network segment is given. Details of the commands used to obtain the results in real time using Iperf are given in appendix D

5.1.1 Access network performance tests and results

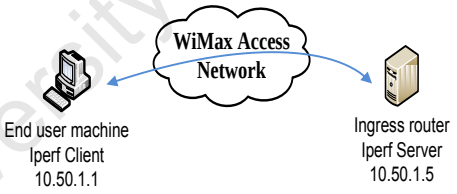


Figure 22: Set up for access network tests

Notes:
Link Bandwidth: 12Mbps
Signal Strength setting: 10db

Figure 22 shows the set up used to obtain performance values. i.e. delay, jitter, packet loss and throughput for the access network. The ingress router and an end-user host are connected in client/server mode. The end-user host connects onto the access network through one of the subscriber stations. The results are recorded at intervals of 90 seconds. During each test period, packets of data are transferred between the two Iperf hosts. Iperf calculates the performance values and the results are obtained in real time. An

average value for the 90-second interval is manually calculated and recorded for each of the QoS metrics. Table 10 shows the results obtained for this segment of the network.

TABLE 10: ACCESS SEGMENT LINK-QUALITY TEST RESULTS

Parameter	Value obtained
Throughput	9.98 Mbps
Delay	31ms
Jitter	2ms
Packet Loss	0.31%

- **Access network Throughput**

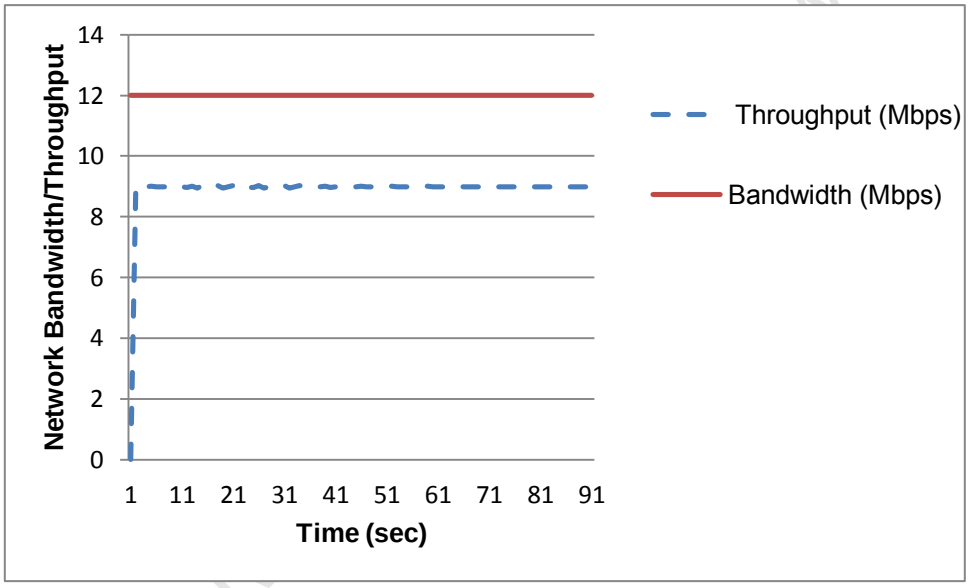


Figure 23: Access network throughput over 90 seconds period

The WiMax solution used in the experiments is designed to provide bandwidth of 12Mbps between the base station and a subscriber station. The network throughput is therefore not expected to be more than 12Mbps. If the required throughput is not specified when running Iperf, the tool gives a maximum value of the network throughput. The average maximum throughput for the access network obtained in the network is 9.98Mbps. This is a variation of 2.02Mbps from the available bandwidth, and about 17% of the link bandwidth. This could be attributed to background traffic in the form of signalling messages between the base station and the subscriber station. Figure 24 shows a screen capture from Alvaricraft, the network management system for the WiMax network. It shows

periodical traffic on the network of up to 100 bytes. This traffic appears as spikes on the network throughput shown in figure 24.

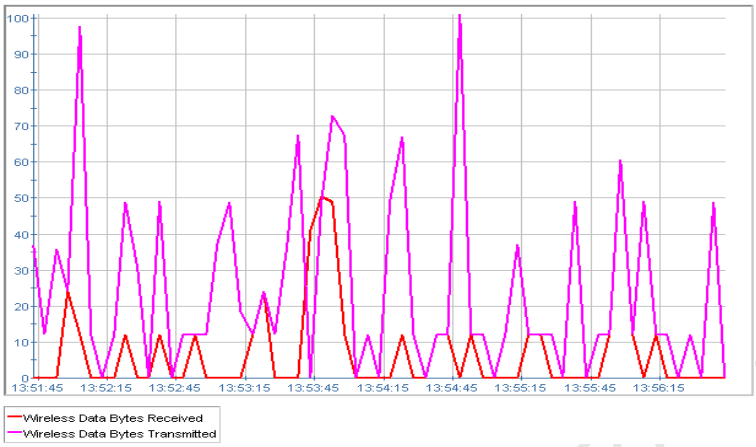


Figure 24: Access network background traffic

The IEEE 802.16 standard specifies a bandwidth request and grant mechanism for allocating bandwidth to subscriber stations. The signalling mechanism could explain the background traffic and takes up a substantial percentage of the available bandwidth. This affects network throughput as the CPE processing time is shared between processing actual data and signalling traffic. In [65], this problem is explained and a proposal is presented to improve the signalling mechanism, and hence the performance of the networks.

- **Effect of Attenuation on throughput**

Figure 25 illustrates the effect of using different values of attenuation on the link between the subscriber stations and the base station. Higher attenuation, which emulates bad wireless connection in a practical environment, resulted in lower throughput. This illustrates the susceptibility of the link to poor conditions in the transmission medium. All evaluation results were therefore taken with the attenuation set to 10dB since at this value, the network reached its optimum performance level.

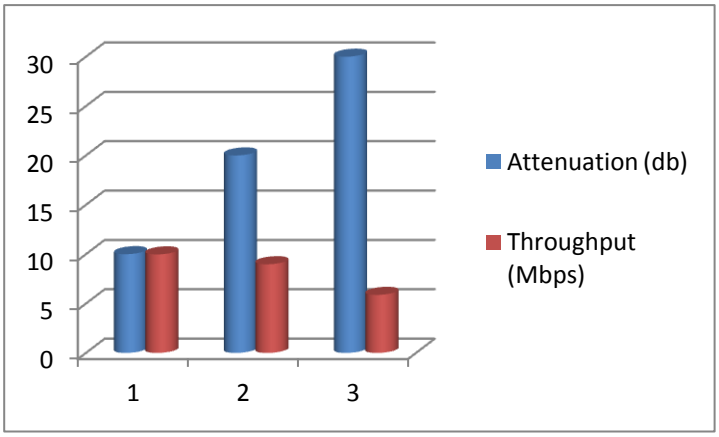


Figure 25: Effect of Attenuation on throughput

- **Access network delay and jitter**

The average value of the delay on the access network is 33.69ms. The value is well within the required value for the selected applications in table 7, as well as those specified for IP networks by the ITU-T. Delay on transmission networks is due to propagation delay, queuing delay in Network Elements and processing delay. On WiMax access network, this is attributed to processing and queuing delay in the base station and processing delay in the subscriber station. The propagation delay is therefore negligible because of the distance between the subscriber station and the base station. Figure 26 illustrates delay in milliseconds over a period of 90 seconds. The 'allowed delay' is the recommended value for IP access networks. The values plotted in red indicate the values obtained on the WiMax network. The values are high due to the connection establishment traffic on the network. Once the link is established, the delay varies between 20ms and 40ms, with a calculated average of 33.69ms shown the results table above.

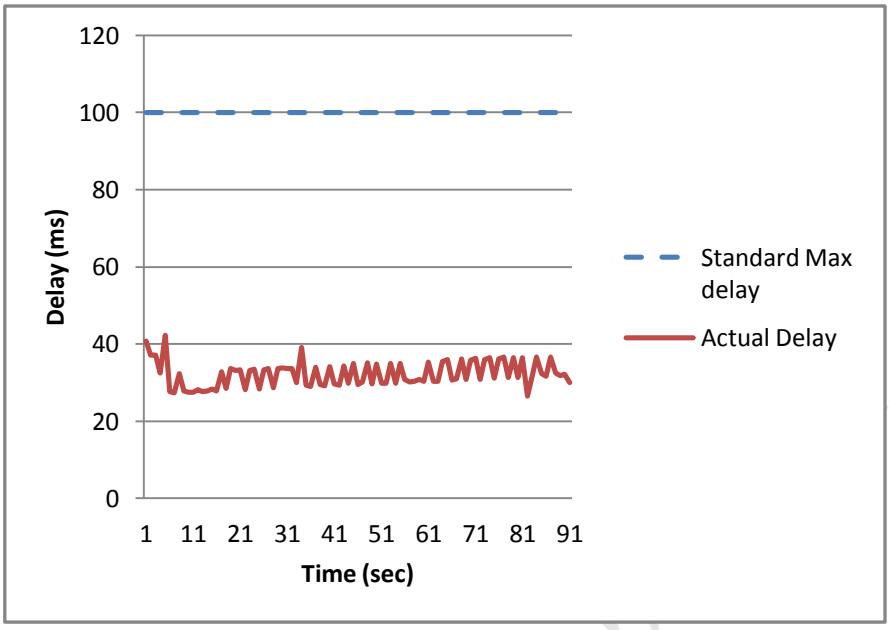


Figure 26 Access network delay

Figure 27 illustrates the access network jitter. Like network delay, jitter is also within the values specified by the ITU-T. The effect of background traffic appears as spikes in the graph. Since the average value is with the minimum required for video, voice and data traffic, the access network is expected to deliver the applications.

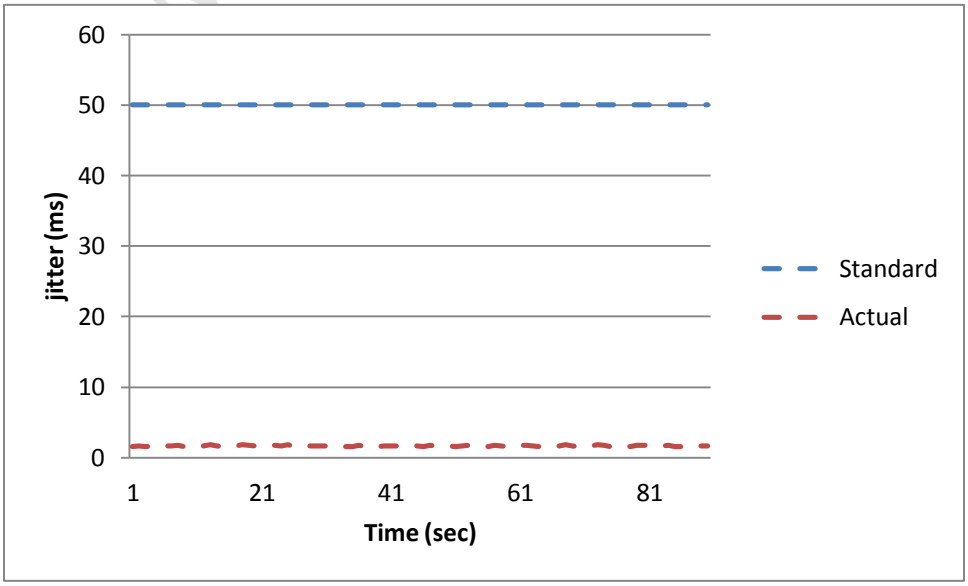


Figure 27: Access network jitter

- **Packet loss**

The average packet loss ratio for the access network is 0.31%. The packet loss value is within the recommended values. The access network must therefore be able to deliver real-time multimedia applications as specified by the ITU-T [1].

Discussion

The results obtained show that the access network performance meets the ITU-T standards. The throughput specified by the manufacturer between a subscriber station and the base station is 12Mbps. There is a difference of about 2Mbps (about 16% of the specified throughput) between this value and that obtained in the tests. The minimum throughput requirement for High Definition IPTV (HDIPTV) service is 10Mbps depending on the codec used. If the service is to be run on such a WiMax network, a subscriber station with a higher throughput, e.g. 20Mbps, must be used to enable the user to run other services without any degradation in the HDIPTV service.

Using Iperf UDP tests, a specific bandwidth can be requested. On the system, a request of 12Mbps between the base station and a subscriber station resulted in an increased packet loss of 3% and a similar degradation in the throughput. Figure 28 shows the effect of requesting higher bandwidth on jitter.

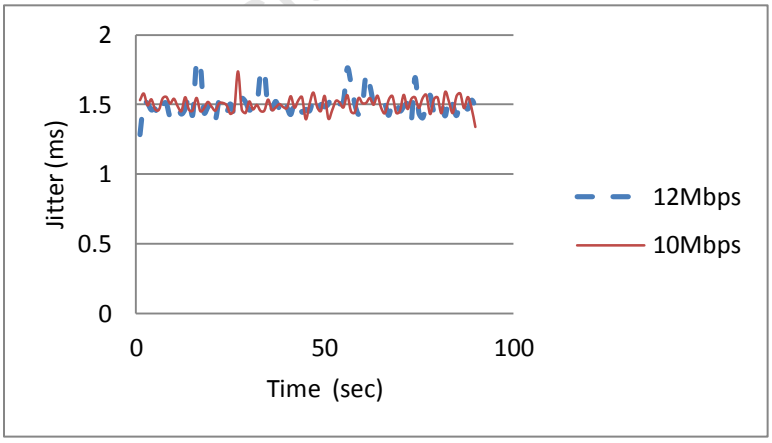


Figure 28: Effect of requesting higher bandwidth on jitter

While the average values for the two requests for bandwidth are the same, there are more instances of high values in the case of requesting 12Mbps, which result in the general deterioration of overall performance of the link. This results in poor service and negative user QoE. This causes jerks in video applications and breaks in voice in applications. A

link can therefore only be used for services whose throughput does not exceed 10Mbps. However, the network may be suitable for standard definition IPTV, IPTV VoD or video streaming services, which require up to 2Mbps throughput, depending on the codec used [55]. Four streams of each of these services can be carried on the access network allowing the end user more than 1Mbps for other services, e.g. Internet and VoIP, which can be delivered with 1Mbps throughput available, running as background traffic.

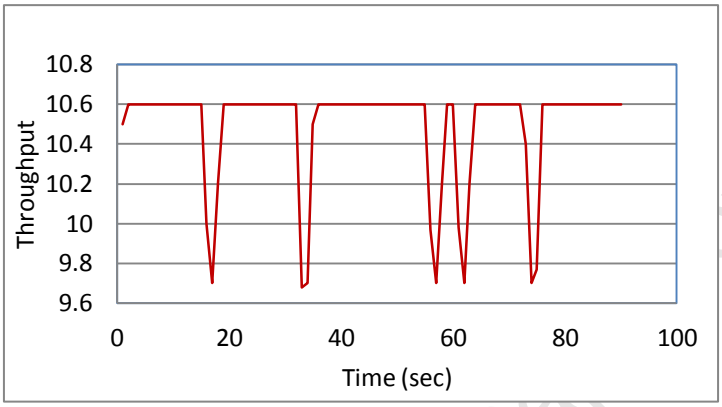


Figure 29: Network throughput when 12Mpbs bandwidth is requested

Figure 29 illustrates the variation of the network throughput with time when 12Mbps is requested. This results in the throughput occasionally dropping down to about 9.7Mbps and the maximum through clipping at 10.6Mbps. The network is therefore stable at a maximum of 9.98Mbps when a bandwidth of 10Mbps bandwidth is requested. The actual throughput available to applications must therefore be lower since signalling information also reduces the throughput of the network. In practical networks where real-time services are to be delivered, a certain percentage of the throughput is reserved for signalling information.

5.1.2 Core network performance tests and results

Figure 30 shows the set up for link quality tests on the core network segment. Iperf tests were run between the ingress router interface connecting to the access network with IP address, 10.50.1.5 and the egress router interface with IP address 10.40.1.2 i.e. the interface connecting to the interior router (refer to figure 20, chapter 4).

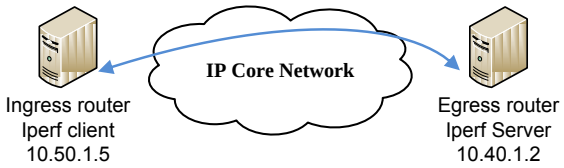


Figure 30: Link quality tests set for core network.

The results for the network performance parameters are as shown in table 11. The results show the average values obtained using Iperf over a period of 90 seconds.

TABLE 11: CORE NETWORK SEGMENT LINK QUALITY TEST RESULTS

QoS Parameter	Value obtained
Throughput	94.5Mbps
Delay	0.19ms
Jitter	0.018ms
Packet Loss	0%

- **Core network throughput**

Figure 31 illustrates the variation of the core network throughput over a period of 90 seconds. The average value measured was 94.5Mbps. CAT5 cables were used to interconnect the core network routers with Fast Ethernet interfaces. This sets the available bandwidth on the network 100Mbps. The core network is therefore capable of delivering both more than one application of both standard and high definition IPTV, which have a minimum throughput of 10Mbps. Voice and data applications can also be carried simultaneously.

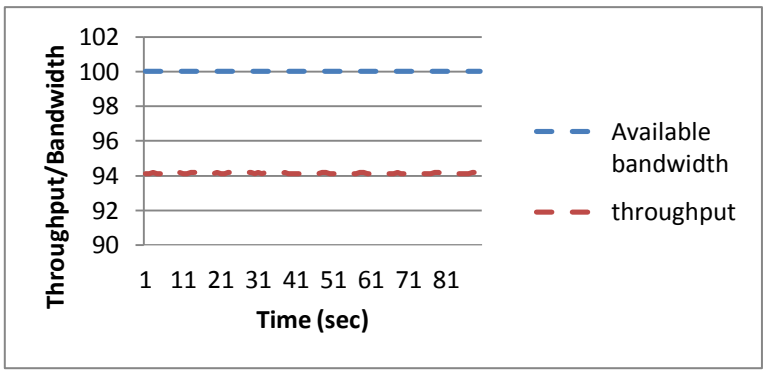


Figure 31: Core network throughput

- **Delay and jitter**

The values of delay and jitter obtained were 0.19ms and 0.01ms respectively. The values were expected to be low since only three routers are interconnected. Network delay and jitter are additive in IP networks, the values per router for the core network would be 0.063ms and 0.003ms respectively. If compare to the minimum 50ms required for deliver of applications, the values of the core network delay and jitter showed that the network is capable of delivering video and voice applications.

- **Packet loss**

The value for packet loss obtained on the network was 0%, compared to the minimum requirement of between 1 x 10^{-3} and 1 x 10^{-4}. This is also is within the requirements of NGN applications.

5.1.3 Discussion

The results of the link quality tests obtained above show that the access and core networks used on the NGN prototype were correctly set up. The values of the performance metrics indicate that the individual networks meet the performance requirements defined for IP networks by the ITU-T. The results also show that the networks are able to deliver voice, video and data applications. While the core network is able to deliver more than one HD and SD IPTV services, the access network can only deliver one SD IPTV service because of the network throughput of about 9Mbps. The end user may also be unable to use other services since the IPTV service exhausts the available network resources.

5.2 End-to-end network performance tests and results

Figure 32 shows the set up for the end-to-end transport network segment. This experiment tests the end-to-end performance of the interconnected networks. Iperf tests were run between one client machine and the egress router, in client/server mode respectively.

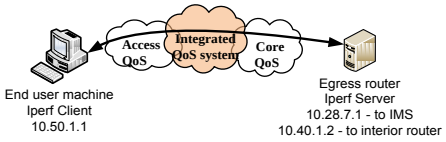


Figure 32: Set up for End-to-end network tests

Table 12 below shows the results obtained for the end-to-end link quality tests. The average values of throughput, delay, jitter and packet loss were found to be 9.92Mbps, 34.94ms, 1.86ms and 0.105% respectively. These results were measured over a period of 90 seconds. The results show that performance metrics of the end-to-end network are within the values specified by the ITU-T for IPTV, video streaming and VoIP applications. The following is an analysis of the values obtained and how they compare with the

TABLE 12: END-TO-END NETWORK SEGMENT LINK QUALITY TEST RESULTS

Parameter	Value
Throughput	9.92Mbps
Delay	34.94ms
Jitter	1.86ms
Packet Loss	0.105%

performance requirements of VoIP, IPTV and video streaming applications.

- **Throughput**

The results show that the access network determines the bandwidth capacity available on the transport network segment. Despite the high throughput of the core network, the throughput of the end-to-end transport network is determined by that of the access network. Figure 33 shows a comparison of the applications performance requirements compared to the actual network performance. IPTV applications have high throughput requirements compared to the other three applications. The requirements however vary, depending on whether the application is standard definition or high definition. High definition broadcast IPTV requires network throughput of between 10Mbps and 15Mbpd depending on the codec used, while standard definition IPTV, IPTV VoD and other premium program sources require between 2.1Mbps and 3.18Mbps depending on the codec used [55].

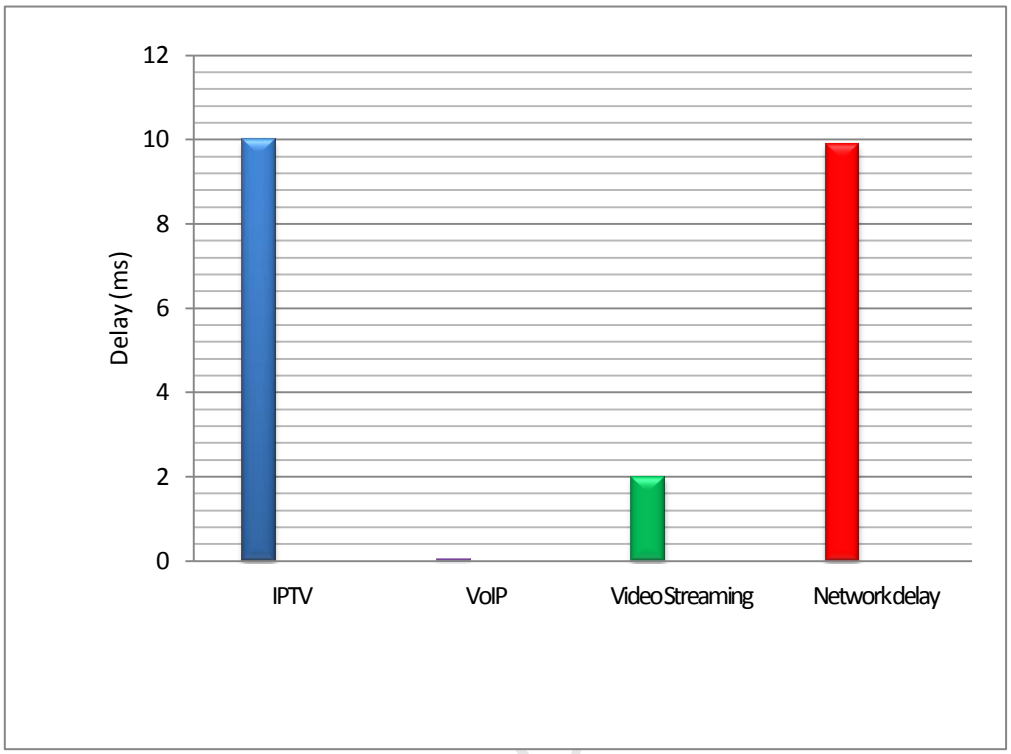


Figure 33: Minimum applications throughput requirements versus network throughput

- **Delay**

The total delay on IP networks with more than one network segment is additive [1]. The delay in practical networks consists of delay components in transmission links, processing delay in network nodes like switches and gateways; the delay on the test bed is mainly due to the processing delay in access Network Elements, i.e. subscriber station and base station indoor/outdoor units. The processing delay in the core network routers and switches is negligible. The access network contributes about 99% of the delay while the core network contributes 1%. The end-to-end delay is higher than the two values obtained for the network separately, this could be due to the delay contributed by the switch interconnecting the two networks. Figure 34 shows a comparison of the transport network delay and the performance requirements of applications.

The video streaming application has less stringent delay requirements compared to IPTV and VoIP applications and is therefore more tolerant to delay than the other applications. The network is however capable of meeting the performance requirements of all the applications.

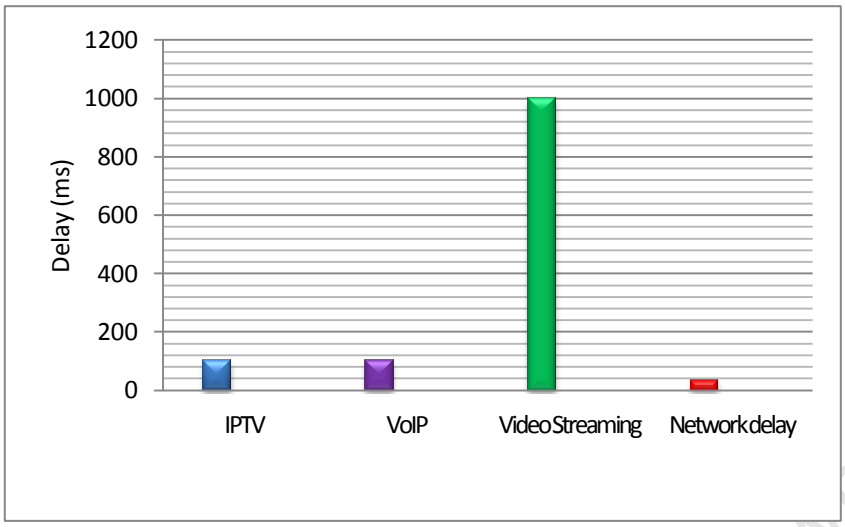


Figure 34: Minimum applications delay requirements versus network delay

- **Packet loss**

The packet loss on the core network is 0% and that for the access network is 0.105%. The access network therefore contributes 100% to the packet loss on the network. Figure 35 shows a comparison of the packet-loss performance requirements of the applications and that of the network. Broadcast IPTV is a strictly low loss application with a stringent requirement of 0.0001% packet loss requirement, which is approximately zero on the graph. While the core network meets the 0% packet loss for this application, the access network may be unable to deliver the application. VoIP, video streaming and IPTV VoD have less stringent packet loss requirements. The network performance is slightly higher than the required values. The actual performance of the application on the network would depend on the codec used. IPTV services, for example, have been classified and associated to the ITU-T RecY.1541 QoS classes [67].

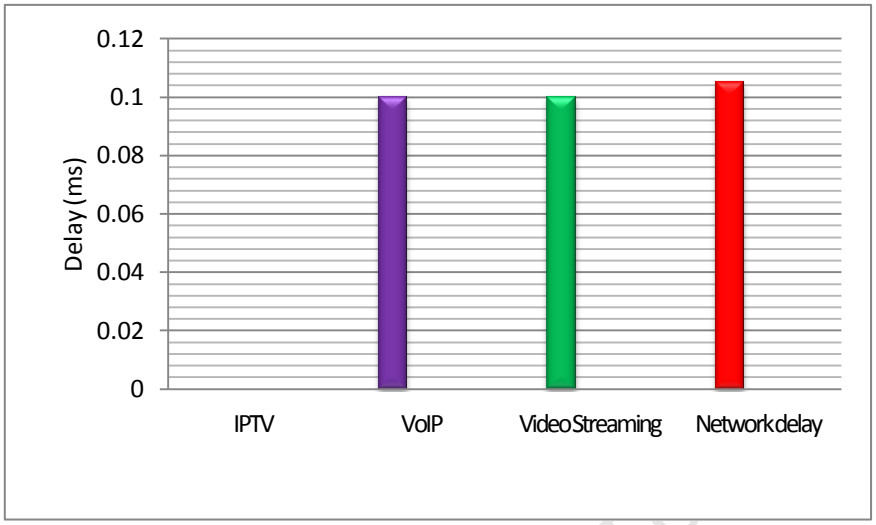


Figure 35: Minimum applications packet loss requirement versus network packet loss

- **Jitter**

Figure 36 shows the jitter performance requirements of the three applications compared to the network performance. While IPTV and VoIP have strict jitter requirements, video streaming is more tolerant to jitter. The network jitter is very low compared to the applications jitter performance requirements. The network therefore meets the QoS requirements of the applications.

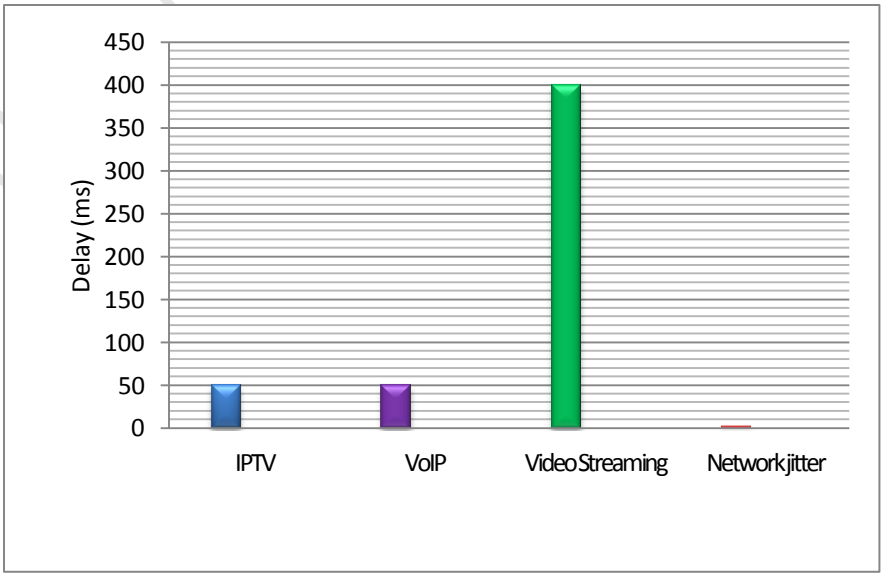


Figure 36: Minimum applications jitter requirement versus network jitter

5.3 Network ability to deliver applications

This section presents an evaluation of the capability of the network to deliver real life applications, i.e. IPTV video on demand, video streaming and data. The experiments evaluate the capability of the proposed QoS architecture within the NGN prototype to deliver applications with different QoS requirements. During the initial tests the network failed to deliver clear video pictures, this was attributed to the Avahi daemon, which runs, by default on all Linux machines. The daemon uses up processor time on the machines causing long delay times, which affect the delivery of the video into and out of the router. The daemon was therefore disabled on all the machines by editing the lines shown in appendix C on each of the routers.

5.3.1 Application tests

Figure 37 shows the flow of services from the terminals through the transport networks to the service networks. The QoS system proposed in this thesis ensures the transport network consistently handles the QoS requirements of the applications as they traverse the network from the point they enter the network at the server side to the point they exit the network at the client side.

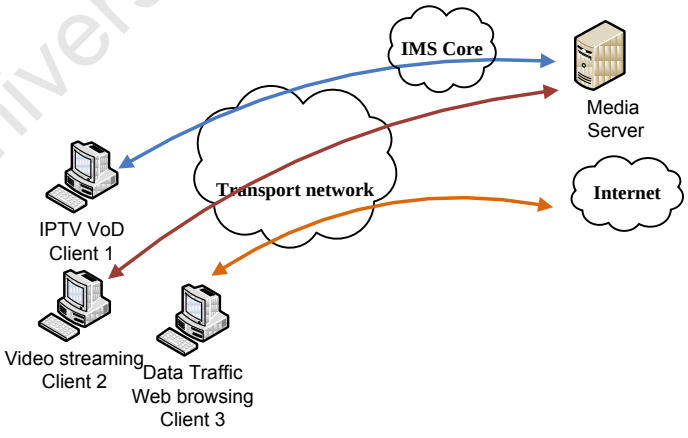


Figure 37: Set up to test network capability to transport integrated services

- **Video streaming and data services**

The video streaming application is run between client 2 and the media server. A VLC client is installed on client 2 machine and the VLC server on the media server. Web browsing/data transfer is accessed from client 3.

- **IPTV VoD service**

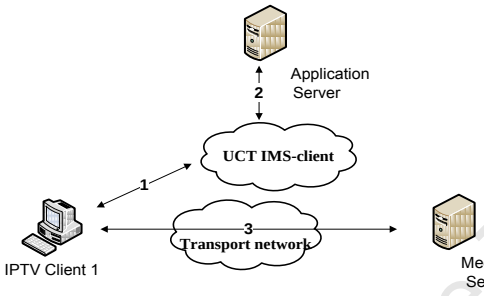


Figure 38: Overview of network set up for IPTV VoD service

Figure 38 shows the connection set up between a client machine, the transport network and the application and media servers. To deliver a service to the client, the process involves session initiation, control process and eventual delivery of content to the UE. The IPTV VoD service is hosted on the media server (a SIP-server on the UCT IMS-client). To access and deliver services from the SIP servers on the IMS networks, the following steps are followed:

Step 1

A terminal registers on the IMS

Step 2

Registration is followed by service discovery and selection on the application server through the IMS control functions

Step 3

The final stage involves streaming media from the media server to the terminal. Details of session set up and signalling for an IPTV service are given in appendix G.

5.3.2 Test Results

Figure 39 shows the results of registration delay experienced by three types of video applications, namely AVI, MPEG and Matroska. These are referred to as videos 1, 2 and 3 in the diagram. Details of the video types used and their codec are given in appendix D.

While the network successfully delivered the applications, it was observed that the quality of the service delivered also depended on the Codecs of the individual video applications. The observed registration delay experienced by the IPTV video on demand service depended on the number and types of applications currently on the network.

Video application 3 experiences the highest registration delay. The video application also suffered from jitter, which caused the picture to appear as broken frames on computer monitor. This could also be attributed to application layer characteristics of the video. There was small variation on the delay values of the rest of the applications including the data application.

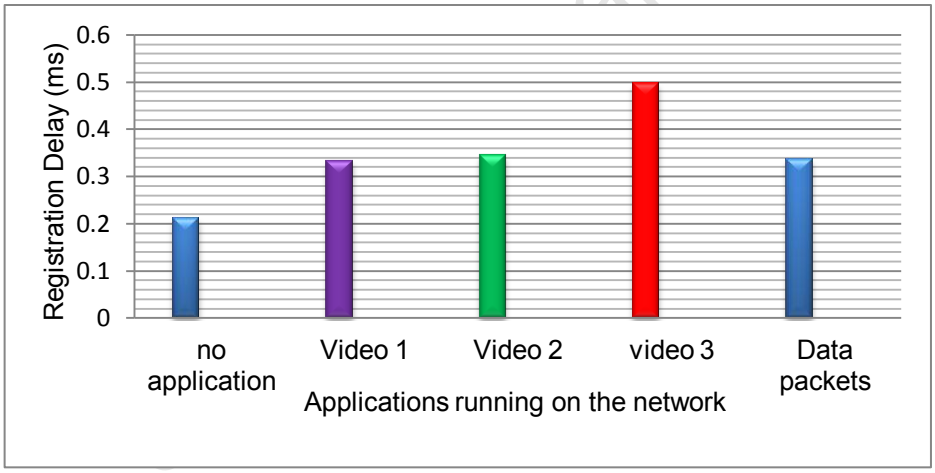


Figure 39:: Applications IMS registration delay for different video applications

It was also observed that the overall end-to-end delay experienced by applications hosted on the IMS SIP servers increased due to additional session set up delay, hence affecting the end-to-end service delivery. The results also showed that the session set up delay experienced by a service depends on the applications/services already running on the network. Figures 40 and 41 show that as the number of services on the network increases, the delay experienced by a service intending to register on the IMS increases.

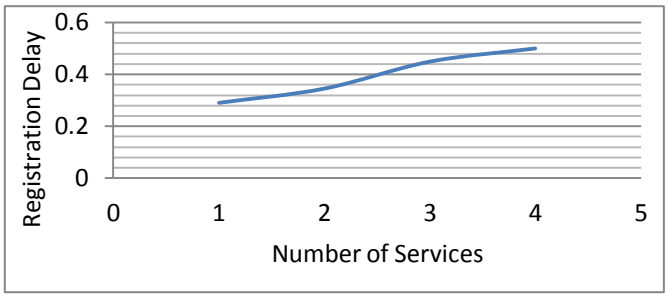


Figure 40: Effect of increasing applications on registration delay

The delay however does not start from zero due to the inherent transport and IMS network delays. During the experiments, it was noted that for a given application, the registration delay also varied. Five instances were therefore noted for each application and services added in turn on the network. Adding video application 2 on the network caused a sharp increase in the registration delay. As subsequent applications are added on the network the delay also increases. The values also show the inherent network delay causing all initial values to be non-zero.

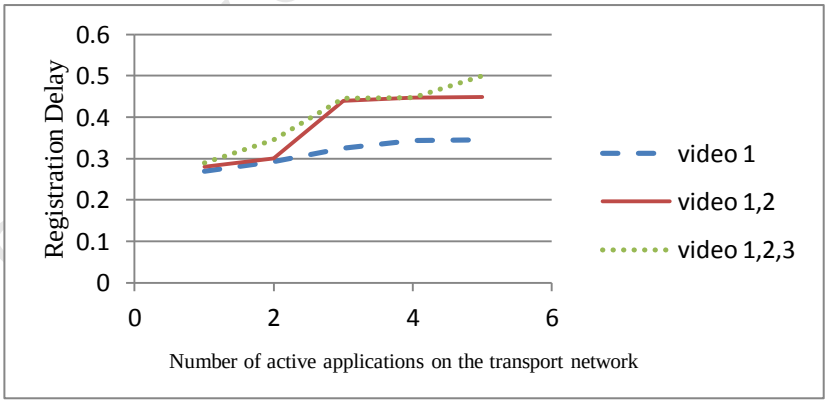


Figure 41: Variation of registration delay as number of active applications increases

- **Results and discussion - jitter and packet loss when two services run on the network**

The following is an analysis of the performance of the transport network when two services are running on the network. Video 1 was used to evaluate the performance of the network. A stream of UDP packets was send between the terminal and the egress router

with the egress router as client first and then with the terminal machine as a client. The results showed that with the egress router in client mode, CPU the Iperf UDP data stream takes up close to 90% of CPU usage causing high packet loss and jitter. The video application experienced delay and the picture quality drops and often shatters. This negatively affects end-user QoE. Figures 42 and 43 below show the variation of jitter and packet loss on the egress router and end user terminal.

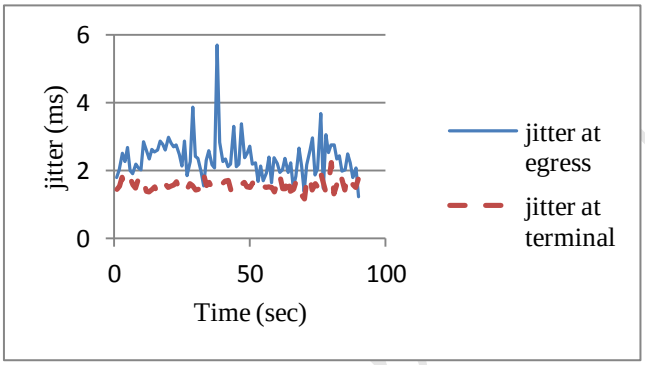


Figure 42: Variation of jitter at the ingress router and on the end user terminal

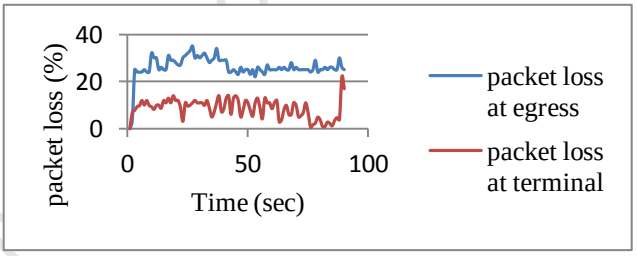


Figure 43: Variation of packet loss at the ingress router and on the end user terminal

5.4 Discussion

The proposed QoS architecture integrates the WiMax and IP core network QoS systems. The results obtained show that the QoS metrics of the network, i.e. delay, throughput, packet loss and jitter are within those recommended by the ITU-T for NGN systems. The first set of experiments, intended to determine the QoS metrics of the access and core networks, showed that the network is able to deliver real time applications such as voice and video. The second set of experiments evaluated the performance of the proposed QoS architecture within NGN prototype system. The results obtained show that the

network is able to deliver voice, video and data applications; since the values of QoS metrics fall within those recommended for the applications by the ITU-T.

A number of video streaming and IPTV video on demand applications were run on the network to determine the network's capability to deliver the applications. While the network was able to deliver the applications across the network, the results showed that the codec of video applications also determined whether the applications could be delivered or not. While the network met the QoS requirements of the applications, application layer metrics also needed to be taken into consideration. The codec determined the number of video applications that could be simultaneously run on the network without degradation of services to the rest of the applications running on the network. While it is possible to define the QoS parameters of applications using the proposed QoS architecture, such parameters as the codec of applications need to be taken into consideration.

Chapter 6

Conclusions and future work

6.0 Conclusions

The evolution of telecommunication networks to NGN systems has provided customers seamless connectivity using heterogeneous network technologies. Services can now be delivered between end users using different network technologies. The performance of the NGN technologies has been enhanced by use of QoS systems. QoS systems have, however been isolated to individual interconnected technologies. Interoperability between QoS systems is one of the challenges faced by network operators to provide QoS guarantees to applications as they traverse different network technologies. A number of end-to-end QoS solutions presented in the literature have addressed this problem. It was found that a number of techniques in the literature have been used to address the end-to-end QoS problem. A thorough review of the literature revealed that practical implementation of the solutions is difficult. Implementation of QoS systems in network devices is one technique that authors have used to achieve QoS control in IP networks. This approach is used in this study on a practical NGN prototype.

Previous chapters have presented the design and implementation of the proposed QoS architecture within the NGN prototype. The proposed QoS architecture achieves consistent QoS between a WiMax access and network and an IP core network by integrating the QoS systems of the two network technologies. The purpose of this proposal was to determine if consistent QoS control can be achieved between the two network technologies with different QoS systems.

A QoS architecture that integrates the WiMax QoS system and that of an IP-based core network was successfully implemented on a NGN prototype test bed. The NGN prototype was used to quantitatively evaluate the performance of the proposed QoS architecture within the NGN prototype system. A number of experiments were carried out on the NGN test bed resembling a real world NGN system. Results were compared to those specified for NGN systems and applications by the ITU-T. Based on the findings in the preceding chapters, the following conclusions were drawn:

- The evaluations showed that the WiMax access network under study conforms to the QoS requirements for a NGN access network as specified by the ITU-T. The access network QoS system also provides a suitable platform for the creation of a QoS solution that enables consistent QoS handling of applications between itself and another network technology, specifically an IP core network.
- The IP core network used in the experiments is based on Linux machines turned into IP routers by installing QoS and routing modules. The core network QoS system provided the second QoS system required to build the proposed QoS architecture required to build the proposed QoS architecture. Evaluations of the core network show that the QoS values of the network are within those defined for IP core networks by the ITU-T.
- The evaluations demonstrated that in NGN systems, network technologies with different QoS systems can provide consistent QoS to applications and services traversing the networks, if the different QoS systems are integrated into a system that ensures consistent QoS handling between the two networks.
- The obtained results have also shown that successful delivery of video applications, depends on the codec used. This is because different codec require different processing time on the Network Elements.
- The test results also showed that DiffServ, which is the widely accepted QoS system for IP networks, can still be used in NGN systems to provide end-to-end QoS to applications and services.

6.1 Recommendations and future work

A number of technologies have been presented in this study. These include the WiMax system and its QoS mechanisms, Linux-based IP routers and routing and QoS mechanisms, DiffServ and the IMS systems. The use of these technologies on the NGN prototype and on some of them in the implementation of the proposed QoS architecture shows the complexities faced by network operators as they roll out NGN systems. A number of issues and areas for further research have therefore been identified that could assist network operators in the successful implementation of NGN systems. Below is a list of some of the important recommendations to be considered.

- The focus of this work was to achieve end-to-end QoS when two transport networks with different QoS systems are involved in the delivery of services or applications. The evaluations focused on the end systems running in client/server mode. An important consideration would be a set up where end users use different access technologies, e.g. WiMax and a DSL network, WiMax and a 3G network.
- MPLS modules for Linux machines are now available. Since MPLS is one of the alternative technologies for core networks, there is need to investigate the performance of the proposed QoS architecture when the core network is based on MPLS routers.
- In this study, only video and data applications were used in evaluating the performance of the network. Voice could be added to the applications to determine the performance of the network when voice, video and data applications are run on the network.
- An Ethernet-based micro base station was used on the WiMax network. This did not require the use of an ASN-GW. QoS implementation was carried out on an IP module that provides an IP interface to the base station. Further work could involve carrying out the same experiments with the ASN-GW connected to the access network.
- In the study, the IMS platform was used for hosting applications. To fully utilize the platform, IMS modules must be installed in the transport Network Elements, routers and the WiMax ASN-GW. This work therefore provides a background work for further study in this area.

Bibliography

[1] ITU-T, “Framework for achieving end-to-end IP performance objectives”, ITU-T REC Y.1542, Feb 2006.

[2] S. Uskela, “Multiprotocol Label Switching (MPLS)”, WiMax Forum, “Stage 2: Architecture Tenets, Reference Model and Reference Points”, WiMax Forum, 2008, http://www.tml.tkk.fi/Opinnot/T-109.501/2002/reports/MPLS.pdf.

[3] ITU-T, “Network performance objectives for IP-based services”, ITU-T REC Y.1541, Feb 2006.

[4] ITU-T, “Resource and admission control functions in Next Generation Networks”, ITU-T REC Y.2111, September 2006.

[5] K. Ishiguro, “Quagga: A routing Software package for TCP/IP networks” July 2006, www.quagga.com, accessed June 2009.

[6] ITU-T, “Transmission Systems and Media, Digital Systems and Networks Multimedia Quality of Service and Performance – generic and user-related aspects, “Framework for achieving end-to-end IP performance objectives”, ITU-T REC G.1080, December 2008.

[7] “Troubleshooting DOCSIS – VoIP Impairments > Delay & Jitter” http://bradyvolpe.com/2009/02/16/voip_impairments_delay_jitter/

[8] “QoS”, http://www.voip-info.org/wiki/view/QoS

[9] V. Ozianyi and N. Ventura, "A Novel Pricing Approach for QoS Enabled 3G Networks," ICN, pp.578-586, The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)l, 2005.

[10] R. Braden, et. al., “Resource Reservation Protocol (RSVP)”, *RFC 2205 IETF* 1997.

[11] S. Blake, et al. “An Architecture for Differentiated Services”, *IETF RFC 2475,* 1998.

[12] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, “Assured Forwarding PHB Group”, *RFC2597* 1999.

[13] Cisco Systems, “Quality of Service (QoS) Networking” www.cisco.com - white paper, accessed June 2009.

[14] G. B. Leao, et, al., “End-to-end Signalling with Heterogeneous QoS Models Support”, The Weird Project, 2008.

[15] S. Iacono, “Policy based management for Next Generation mobile networks”, IEEE, 2003 Wireless.

[16] 3GPP, “Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS)”, 3GPP TS 23.228 V7.7.0 Release 7 March 2007.

[17] L. Skorin-Kapov et. al, “Application-level QoS negotiation and signalling for advanced Multi-media services in the IMS”, IEEE Communications Magazine Vol 45, 2007, pg 144-150.

[18] C. E. Rothenburg et. a.l, “A review of Policy-Based Resource and Admission Control Functions in Evolving Access and Next Generation Networks” Springer Science+Business Media, 2008.

[19] Z. Zhang at. el., “Decoupling QoS control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services”, *SIGCOM,* 2000.

[20] Z. Duan et. al., “Service Overlay Networks: SLA’s, QoS and Bandwidth Provisioning” IEEE/ACM, Transactions on Networking, Vol 11 pages 870-883, December 2003.

[21] IEEE, “Part 16: Air Interface For Fixed Broadband Wireless Access System” IEEE Computer Society IEEE STD 802.16-2004 Oct 2004.
http://en.wikipedia.org/wiki/List_of_deployed_WiMAX_networks

[22] WiMax Forum, “IEEE 802.16a Standard and WiMAX Igniting Broadband Wireless Access”, www.wimax.com accessed July 2008.

[23] J. Chen, W. Jiao, Q. Guo “ An Integrated QoS Control Architecture for IEEE 802.16 Broadband Wireless Access Systems”, www.lucent.com accessed July 2008.

[24] WiMax Forum, Network Architecture (Stage 2: Architecture Tenets, Reference Model and Reference Points) [3GPP2 – WiMAX Interworking] Release 1, Version 1.2, January 11, 2008.

[25] WiMax Forum, “Mobile WiMax – Part 1: A Technical Overview and Performance Evaluation”, August 2006.

[26] WiMax Forum, “Stage 2: Architecture Tenets, Reference Model and Reference Points”, WiMax Forum, 2008.

[27] L. Berger, “Generalized Multi-Protocol Label Switching (GMPLS) Signalling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions” ietf, RFC 3473.

[28] A. Narayanan, “RSVP Extensions for Flexible Resource Sharing”, ietf draft, October 2009.

[29] G. Lee, L. Li, W Chien, “Heterogeneous RSVP Extension for End-to-End QoS Support in UMTS/WLAN Interworking Systems”, Proceedings of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, Pages: 170-175, 2006.

[30] W. Lee, S. Kim, J. Park, “A lightweight implementation of RSVP-TE protocol for MPLS-TE signalling Source”, Computer Communications Volume 30 , Issue 6 pages 1199-1204, April 2007.

[31] S. Van Den et el, “NSIS Network Service Layer Protocol QoS Signalling Proof-of-Concept”, ietf, Feb 2004.

[32] S. Mignanti et al, “WEIRD – Real Use Cases and Applications for the WiMax Technology”, IEEE CCNC 2008.

[33] J. Rosenberg, “SIP: Session Initiation Protocol”, ietf RFC3261, June 2002.

[34] Ciulli, N.et. al., “A QoS Model Based on NSIS Signalling Applied to IEEE 802.16 Networks”, IEEE CCNC, Vol. pp. 953 – 957, Jan 2008.

[35] M. Boucadair, et. al., “A Framework for end-to-end service differentiation: Network Planes and Parallel Internet”, IEEECommunication Magazine, vol.45, page 134-143, September 2007.

[36] S. Zaghloul and A, Jukan, “Extending QoS from Radio Access to an All-IP Core in 3G Networks: An operator’s Perspective”, IEEE Communication magazine, vol 45 issue 9, page 124-132, September 2007

[37] 3GPP, “The Mobile Broadband Evolution 3GPP Release 8 and beyond HSPA+, SAE/LTE and LTE-Advanced”, 3GPP, Feb 2009.

[38] W. Zhuang et. al, “Policy-based QoS Management Architecture in an Integrated UMTS and WLAN Environment”., IEEE Communications Magazine, vol 41 issue 11, page 118-125, November 2003.

[39] “MONASIDRE (Management of Networks and Services in a Diversified Radio Environment)”, ftp://ftp.cordis.europa.eu/pub/ist/docs/ka4/mob_monasidre.pdf

[40] T. Magdanz et, al, “Experiences on the Establishment and Provisioning of NGN/IMS Testbeds - The Fokus Open IMS Playground and the Related Open Source IMS Core” http://www.icin.biz/files/programmes/Session2B-2.pdf

[41] “Future Internet Infrastructures for FI Prototyping” http://www.fokus.fraunhofer.de/en/ngni/_files/FOKUSinno-FI-2009-11.pdf

[42] B. Ionescu et. al., “A Test bed and Research Network for Next Generation Services over Next Generation Networks”, pp. 22-31, First International Conference on Test beds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM'05), 2005.

[43] PAN European Laboratory Infrastructure, ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/fire/firelaunch-paris-100908-3-2_en.pdf

[44] Fraunhofer Fokus Institute, http://www.denic.de/fileadmin/Dialog/ENUMTage/Fiedler_20080418.pdf

[45] http://uctimsclient.berlios.de

[46] The Akari Project, http://akari-project.nict.go.jp/eng/document/asiafi-seminar-harai080826.pdf

[47] Cisco Systems, “IP Class of service for mobile networks” www.cisco.com.

[48] V. Fineberg, “QoS Support in MPLS Networks”, MPLS/Frame Relay Alliance, My 2003

[49] R. Prabagaran & J. Evans, “Experiences with Class of Service (CoS) Translations in IP/MPLS Networks”, http://www.ittc.ku.edu/~evans/papers/lcn01_mpls_cos.pdf

[50] ITU-T, “Internet protocol aspects – Quality of service and network performance Network performance objectives for IP-based services Amendment 1: Revised Appendix VI: Applicability of the Y.1221 transfer capabilities and IETF differentiated services to IP QoS classes” ITU-T REC Y.1541 Aug 2003.

[51] M. A. Brown, “Traffic Control HOWTO version 1.0.2” www.fespppr.br/~airton/tmp/iproute.pdf.

[52] G. Stattenberger et al, “Performance Evaluation of a Linux DiffServ implementation”, Elsevier, Computer Communications Vol 25, 2002, pages 1195-1213

[53] Martin A. Brown, “Traffic control using tcng and HTB HOWTO version 1.0”, http://www.faqs.org/doc/Linux-HOWTO/Traffic -Control-tcng-HTB-HOWTO.html

[54] Jussi Lemponen, “Implementation of Differentiated Services Policy Information Base on Linux”, www.atm.tut.fi/faster/qbone/**linux**-pep.pdf

[55] ITU-T, “Transmission Systems and Media, Digital Systems and Networks Multimedia Quality of Service and Performance – generic and user-related aspects,

"Framework for achieving end-to-end IP performance objectives"", ITU-T REC G.1080, December 2008.

[56] www.alvarion.com

Appendix A

End-user Quality of Experience (QoE)

This appendix describes extra details relevant to end-user quality of experience (QoE) in IP networks. The ITU-T defines QoE as follows: “The overall acceptability of an application or service, as perceived subjectively by the end user. It includes the complete end-to-end system effects i.e. client, terminal, network, services infrastructure, etc, and may be influenced by user expectations and context. In principle, QoE is measured subjectively by the end user and may differ from one user to the other. However it is often estimated using objective measurements [55]”. QoE is therefore a factor of the network QoS metrics as well as the perception and expectation of the end-user. A NGN platform delivering applications like IPTV and video streaming consists of a content acquisition, encoding and play out source; a core network; an access network and the customer network [55]. Each of the network segments contribute to the overall QoE expected by the end-user.

QoE at the application level is determined by the codec used, the quality of the source material, resolution, bit rate, video encoding and pre-processing mechanisms used. At the network level, network QoS metrics, i.e. delay, jitter, packet loss and throughput affect the quality of the overall service offered to the applications by the transport network.

Real time applications like video on demand may impose different delay and jitter requirements on the network depending on the codec used. According to the ITU-T Recommendation G.1080, network latency and jitter for IPTV must be less than the set-top box jitter-buffer provisioning. Jitter beyond this value manifests itself as packet loss. Recommended packet loss rates are also provided for given video and audio applications to ensure end-user satisfaction. Packet loss objectives are divided into loss distance, the distance between consecutive packet loss events, loss period, and the duration of the loss event. If the network performance is below the recommended values, mechanisms like forward error correction (FEC) and bit interleaving must be used on the transport network. At the application layer, loss concealment and FEC can also be used. Characteristics of video applications used in this thesis are given in appendix E.

Appendix B

Details of the WiMax network used

B.1 WiMax access network set up

The WiMax access network used in this thesis is based on the IEEE802.16 standard, specifically the 802.16d family supplied by Alvarion. The network configuration is fixed access point-to-multipoint. The network was set up in earlier research work in the lab.

Figure 44 shows the layout of the network. The network consists of three subscriber stations and a single-sector base station. The WiMax network connects to the core network via a switch.

RF cables are used to connect the base station and subscriber stations to reduce the hazard of electromagnetic radiation in the lab. An indoor installation was preferred to an outdoor installation to eliminate the need register the network with the national regulator.

Channel conditions, i.e. signal strength between the base station and the subscriber stations can be varied using a variable attenuator. Variation of the signal emulates the varying air interface conditions on practical links. The response of the system to the changing signal quality is used determine the performance of the system in real life networks. A splitter attached to the RF link from the base station connects the subscriber stations to the base station.

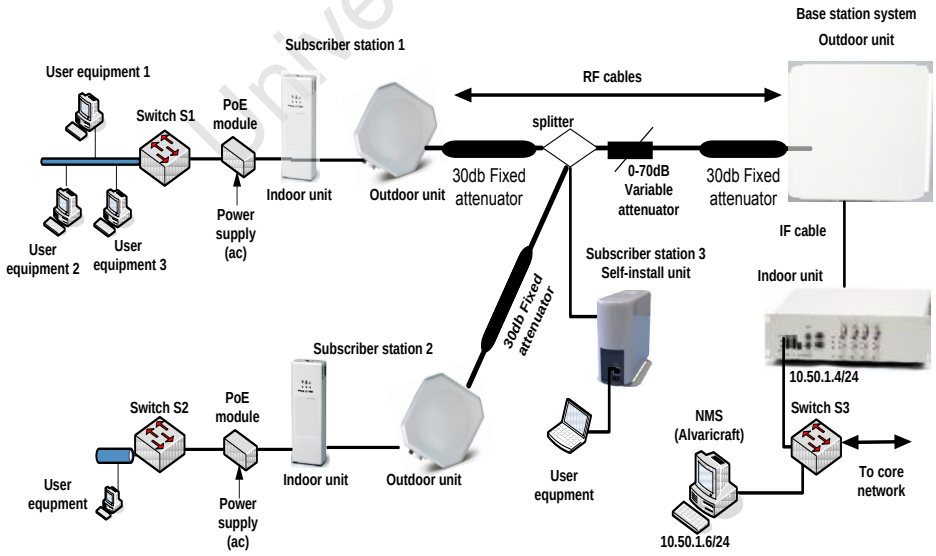


Figure 44: Access Network Implementation

Each subscriber station consists of an indoor and outdoor unit. A 48VDC Power over Ethernet (PoE) module provides power to each of the subscriber stations through a CAT5 cable. The self-install subscriber station in the diagram is an indoor subscriber station unit with the radio and IP modules integrated into a single unit. The unit does not use the power over Ethernet module. It obtains power directly from an AC source.

Switches S1 and S2 shown in the diagram connect the subscriber stations to multiple computers at the customer end to the subscriber stations. Switch S3 connects the WiMax network to the IP core network.

The WiMax network is managed using SNMP or telnet tools. When SNMP is used, the radio configuration and provisioning is done using Alvaricraft, network management software which is supplied together with the radio. The software is installed on the base station indoor unit and accessed via a windows-based machine, indicated as NMS in the diagram. The IP address of the machine must be in the same subnet as the base station indoor unit. This restricts access to the network to this machine only. When telnet is used, the network management machine is registered manually on the base station unit and assigned login details.

B.2 Subscriber station set up

Figure 45 shows the subscriber station connection set up. The PoE module has two Ethernet interfaces. One for provides an Ethernet connection for data only to and from end-user equipment and the other carries both data and power to and from the subscriber station indoor unit. The outdoor unit consists of the radio modules and the indoor unit consists of the IP modules.

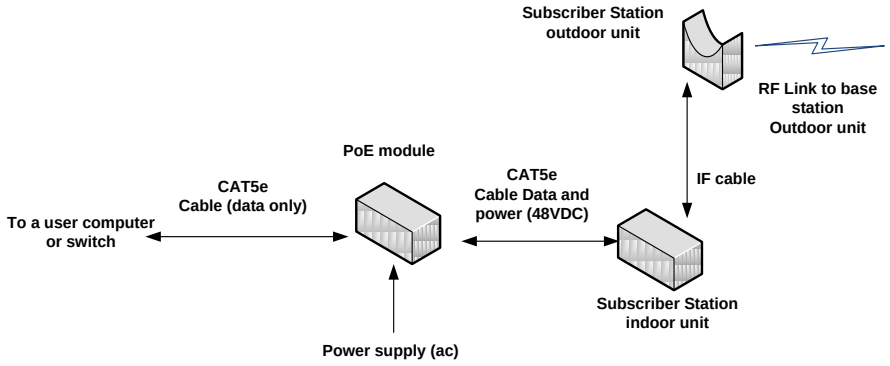


Figure 45: Subscriber station network elements

B.3 Base station connection set up

Figure 46 below shows connection set up for the base station. The indoor unit connects the WiMax network to the core network via a switch. The unit also hosts the Network Management System (NMS) required to control, monitor and provision services on the WiMax network. Communication with upstream IP networks is via an Ethernet interface i.e. a data port. A wireless interface connects the unit to the outdoor unit.

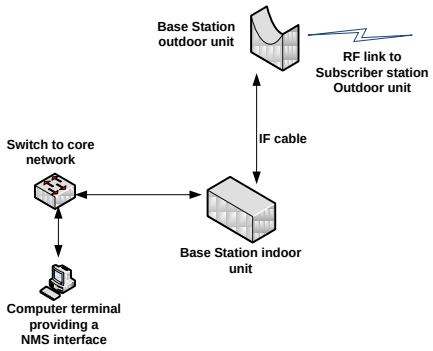


Figure 46: Base station network elements

Another Ethernet interface is available for connecting a laptop or desktop computer for managing the WiMax network. The data port can carry both data and management signals. The management port is used for management purposes only. For simpler connections to the access network, the data port is used in this thesis for both management and data traffic. This removes the need for dedicated network management resources.

The WiMax network IP connectivity is available on the base station indoor unit data port. The port was therefore assigned an IP address on the test bed. Subscriber stations register on the network when they connect to the base station for the first time. They are identified on the network by their MAC addresses stored on the base station indoor unit.

It is a requirement that the network management terminal be in the same subnet address as the base station indoor unit. In the implementation, the NMS computer was assigned IP address 10.50.1.6/24 and the data port of the base station indoor unit assigned the IP address 10.50.1.4/24. Management of the network can also be done using a telnet or SNMP connection.

Appendix C

Router Implementation Issues and Procedures in Linux

This section describes the procedures used in this thesis to implement routing and QoS in the Linux machines used as routers on the core network. The first part describes the Quagga routing software used to implement routing in the Linux machines. The second part describes the procedures used to assign IP addresses to the router interfaces and enable the routers to send and receives traffic both within and from other networks. The last part of the appendix describes the procedures used to enable DiffServ QoS on the routers and how traffic in the network was assigned to the various traffic classes as part of the QoS management in the core network.

C.1 Quagga routing software

Routing in the core network used in this thesis was achieved by installing an open routing software package called Quagga onto the Linux routers. Quagga is an open source routing software package used to provide TCP/IP-based routing services. The software supports routing protocols such as RIP, BGP and OSPF. With Quagga installed on it, a Linux machine can act as a dedicated router, exchanging routing information with other routers.

Quagga is composed of several daemons that work together to create a routing table. One daemon, zebra is the kernel routing table manager, which acts as an abstraction layer to the underlying Linux kernel. Zebra has an interactive user interface for each routing protocol. Due to this design, routing protocols can easily be added.

A daemon exists for each of the supported routing protocols. In this thesis, the routing protocol RIP was used due to the small size of the network. The RIP and Zebra daemons were therefore activated. A set of configuration is available for Quagga, similar to those used on Unix-based routers and these were used to configure the routing functionalities on the routers.

Among other platforms, Quagga supports the GNU/LINUX 2.4.x and higher platforms. The platform used in this study, is GNU/LINUX 2.6.27-11. Quagga is available for download on the official Quagga website: www.quagga.net. The software used in this thesis was down loaded from the UCT repository using the following steps on each of three Linux machines

used as routers:

1. System
2. Administration
3. Synaptic package manager
4. Settings
5. Repositories
6. **Uncheck source code**
7. **Download from**: other
8. **Select the repository:** ftp.leg.uct.ac.za
9. **Choose:** server
10. **select:** reload
11. **Select:** Quagga
12. **select:** mark for installation
13. **Select:** apply

A configuration script is available which automatically detects host configurations. Configuration options are available to customize the script to specific requirements via the configuration interface.

Installation of Quagga on a Linux machine involves configuration, compilation and installation. The following section describes these processes.

C.1.1 Software Configuration, Compilation and Installation

Quagga software was downloaded from the UCT repository: ftp.leg.uct.ac.za onto each the three Linux machines and saved to */etc/quagga/quagga.conf.* Quagga automatically detects most host configurations. It also allows the administrator to customize the configuration script. In this study, it was necessary to disable IPv6 and enable IPv4. This was necessary because the rest of the network uses IPv4 IP addresses. The Zebra daemon was enabled since it provides a configuration interface for all the routing protocols. The RIP daemon was also enabled because this is the routing protocol used on the core network. The configurations were done by editing the daemon file as follows:

#vim /etc/quagga/daemons:
Zebra=yes
bgpd=no
ospfd=no
ospf6d=no
ripd=yes
ripngd=no

To enable an administrator to configure the routers remotely, it was necessary to enable

"vty" as follows:

```
#vim /etc/quagga/debian.conf
vtysh_enable=yes
zebra_options="--daemon"
bgpd_options="--daemon -A"
ospfd_options="--daemon -A"
ospf6d_options="--daemon -A"
ospf6d_options="--daemon -A"
ripngd_options="--daemon -A"
isisd_options="-daemon -A"
```

By removing the "-A" option in the daemons in this file, the router can be access using any set IP address other than the loopback address "127.0.0.1", which is the default IP address for each daemon.

After each configuration process, it is necessary to restart Quagga for the changes to take effect issuing the command:

 # /etc/init.d/quagga restart

Once all the changes were done, compiling the software to the system was achieved by issuing the command "*make*" in the route of the source directory.

A list of commands used to configure the router hostname, password and other actions is available in [5]. These commands make it possible to telnet into any of the routers using an IP address assigned to one of the interfaces. To assign a hostname to the router the following command was used:

 # hostname <hostname>, hostnames used in the thesis are *ingress*, *interior* and *egress* for the ingress, interior and egress routers respectively.

To log into the router, it was necessary to create a username and password to prevent other users from changing the configurations. The following commands are used to set the username and password:

 # username <username>

 # password <password>

The next section describes IP configurations set up on the routers.

C.2 IP configurations

IP configurations includes assigning IP address to all the network interfaced cards, Domain Name Server addresses and configuring the routers to forward IP traffic so that they can communicate with other networks. The IP addresses assigned to the ingress,

interior and egress routers are shown in figure 20 of chapter 4. Traffic enters or leaves each router via the interfaces eth0 and eth1 depending on the direction of the traffic.

By editing the interfaces file, the following commands were used to assign IP addresses to the ingress router interfaces. The first command line accesses the interfaces file, which hosts the interface configuration options.

```
#sudo vi /etc/network/interface/interfaces
>auto eth0
>iface eth0 inter static
>address: 10.50.1.5
>mask: 255.255.255.0
>gateway: 10.30.1.2
```

The above statements assign the IP address 10.50.1.5 to the interface eth0 of the ingress router. The gateway address 10.30.1.2 determines the next hop for the traffic as it leaves the ingress router. Editing the same file as follows assigns an IP address to interface eth1.

```
#sudo vi /etc/network/interface/interfaces
>auto eth1
>iface eth1 inter static
>address: 10.30.1.1
>mask: 255.255.255.0
>gateway: 10.40.1.1
#sudo /etc/init.d/networking restart
```

The last statement is necessary for the changes to take effect.

The same procedure was repeated to assign IP and addresses the interior and egress routers. The following commands were used to assign IP addresses and a gateway address to the interior router eth0 and eth1 interfaces.

Interface eth0:

```
#sudo vi /etc/network/interface/interfaces
>auto eth0
>iface eth0 inter static
>address: 10.10.30.1.2
>mask: 255.255.255.0
>gateway: 10.40.1.2
```

Interface eth1:

```
#sudo vi /etc/network/interface/interfaces
>auto eth1
>iface eth1 inter static
>address: 10.40.1.1
>mask: 255.255.255.0
>gateway: 10.128.7.1
#sudo /etc/init.d/networking restart
```

The following commands assign IP addresses to the egress router.

Interface eth0:

```
#sudo vi /etc/network/interface/interfaces
>auto eth0
>iface eth0 inter static
>address: 10.40.1.2
>mask: 255.255.255.0
>gateway: 10.128.7.2
```

Interface eth1:

```
#sudo vi /etc/network/interface/interfaces
>auto eth1
>iface eth1 inter static
>address: 10.128.7.1
>mask: 255.255.255.0
#sudo /etc/init.d/networking restart
```

By default, forwarding IP traffic in Linux routers is disabled. This is enabled by editing the syctl.conf file in the /etc directory. The following line was edited in all the three routers:

```
 # gedit /etc/syctl.conf
>net.ipv4.ip-forward=1
```

IP version 4 (IPv4) IP addresses were used in the thesis. The statement therefore enables the router to forward traffic with IPv4 address to each other and other networks.

C.3 Configuring routers to support QoS using DiffServ

DiffServ is used is used to support QoS on the NGN prototype implemented in this thesis. This section describes the implementation of DiffServ in the core network routers. DiffServ implementation on Linux has a set of traffic conditioning modules which allow the user to set up QoS on any of the edge and interior routers of a DiffServ domain. The traffic conditioners used in this thesis include a marker – to mark the different traffic flows, a classifier – to place the traffic flows into the appropriate QoS classes and services handlers that identify the EF and AF DiffServ traffic classes. The traffic conditioners were implemented as kernel modules that are activated using the traffic control (tc) command which part of the *iproute2 source package*, installed on the Linux machines.

C.3.1 Activating DiffServ in the Linux kernel

To activate DiffServ on the routers, the following change was effected in the ***config*** file of each of the routers:

>tc_config_diffserv=n,

was changed to

> *tc_config_diffserv=y.*

For the Linux kernel to support the "tc" command, the "tcng" source file must be enabled. This achieved by running the following command:

 # ./configure –notcsim,

C.3.2 Disabling the Avahi daemon

The Avahi daemon was disabled by editing the following lines in the Liunx kernel for each of the routers.

```
# 1 = Try to detect unicast dns servers that serve .local and disable avahi
in
# that case, 0 = Don't try to detect .local unicast dns servers, can cause
# troubles on misconfigured networks
AVAHI_DAEMON_DETECT_LOCAL=0
```

C.3.2 Traffic classification

Three traffic flows were used in this thesis, i.e. IPTV video on demand, video streaming and data. For simplification, traffic classification was effected on ingress router. The interior and egress routers were configured to use by default, the traffic classes defined on the ingress. This is possible in a DiffServ domain to avoid further traffic processing inside the network that could slow down traffic and cause network congestion.

The following commands were used to create traffic classes on the ingress router's interface eth0 and to assign DSCP values to the traffic streams. The commands translate the DiffServ QoS classes used in this thesis, i.e. EF, AF_{11} and BE for IPTV video on demand, video streaming and data respectively, into codes that can be read by the machine. EF, AF_{11} and BE are translated into the hexadecimal machine codes *0x42, 0x10* and *0x34* respectively.

1. *tc qdisc add dev eth0 handle 1:0 root dsmark indices 64*
2. *tc class change dev eth0 parent 1:0 classid 1:1 dsmark \ mask 0 value 0x46*

3. *tc class change dev eth0 parent 1:0 classid 1:2 dsmark \ mask 0 value 0x10*
4. *tc class change dev eth0 parent 1:0 classid 1:3 dsmark \ mask 0 value 0x34*
5. *tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32\ match ip src 10.50.1.1/24 flowid 1:1*
6. *tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32\ match ip src 10.50.1.1/24 flowid 1:2*
7. *tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32\ match ip src 10.50.1.1/24 flowid 1:3*

The first command attaches a DiffServ queuing discipline under which a number of DiffServ QoS classes can be created. In this thesis, a single queuing discipline was created with three traffic classes. Command lines 2, 3 and 4 create the three traffic classes under the queuing discipline 1:0 on interface eth0 of the ingress router. The traffic classes are assigned class identities classid 1:1, 1:2 and 1:3 for IPTV video on demand, video streaming and data traffic respectively on interface eth0 of the ingress router. The traffic type is represented in machine-readable hexadecimal format 0x46 for IPTV video on demand, 0x10 for video streaming and 0x34 for data traffic.

Command lines 5, 6 and 7 ensure that traffic entering the ingress router through the interface identified by the IP address 10.50.1.1 is assigned flow ID's 1:1, 1:2 and 1:3. u32 is a DiffServ filter on the root queuing discipline which will match all IP packets with source address 10.50.1.1 and direct them to the classes defined in lines 2, 3 and 4.

This section described the steps and commands required to set up or enable DiffServ on the Linux routers. It was also illustrated how, using the "tc" command, hexadecimal values used by the Linux kernel to identify the three traffic classes used in the thesis, are mapped to the three classes of service to be used in the DiffServ domain. The following section describes the procedures used on WiMax network to classify traffic into the same DiffServ classes to achieve consistent QoS on the network.

Appendix D

Service provisioning and QoS implementation on WiMax

The WiMax system used in this thesis classifies traffic into four QoS classes namely UGS, rtPS, nrtPS and BE, as described in chapter 2. The three traffic types in the thesis i.e. IPTV video on demand, video streaming and data were classified into UGS, rtPS and BE respectively. Implementation of QoS on the network was based on the inbuilt Service profiles on the system. Assignment of traffic to the defined QoS profiles was achieved via the network management system (NMS) installed on the NMS machine shown in figure 20 of chapter 4.

Details of the implementation of QoS and service provisioning on the system are available on the system manual which can downloaded from the Alvarion website [56]. The system offers two options for prioritizing traffic, 802.1p and DSCP. The latter was used since it is the basis for end-to-end QoS management used in this thesis. The following is a set of the configuration options used on the system to create services and assign them to DiffServ QoS classes. The system provides a hierarchical system of service provisioning which allows the user to enter the service information based on prior settings. The service options fall under QoS profile, which must be entered first, followed by priority classifiers, forwarding rules, services profiles, subscribers and services. Subscriber is the name of the pc that accesses the services on the network. This is recognized on the WiMax network by its MAC address.

Key in these configurations is the QoS type and the priority marking mode. This information is used by the upstream network, i.e. the core network to identify the traffic class and place it into the appropriate DiffServ class. The upstream network therefore respects this marking on the traffic, which is appended to the IP header TOS bits of the traffic.

1. Data service

   QoS profile

   Name: ***bmdata***

   QoS type: ***BE***

   CIR: ***512kbps***

   Priority Classifiers

Name: ***BE data***

Forwarding rules: ***BE***

Service profiles

Name: ***BE data***

Type: ***BMax L2***

Forwarding rule: ***BE***

Priority classifier: ***BE data***

Priority marking mode: ***DSCP***

Priority marking value: ***34***

Maximum number of voice calls: ***1***

Subscribers: ***datapc***

Services

Name: ***bmdata***

Type: ***bmax data***

Service profile: ***BE***

SUMAC add: ***CPEpro***

User: ***data***

2. Video streaming service

QoS profile

Name: ***bmvideo***

QoS type: ***rtPS***

Priority Classifiers

Name: ***rtPS video***

Forwarding rules: ***rtPS***

Service profiles

Name: ***rtPS video***

Type: ***BMax L2***

Forwarding rule: ***rtPS***

Priority classifier: ***rtPS video***

Priority marking mode: ***DSCP***

Priority marking value: ***10***

Maximum number of voice calls: ***1***

Subscribers: ***videopc***

Services

Name: ***bmvideo***

Type: bmax ***video***

Service profile: ***rtPS***

SUMAC add: ***CPEpro***

User: ***video***

3. IPTV video on demand service

QoS profile

Name: ***bmiptv***

QoS type: ***CG***

Priority Classifiers

Name: ***CG iptv***

Forwarding rules: ***CG***

Service profiles

Name: ***CG iptv***

Type: ***BMax L2***

Forwarding rule: ***CG***

Priority classifier: ***CG iptv***

Priority marking mode: ***DSCP***

Priority marking value: ***46***

Maximum number of voice calls: ***1***

Subscribers: ***iptvpc***

Services

Name: ***bmvideo***

Type: bmax ***video***

Service profile: ***rtPS***

SUMAC add: ***CPEpro***

User: ***video***

Appendix E

Application layer performance metrics for video services

Video applications have different QoS requirements on a network depending on their type and codec. Properties of IPTV VoD and video streaming applications used for evaluation of the implemented QoS system are shown in the tables 13, 14 and 15 below.

TABLE 13: VIDEO APPLICATIONS TYPE AND CODES

Application	Type	Codec
Video 1	AVI	XVID H.264
Video 2	MPEG	MPEG-1
Video 3	Matroska	H.264

TABLE 14: VIDEO 1 APPLICATION LAYER PERFORMANCE METRICS-AKEELA

Application	Property	Performance metric
Video	Video type	AVI
	Dimensions	640x272
	Codec	XVID MPEG-4
	Frame rate	24 frames/s
Audio	Codec	MPEG-1 Audio (MP3)
	Sample rate	48kHz
	Bitrate	1162kbps

TABLE 15: VIDEO 2 APPLICATION LAYER PERFORMANCE METRICS-J TIMBERLAKE

Application	Property	Performance metric
Video	Video type	MPEG
	Dimensions	352x240
	Codec	MPEG1 video
	Frame rate	30 frames/s
Audio	Codec	MPEG-1 Audio (MP3)
	Sample rate	44.1kHz
	Bitrate	224kbps

TABLE 16: VIDEO 3 APPLICATION LAYER PERFORMANCE METRICS-EMINEM

Application	Property	Performance metric
Video	Video type	Matroska
	Dimensions	544x416
	Codec	H.264
	Frame rate	30frames/s
Audio	Codec	MPEG-1 Audio layer (MP3)
	Sample rate	48kHz
	Bitrate	62kbps

Appendix F

Iperf results for link quality tests

This appendix presents the Iperf results of the link quality test experiments carried out on the access and core networks. The test results were used to verify the conformance of the access and core networks to the basic requirements of a NGN transport network. The values of throughput, jitter, delay and packet loss obtained were compared to those defined for NGN systems by the ITU-T. The Iperf network testing tool described in chapter 5 was used in the experiments.

F.1 Access network

The following is an output obtained after running iperf in server and client mode between two network points that delimited the access network. The points are identified by the IP addresses of the machines used. All the tests were run for 90 seconds, the ITU-T recommended test time for links.

Throughput

To obtain the throughput of the WiMax network, the following commands were executed. The throughput is recorded on the server machine. Repeated experiments showed the average throughput to be 10.3Mbps. The initial value was always lower than this value and is attributed to connection set up traffic on the WiMax network, which takes up some of the available bandwidth.

server side:

```
wimax@wimax-pc-4:~$ iperf -s -t 90 -i 1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 10.50.1.1 port 5001 connected with 10.50.1.5 port
39712
[  4]  0.0- 1.0 sec    812 KBytes  6.65 Mbits/sec
```

```
[  4]  1.0- 2.0 sec  1.30 MBytes  10.9 Mbits/sec
[  4]  2.0- 3.0 sec  1.21 MBytes  10.2 Mbits/sec
[  4]  3.0- 4.0 sec  1.22 MBytes  10.2 Mbits/sec
[  4]  4.0- 5.0 sec  1.22 MBytes  10.2 Mbits/sec
[  4]  5.0- 6.0 sec  1.22 MBytes  10.2 Mbits/sec
[  4]  6.0- 7.0 sec  1.22 MBytes  10.2 Mbits/sec
[  4]  7.0- 8.0 sec  1.22 MBytes  10.2 Mbits/sec
[  4]  8.0- 9.0 sec  1.22 MBytes  10.2 Mbits/sec
[  4]  9.0-10.0 sec  1.22 MBytes  10.2 Mbits/sec
```

Client side:

```
wimax@wimax-1:~$ iperf -c 10.50.1.1 -t 90 -i 1
------------------------------------------------------------
Client connecting to 10.50.1.1, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  3] local 10.50.1.5 port 39712 connected with 10.50.1.1 port 5001
[  3]  0.0- 1.0 sec  1.27 MBytes  10.7 Mbits/sec
[  3]  1.0- 2.0 sec  1.39 MBytes  11.7 Mbits/sec
[  3]  2.0- 3.0 sec  1.09 MBytes  9.11 Mbits/sec
[  3]  3.0- 4.0 sec  1.32 MBytes  11.1 Mbits/sec
[  3]  4.0- 5.0 sec  1.09 MBytes  9.11 Mbits/sec
[  3]  5.0- 6.0 sec  1.30 MBytes  10.9 Mbits/sec
[  3]  6.0- 7.0 sec  1.31 MBytes  11.0 Mbits/sec
[  3]  7.0- 8.0 sec  1.09 MBytes  9.11 Mbits/sec
[  3]  8.0- 9.0 sec  1.30 MBytes  10.9 Mbits/sec
[  3]  9.0-10.0 sec  1.09 MBytes  9.18 Mbits/sec
[  3] 10.0-11.0 sec  1.30 MBytes  10.9 Mbits/sec
```


- **Jitter and packet loss**

Iperf UDP tests give the results for both jitter and packet loss. By default Iperf uses TCP, the UDP option. Using '-u' after the IP address of the client machine allows the application to run tests in UDP mode. The 's' specifies the server mode. The 'i' option specifies the interval, which is by default 10 seconds. An interval of 1 second was used in the experiments. The '-b' option specifies the required bandwidth of 10Mbps, specified as '10m'. The command must be run on the server side first, then on the client side. The output appears on the server. Results for the first 10 1-second intervals are shown.

Server side:

```
wimax@wimax-1:~$ iperf -s -u -i 1 -b 10m
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:   108 KByte (default)
------------------------------------------------------------
[  3] local 10.50.1.5 port 5001 connected with 10.50.1.1 port 34209
                                                    jitter      packet loss
[  3]  0.0- 1.0 sec  1.07 MBytes  9.01 Mbits/sec  1.610 ms   22/  788 (2.8%)
[  3]  1.0- 2.0 sec  1.07 MBytes  8.98 Mbits/sec  1.700 ms    0/  764 (0%)
[  3]  2.0- 3.0 sec  1.07 MBytes  8.98 Mbits/sec  1.541 ms    0/  764 (0%)
[  3]  3.0- 4.0 sec  1.07 MBytes  9.00 Mbits/sec  1.625 ms   30/  795 (3.8%)
[  3]  4.0- 5.0 sec  1.07 MBytes  8.98 Mbits/sec  1.794 ms   87/  851 (10%)
[  3]  5.0- 6.0 sec  1.07 MBytes  8.98 Mbits/sec  1.655 ms   85/  849 (10%)
[  3]  6.0- 7.0 sec  1.07 MBytes  8.98 Mbits/sec  1.629 ms   86/  850 (10%)
[  3]  7.0- 8.0 sec  1.07 MBytes  8.96 Mbits/sec  1.753 ms   86/  848 (10%)
[  3]  8.0- 9.0 sec  1.08 MBytes  9.02 Mbits/sec  1.589 ms   87/  854 (10%)
[  3]  9.0-10.0 sec  1.07 MBytes  8.98 Mbits/sec  1.756 ms   87/  851 (10%)
[  3] 10.0-11.0 sec  1.07 MBytes  8.96 Mbits/sec  1.699 ms   85/  847 (10%)
```

Client side:

```
wimax@wimax-pc-4:~$ iperf -c 10.50.1.5 -u -i 1 -t 90 -b 10m (10m @10db)
------------------------------------------------------------
Client connecting to 10.50.1.5, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:   108 KByte (default)
------------------------------------------------------------
[  3] local 10.50.1.1 port 34209 connected with 10.50.1.5 port 5001
[  3]  0.0- 1.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  1.0- 2.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  2.0- 3.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  3.0- 4.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  4.0- 5.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  5.0- 6.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  6.0- 7.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  7.0- 8.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  8.0- 9.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  9.0-10.0 sec  1.19 MBytes  10.0 Mbits/sec
```


- **Delay**

The delay across the network is obtained using the ping command as follows:

Server side:

```
wimax@wimax-pc-4:~$ ping 10.50.1.5
PING 10.50.1.5 (10.50.1.5) 56(84) bytes of data.
64 bytes from 10.50.1.5: icmp_seq=1 ttl=64 time=40.8 ms
```

```
64 bytes from 10.50.1.5: icmp_seq=2 ttl=64 time=37.1 ms
64 bytes from 10.50.1.5: icmp_seq=3 ttl=64 time=37.1 ms
64 bytes from 10.50.1.5: icmp_seq=4 ttl=64 time=32.5 ms
64 bytes from 10.50.1.5: icmp_seq=5 ttl=64 time=42.3 ms
64 bytes from 10.50.1.5: icmp_seq=6 ttl=64 time=27.7 ms
64 bytes from 10.50.1.5: icmp_seq=7 ttl=64 time=27.3 ms
64 bytes from 10.50.1.5: icmp_seq=8 ttl=64 time=32.4 ms
64 bytes from 10.50.1.5: icmp_seq=9 ttl=64 time=27.9 ms
64 bytes from 10.50.1.5: icmp_seq=10 ttl=64 time=27.5 ms
```

Ping statistics:

```
--- 10.50.1.5 ping statistics ---
120 packets transmitted, 120 received, 0% packet loss, time
90000ms
rtt min/avg/max/mdev = 27.581/34.332/42.500/3.216 ms
```

F.2 Core network

- **Throughput**

The following output values were obtained on the core network. Readings for the first 10 seconds are shown. The values for the rest for the rest of the duration of 90 seconds remain the same.

Server side:

```
wimax@wimax-1:~$ iperf -s -t 90 -i 1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 10.50.1.5 port 5001 connected with 10.40.1.2 port 32852
[  4]  0.0- 1.0 sec  11.2 MBytes  94.1 Mbits/sec
[  4]  1.0- 2.0 sec  11.2 MBytes  94.1 Mbits/sec
[  4]  2.0- 3.0 sec  11.2 MBytes  94.2 Mbits/sec
[  4]  3.0- 4.0 sec  11.2 MBytes  94.1 Mbits/sec
[  4]  4.0- 5.0 sec  11.2 MBytes  94.1 Mbits/sec
[  4]  5.0- 6.0 sec  11.2 MBytes  94.2 Mbits/sec
[  4]  6.0- 7.0 sec  11.2 MBytes  94.1 Mbits/sec
[  4]  7.0- 8.0 sec  11.2 MBytes  94.2 Mbits/sec
[  4]  8.0- 9.0 sec  11.2 MBytes  94.1 Mbits/sec
[  4]  9.0-10.0 sec  11.2 MBytes  94.2 Mbits/sec
```

Client side:

```
wimax@wimax-pc-3:~$ iperf -c 10.50.1.5 -t 90 -i 1
------------------------------------------------------------
Client connecting to 10.50.1.5, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  3] local 10.40.1.2 port 32852 connected with 10.50.1.5 port 5001
[  3]  0.0- 1.0 sec  12.5 MBytes    105 Mbits/sec
[  3]  1.0- 2.0 sec  11.0 MBytes  92.6 Mbits/sec
[  3]  2.0- 3.0 sec  11.0 MBytes  92.1 Mbits/sec
[  3]  3.0- 4.0 sec  11.6 MBytes  97.6 Mbits/sec
[  3]  4.0- 5.0 sec  11.3 MBytes  95.0 Mbits/sec
[  3]  5.0- 6.0 sec  11.0 MBytes  92.1 Mbits/sec
[  3]  6.0- 7.0 sec  11.5 MBytes  96.4 Mbits/sec
[  3]  7.0- 8.0 sec  11.2 MBytes  93.8 Mbits/sec
[  3]  8.0- 9.0 sec  11.1 MBytes  93.3 Mbits/sec
[  3]  9.0-10.0 sec  11.1 MBytes  93.5 Mbits/sec
[  3] 10.0-11.0 sec  11.0 MBytes  92.6 Mbits/sec
```

- **Packet loss and jitter**

The following commands were run on the client and server machines. The output obtained is as shown.

Server side:

```
wimax@wimax-pc-3:~$ iperf -s -u -i 1
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:   108 KByte (default)
------------------------------------------------------------
[  3] local 10.40.1.2 port 5001 connected with 10.30.1.1 port 37641
[  3]  0.0- 1.0 sec  0.00 Bytes  0.00 bits/sec  0.000 ms    0/    0 (nan%)
[  3]  1.0- 2.0 sec  2.87 KBytes  23.5 Kbits/sec  0.002 ms    0/    2 (0%)
[  3]  2.0- 3.0 sec  1.44 KBytes  11.8 Kbits/sec  0.004 ms    0/    1 (0%)
[  3]  3.0- 4.0 sec  1.44 KBytes  11.8 Kbits/sec  0.007 ms    0/    1 (0%)
[  3]  4.0- 5.0 sec  1.44 KBytes  11.8 Kbits/sec  0.010 ms    0/    1 (0%)
[  3]  5.0- 6.0 sec  1.44 KBytes  11.8 Kbits/sec  0.009 ms    0/    1 (0%)
[  3]  6.0- 7.0 sec  1.44 KBytes  11.8 Kbits/sec  0.011 ms    0/    1 (0%)
[  3]  7.0- 8.0 sec  1.44 KBytes  11.8 Kbits/sec  0.012 ms    0/    1 (0%)
[  3]  8.0- 9.0 sec  1.44 KBytes  11.8 Kbits/sec  0.014 ms    0/    1 (0%)
[  3]  9.0-10.0 sec  1.44 KBytes  11.8 Kbits/sec  0.015 ms    0/    1 (0%)
```

Client side:

```
wimax@wimax-1:~$ iperf -c 10.40.1.2 -u -i 1 -t 90 -b 100
WARNING: delay too large, reducing from 117.6 to 1.0 seconds.
------------------------------------------------------------
Client connecting to 10.40.1.2, UDP port 5001
Sending 1470 byte datagrams
```

```
UDP buffer size:   108 KByte (default)
------------------------------------------------------------
[  3] local 10.30.1.1 port 37641 connected with 10.40.1.2 port 5001
[  3]  0.0- 1.0 sec  1.44 KBytes  11.8 Kbits/sec
[  3]  1.0- 2.0 sec  1.44 KBytes  11.8 Kbits/sec
[  3]  2.0- 3.0 sec  1.44 KBytes  11.8 Kbits/sec
[  3]  3.0- 4.0 sec  1.44 KBytes  11.8 Kbits/sec
[  3]  4.0- 5.0 sec  1.44 KBytes  11.8 Kbits/sec
[  3]  5.0- 6.0 sec  1.44 KBytes  11.8 Kbits/sec
[  3]  6.0- 7.0 sec  1.44 KBytes  11.8 Kbits/sec
[  3]  7.0- 8.0 sec  1.44 KBytes  11.8 Kbits/sec
[  3]  8.0- 9.0 sec  1.44 KBytes  11.8 Kbits/sec
[  3]  9.0-10.0 sec  1.44 KBytes  11.8 Kbits/sec
```

- **Delay**

The ping command is executed on the client machine with the IP address of the egress router as follows:

```
wimax@wimax-1:~$ ping 10.128.7.1
PING 10.128.7.1 (10.128.7.1) 56(84) bytes of data.
64 bytes from 10.128.7.1: icmp_seq=1 ttl=63 time=3.26 ms
64 bytes from 10.128.7.1: icmp_seq=2 ttl=63 time=0.229 ms
64 bytes from 10.128.7.1: icmp_seq=3 ttl=63 time=0.248 ms
64 bytes from 10.128.7.1: icmp_seq=4 ttl=63 time=0.234 ms
64 bytes from 10.128.7.1: icmp_seq=5 ttl=63 time=0.231 ms
64 bytes from 10.128.7.1: icmp_seq=6 ttl=63 time=0.183 ms
64 bytes from 10.128.7.1: icmp_seq=7 ttl=63 time=0.233 ms
64 bytes from 10.128.7.1: icmp_seq=8 ttl=63 time=0.230 ms
64 bytes from 10.128.7.1: icmp_seq=9 ttl=63 time=0.233 ms
```

Delay statistics:

```
--- 10.128.7.1 ping statistics ---
132 packets transmitted, 132 received, 0% packet loss, time 131001ms
rtt min/avg/max/mdev = 0.183/0.253/3.262/0.263 ms
```

F.2 End-to-end link quality test

- **Jitter and packet loss**

To obtain results for jitter and packet loss the following commands are run on the client and server. The bandwidth request at the client side was specified as 10Mbps because this is the maximum link capacity of the access network.

Server side:

```
------------------------------------------------------------
Server listening on UDP port 5001
```

```
Receiving 1470 byte datagrams
UDP buffer size:   108 KByte (default)
------------------------------------------------------------
[  3] local 10.40.1.2 port 5001 connected with 10.50.1.1 port 40885
[  3]  0.0- 1.0 sec  1.05 MBytes  8.82 Mbits/sec  1.517 ms   25/  775 (3.2%)
[  3]  1.0- 2.0 sec  1.08 MBytes  9.02 Mbits/sec  1.607 ms    0/  767 (0%)
[  3]  2.0- 3.0 sec  1.07 MBytes  8.96 Mbits/sec  1.556 ms    0/  762 (0%)
[  3]  3.0- 4.0 sec  1.07 MBytes  9.01 Mbits/sec  1.732 ms   29/  795 (3.6%)
[  3]  4.0- 5.0 sec  1.07 MBytes  8.97 Mbits/sec  1.649 ms   86/  849 (10%)
[  3]  5.0- 6.0 sec  1.07 MBytes  8.97 Mbits/sec  1.571 ms   86/  849 (10%)
[  3]  6.0- 7.0 sec  1.08 MBytes  9.02 Mbits/sec  1.738 ms   87/  854 (10%)
[  3]  7.0- 8.0 sec  1.07 MBytes  8.95 Mbits/sec  1.601 ms   85/  846 (10%)
[  3]  8.0- 9.0 sec  1.08 MBytes  9.03 Mbits/sec  1.704 ms   87/  855 (10%)
[  3]  9.0-10.0 sec  1.07 MBytes  8.96 Mbits/sec  1.685 ms   86/  848 (10%)
```

Client side:

```
wimax@wimax-pc-4:~$ iperf -c 10.40.1.2 -u -i 1 -t 90 -b 10m
------------------------------------------------------------
Client connecting to 10.40.1.2, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:   108 KByte (default)
------------------------------------------------------------
[  3] local 10.50.1.1 port 40885 connected with 10.40.1.2 port 5001
[  3]  0.0- 1.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  1.0- 2.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  2.0- 3.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  3.0- 4.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  4.0- 5.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  5.0- 6.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  6.0- 7.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  7.0- 8.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  8.0- 9.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  9.0-10.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3] 10.0-11.0 sec  1.19 MBytes  10.0 Mbits/sec
```


- **Network throughput**

The following commands were executed to obtain values of the end-to end network throughput.

Server side:

```
wimax@wimax-pc-4:~$ iperf -s -i 1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 10.50.1.1 port 5001 connected with 10.40.1.2 port 44782
[  4]  0.0- 1.0 sec    770 KBytes  6.31 Mbits/sec
[  4]  1.0- 2.0 sec  1.25 MBytes  10.5 Mbits/sec
```

```
[  4]  2.0- 3.0 sec  1.10 MBytes  9.26 Mbits/sec
[  4]  3.0- 4.0 sec  1.10 MBytes  9.24 Mbits/sec
[  4]  4.0- 5.0 sec  1.10 MBytes  9.26 Mbits/sec
[  4]  5.0- 6.0 sec  1.10 MBytes  9.24 Mbits/sec
[  4]  6.0- 7.0 sec  1.10 MBytes  9.24 Mbits/sec
[  4]  7.0- 8.0 sec  1.10 MBytes  9.26 Mbits/sec
[  4]  8.0- 9.0 sec  1.10 MBytes  9.23 Mbits/sec
[  4]  9.0-10.0 sec  1.10 MBytes  9.24 Mbits/sec
```

Client side:

```
wimax@wimax-pc-3:~$ iperf -c 10.50.1.1 -t 90 -i 1
------------------------------------------------------------
Client connecting to 10.50.1.1, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.40.1.2 port 43905 connected with 10.50.1.1 port 5001
[  3]  0.0- 1.0 sec  1.23 MBytes  10.4 Mbits/sec
[  3]  1.0- 2.0 sec  1.16 MBytes  9.76 Mbits/sec
[  3]  2.0- 3.0 sec  1.12 MBytes  9.44 Mbits/sec
[  3]  3.0- 4.0 sec  1.12 MBytes  9.44 Mbits/sec
[  3]  4.0- 5.0 sec  1.27 MBytes  10.7 Mbits/sec
[  3]  5.0- 6.0 sec  1.14 MBytes  9.57 Mbits/sec
[  3]  6.0- 7.0 sec  1.16 MBytes  9.70 Mbits/sec
[  3]  7.0- 8.0 sec  1.14 MBytes  9.57 Mbits/sec
[  3]  8.0- 9.0 sec  1.15 MBytes  9.63 Mbits/sec
[  3]  9.0-10.0 sec  1.26 MBytes  10.6 Mbits/sec
```

- **Delay test**

The end-to-end delay is run between the client end-user client machine and the egress router interface connecting to the IMS. The following command is executed on the client machine:

```
wimax@wimax-pc-4:~$ ping 10.128.7.1
PING 10.128.7.1 (10.128.7.1) 56(84) bytes of data.
64 bytes from 10.128.7.1: icmp_seq=1 ttl=62 time=34.2 ms
64 bytes from 10.128.7.1: icmp_seq=2 ttl=62 time=29.6 ms
64 bytes from 10.128.7.1: icmp_seq=3 ttl=62 time=34.7 ms
64 bytes from 10.128.7.1: icmp_seq=4 ttl=62 time=35.0 ms
64 bytes from 10.128.7.1: icmp_seq=5 ttl=62 time=29.8 ms
64 bytes from 10.128.7.1: icmp_seq=6 ttl=62 time=34.9 ms
64 bytes from 10.128.7.1: icmp_seq=7 ttl=62 time=29.8 ms
64 bytes from 10.128.7.1: icmp_seq=8 ttl=62 time=34.9 ms
```

```
64 bytes from 10.128.7.1: icmp_seq=9 ttl=62 time=30.4 ms
64 bytes from 10.128.7.1: icmp_seq=10 ttl=62 time=35.4 ms
```

Delay statistics:

```
--- 10.128.7.1 ping statistics ---
123 packets transmitted, 123 received, 0% packet loss, time 122003ms
rtt min/avg/max/mdev = 26.982/33.426/40.715/3.062 ms
```

Appendix G

IMS signaling diagrams

This appendix presents a summary of the UE registration on the IMS, and session initiation and control signaling diagram for IPTV VoD applications. Figure 47 shows the registration signaling diagram for the UE [68]. Figure 48 shows the procedure for session initiation and control for an IPTV application.

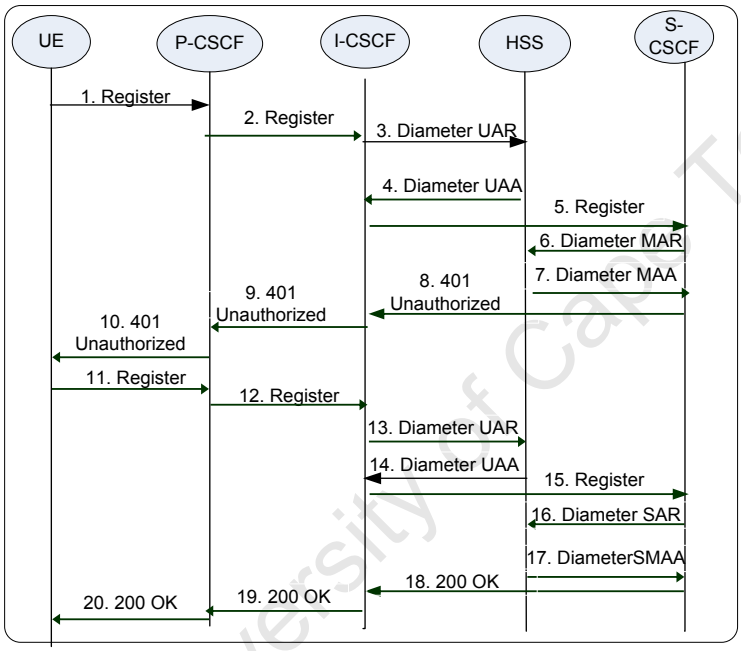


Figure 47: Terminal registration process on the IMS core

Notes:
UAR: User Authentication Request UAA: User Authentication Answer
MAR: Multimedia Authentication Request MAA: Multimedia Authentication

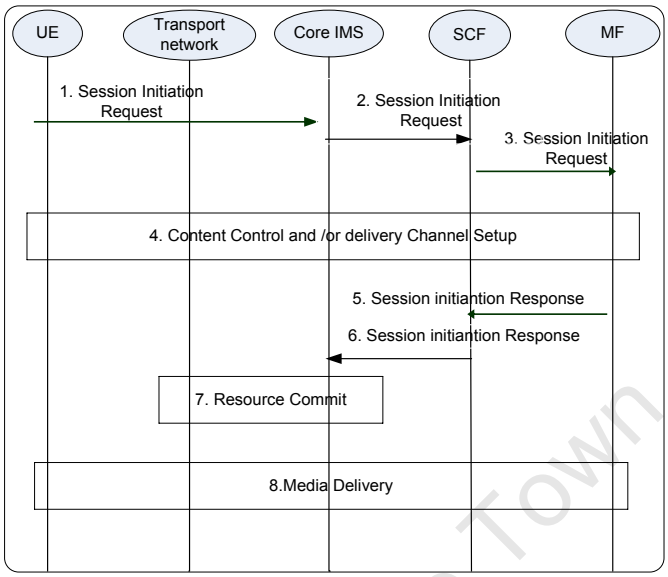


Figure 48: Processes for service initiation, session control and content delivery

Notes:
SCF: Service Control Function MF: Media Function

Appendix H

Details of Machines used on the test bed

- **Core router**

wimax@wimax-pc-2:~$ head /proc/cpuinfo

vendor_id : GenuineIntel

cpu family : 15

model : 2

model name : Intel(R) Celeron(R) CPU 2.40GHz

stepping : 9

cpu MHz :2400.176

cache size : 128 KB

wimax@wimax-pc-2:~$ head /proc/meminfo

MemTotal : 483060 kB

wimax@wimax-pc-2:~$ lsb_release -a

Distributor ID :Ubuntu

Description :Ubuntu 8.04.2

Release :8.04

Codename :hardy

wimax@wimax-pc-2:~$ uname -a

OS Kernel :Linux 2.6.24-23-generic

- **Ingress router**

wimax@wimax-1:~$ head /proc/cpuinfo

vendor_id : GenuineIntel

cpu family : 15

model : 1

model name : Intel(R) Celeron(R) CPU 1.70GHz

stepping : 3

cpu MHz : 1699.855

cache size : 128 KB

wimax@wimax-1:~$ head /proc/meminfo

MemTotal: 513852 kB

wimax@wimax-1:~$ lsb_release -a

No LSB modules are available.

Distributor ID: Ubuntu

Description: Ubuntu 8.04.1

Release :8.04

Codename :hardy

wimax@wimax-1:~$ uname -a

OS Kernel :Linux 1 2.6.24-19-generic

- **Egress router**

wimax@wimax-pc-3:~$ head /proc/cpuinfo

vendor_id : GenuineIntel

cpu family : 15

model : 1

model name : Intel(R) Celeron(R) CPU 1.70GHz

stepping : 3

cpu MHz : 1699.899

wimax@wimax-pc-3:~$ head /proc/meminfo

MemTotal: 513852 kB

wimax@wimax-pc-3:~$ lsb_release -a

Distributor ID:Ubuntu

Description :Ubuntu 8.04.1

Release :8.04

Codename :hardy

wimax@wimax-pc-3:~$ uname -a

OS Kernel :Linux 2.6.24-19-generic

- **End-user Terminals**

Client 1:

Pentium 4 CPU 3.00GHz, 3.00 GHz, 512MBof RAM, operating system- Windows XP

professional version 2002 Service Pack 2.

Client 2:

Intel (R) Celeron (R) CPU 2.4GHz, 2.4GHz, 504MB of RAM, Windows XP professional version2002 Service Pack 3.

Client 3:
wimax@wimax-pc-4:~$ head /proc/cpuinfo

vendor_id : GenuineIntel

cpu family : 15

model : 1

model name : Intel(R) Celeron(R) CPU 1.70GHz

stepping : 3

cpu MHz : 1699.830

wimax@wimax-pc-4:~$ head /proc/meminfo

MemTotal :506748 kB

wimax@wimax-pc-4:~$ lsb_release -a

Distributor ID: Ubuntu

Description: Ubuntu 8.04.3 LTS

Release :8.04

Codename :hardy

wimax@wimax-pc-4:~$ uname -a

OS Kernel :Linux 2.6.24-19-generic

- **WiMax test bed equipment**

  - BreezeMax Micro Base Station (μBST) Indoor Unit (IDU) (Product number: BMAX-MBST-IDU-2CH-AC-3.5).
  - BreezeMax Base Station Outdoor Unit (ODU) with connector for separate antennae (Product number: BMAX-BST-AU-ODU-2CH-3.5a1).
  - 2 x BreezeMax Data Bridge IDU (Product number: BMAX-CPE-IDU-1D).
  - 2 x BreezeMax CPE PRO ODU with connector for separate antennae (Product number: BMAX-CPE-ODU-PRO-SE-3.5).
  - 3 X Agilent 30 decibels (dB) fixed attenuators (Product number: 8495A-001).

- Agilent manual step attenuator 0-70dB (Product number: 8491A-030).

The physical layer parameters of the Wimax access network are listed below.

TABLE 17: WIMAX NETWORK PHYSICAL LAYER PARAMETERS

Parameter	**Value**	**Units**
Operating Frequency	3500	MHz
Bandwidth	3451.75-3481.75	MHz
Maximum Tx Power	20	dBm
Current Modulation Scheme	QAM-6 ¾	n/a
Uplink S/R ratio	17.4	dB
Downlink S/R ratio	18	dB
Uplink RSSI	-85.70	dBm
Downlink RSSI	-85	dBm

Appendix G

The work in this thesis has been accepted for publication in the following conferences:

1. Bessie Malila and Neco Ventura, “End-to-end QoS control for an All-IP (NGN) Platform using WiMax as an Access layer network” *Proceedings of SATNAC,* September 2009.

2. Bessie Malila and Neco Ventura, “An Evaluation of the QoS capabilities of a NGN Test bed”, *Proceedings of the International Conference on Ultra Modern Telecommunications (ICUMT 2009), Petersburg, Russia,* October 2009.

Appendix H

Accompanying CD-ROM

The thesis submission includes an accompanying CD-ROM with the following information:

- Thesis documents – PDF format of the main thesis document and abstract
- Software – QoS system, routing software and test tools used
- Evaluation results – raw data collected during performance evaluation
- Reference material – Some of the publications used in the bibliography
- Published articles – The papers published during the course of this work