Voice Quality Control in Packet Switched Wireless Networks

Nikesh Nageshar

A thesis submitted to the Faculty of Engineering, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Doctor of Philosophy, supervised by Professor Rex van Olst.

Johannesburg, December 2012

Declaration

I declare that this thesis is my own, unaided work, except where otherwise acknowledged or referenced. It is hereby submitted for the degree of Doctor of Philosophy at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other university.

Signed this ____ day of ____________ 2012

____

Nikesh Nageshar

Acknowledgements

The following research was performed under the auspices of the Centre for Telecommunications Access and Services (CeTAS) at the University of the Witwatersrand, Johannesburg, South Africa and Neotel, South Africa. This centre is funded by Telkom SA Limited, Vodacom, Siemens Telecommunications and the Department of Trade and Industries THRIP programme. This financial support was much appreciated.

I thank God for His many blessings and guidance. I thank my supervisor Professor Rex van Olst for his support and guidance for the duration of this research. I also thank my colleagues at CeTAS and friends for their valuable inputs during this research. Finally, I thank my family for their prayers, love, support and patience.

Abstract

Wireless systems have engaged the evolutionary migration from traditional circuit switch technology to packet based technology. Presently all next generation wireless networks have been specified with a packet based Radio Access Networks (RAN), which implies that all the flaws of traditional packet based networks now also apply to voice. These flaws result in decreased speech quality derived from increased latency, jitter and packet loss. This thesis provides the basis for a solution that will facilitate voice quality control in a packet switched wireless network based on the integrated approach of providing Quality of Service (QoS) control across the Admission Control (AC) component, Bearer or Service Flow component and mapping across these components to the appropriate Quality of Service (QoS) metrics at the transport network.

The original contribution of knowledge to the field of electrical and information engineering is the proposal of a Quality of Service (QoS) framework and control mechanisms that result in the transmission of quality voice over a packet switched wireless network autonomous to voice specific signalling or media protocols. These proposals include: Heuristic Analysis in the Admission Control (AC) component; the addition of a voice service class Admission Control (AC) model; selection of a voice specific Bearer or Service Flow and the mapping thereof to a voice specific Quality of Service (QoS) queue or Service Flow at the transport or backhaul network. All these solutions are presented with the goal of ensuring the preservation of quality voice over a packet switched wireless network as governed by network quality metrics such as latency, jitter and packet loss.

This research delivers a comprehensive analysis of 4th Generation (4G) networks such as, Worldwide Interoperability for Microwave Access (WiMAX) and Long Term Evolution (LTE), as specified by the standards bodies yet with focused orientation to the Quality of Service (QoS) framework provided by each of the standards. Specific investigations are targeted towards the Admission Control (AC) and Scheduling of physical resources over the air interface by the Media Access Control (MAC) and Radio Link Control (RLC) layer. Current research and industry led initiatives in the provisioning of quality voice, such as Circuit Switch Fallback (CSFB) and IP Multimedia Subsystem (IMS) are presented and include the associated advantages and disadvantages.

The results and recommendations of this research consist of a multi-faceted solution, commencing with the addition of Heuristic Analysis with Deep Packet Inspection (DPI) being proposed at the eNodeB or WiMAX Base Station (BS) level. An Admission Control (AC) scheme tailored for voice utilising Heuristic Analysis as an input is created, thereafter an identified QoS Class Identifier (QCI) Bearer or Service Flow and transportation Quality of Service (QoS) Identifier for voice is triggered by the User Equipment (UE) application or Bearer initiation procedures. The LTE Bearers and WiMAX Service Flows are tested with the intention of recommending an LTE Bearer and WiMAX Service Flow that will ensure compliance to the minimum required network quality metrics. Finally the testing of the invoking mechanisms is presented mapping the Quality of Service (QoS) metrics across each of the network components thereby completing the solution.

Contents

List of Figures

List of Tables

List of Abbreviations

2G	Second Generation of Cellular Mobile Communication Systems
3G	Third Generation of Cellular Mobile Communication Systems
3GPP	Third Generation Partnership Project
3GPP2	Third Generation Partnership Project 2
4G	Fourth Generation of Cellular Mobile Communication Systems
AC	Admission Control
ACK	Acknowledgement
AMBR	Aggregate Maximum Bit Rate
APN	Access Point Name
ARP	Allocation and Retention Priority
ARQ	Automatic Repeat Request
ASN- GW	Access Service Network Gateway
AS	Application Server
BRAS	Broadband Radius Aggregation Server
BS	Base Station
BSC	Base Station Controller
CDMA	Code Division Multiple Access
CID	Connection Identifier
C-RNTI	Cell Radio Network Temporary Identifier
CS	Convergence Sublayer
CSCF	Call Session Control Function
CSFB	Circuit Switched Fall Back
CSN	Connectivity Service Network
DSCP	Differentiated Services Code Point
DPI	Deep Packet Inspection
EPC	Evolved Packet Core
EPS	Evolved Packet System
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
eNodeB	Evolved Node B
GBR	Guaranteed Bit Rate
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications

HA	Home Agent
HARQ	Hybrid Automatic Repeat Request
HLR	Home Location Register
HSS	Home Subscriber Server
ICSCF	Interrogating - Call Session Control Function
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMS	Internet Protocol Multimedia Subsystem
IP	Internet Protocol
ISC	IMS Service Control
ITU	International Telecommunications Union
LTE	Long Term Evolution
MAC	Media Access Control
MBR	Maximum Bit Rate
MGCP	Media Gateway Control Protocol
MS	Mobile Station
MIMO	Multiple Input Multiple Output
MME	Mobility Management Entity
MOS	Mean Opinion Score
MPLS	Multiprotocol Label Switching
MS	Mobile Station
NACK	Negative Acknowledgement
NAP	Network Access Provider
Non-GBR	Non - Guaranteed Bit Rate
NSP	Network Service Provider
OFDM	Orthogonal Frequency Division Multiplexing Modulation
OFDMA	Orthogonal Frequency Division Multiple Access
OSI	Open Systems Interconnection
OTT	Over the Top
PCC	Policy Charging and Control
PCEF	Policy and Charging Enforcement Function
PCRF	Policy Control and Charging Rules Function
P-CSCF	Proxy Call Session Control Function
PDCCH	Physical Downlink Control Channel

PDCP	Packet Data Convergence Protocol
PDF	Policy Decision Function
PDN	Packet Data Network
PDU	Protocol Data Units
P-GW	Packet Data Network Gateway
PLC	Packet Loss Concealment
PUCCH	Physical Uplink Control Channel
QCI	Quality of Service Class Identifier
QoS	Quality of Service
RAN	Radio Access Network
RoHC	Robust Header Compression
RRM	Radio Resource Management
SAE	System Architecture Evolution
SC-FDMA	Single Carrier – Frequency Division Multiple Access
S-CSCF	Serving Call Session Control Function
SDP	Session Description Protocol
SDU	Service Data Unit
SFID	Service Flow Identifier
SIP	Session Initiation Protocol
SRVCC	Single Radio Voice Call Continuity
SS	Subscriber Station
TDMA	Time Division Multiple Access
TFT	Traffic Flow Templates
UE	User Equipment
UMTS	Universal Mobile Telephone Service
VANC	VoLGA Access Network Controller
VLAN	Virtual Local Area Network
VoLGA	Voice over LTE via Generic Access
W-CDMA	Wideband Code Division Multiple Access
WiMAX	Worldwide Interoperability for Microwave Access (802.16)

1 Voice Quality Control over Packet Switched Wireless Networks

1.1 Introduction

Basic voice telephony has always been considered as a common instrument in both the developing and developed worlds. Over the last decade the African continent has seen a significant growth in universal access primarily driven by expansive wireless networks roll out by mobile operators [1]. Despite the prevalence of such network rollouts a significant gap still exists between the developed and developing worlds in respect of telecommunications access. This gap is exasperated by the population spread in developing worlds as a significant portion of the population in developing countries reside in rural or underserved areas [2] [3].

Whilst the developing world is in need of further prominence with regard to basic telephony access [4] [5] the developed world has moved to a position where voice traffic is considered secondary or rather an application layer service on top of a purely data network. This is probably the logical approach considering the exponential growth of data traffic originating from internet based services, smart phone applications, social networking etc. However considering the requirements of the developing world, basic voice telephony must not take an ancillary position to the applications listed or else the developed world's drive for comprehensive application enablement may result in a greater rift between the developed and developing worlds.

Within the South African context many local operators have deployed extensive Global System for Mobile Communications (GSM), Wideband Code Division Multiple Access (W-CDMA) and Code Division Multiple Access 2000 (CDMA2000) networks to cater for voice. The South African situation echoes that of the rest of the continent; whereby the majority of the population are mobile telephony users. Among these an estimated 40 million subscribers, about 2 million are considered as broadband subscribers with a further 6 million that utilise their mobile telephones for the purposes of e-mail access and basic data [1] [2]. In previous cellular telecommunications standards such as Code Division Multiple Access (CDMA) and Global System for Mobile Communications (GSM), voice was inherently offered over dedicated circuit switched channels whereas in the latter wireless standards video, audio and interactive data services are considered with voice yet without an explicit priority consideration for voice [6].

In contrast to the traditional circuit-switched model of previous cellular standards, Long Term Evolution (LTE) and Worldwide Interoperability for Microwave Access (WiMAX) have been designed to support only packet-switched services. The aim of a packet switched wireless network is to provide seamless Internet Protocol (IP) connectivity between User Equipment (UE) and the Packet Data Network (PDN), without disruption to the end users' applications [7]. In the development of a 4th Generation (4G) network strategy the radio access solution was of a primary consideration as this played a central role in enhancing mobility, service control and the efficient use of network resources. As a result the network architecture called for a 'flat' core network, called the Evolved Packet Core (EPC). The Evolved Packet Core (EPC) features a simplified architecture with open interfaces, higher throughput and lower latency [6] [8].

1.2 Packet Switched Wireless Networks: "The Move to an all-IP Architecture"

Packet switched is a description of the type of network in which small units of data termed packets are routed through a network based on the destination address contained in the header of each packet. Breaking communication down into packets allows the same data channel to be shared among many users in the network. Each end of the conversation is broken down into packets that are reassembled at the other end [7].

Within the wireless realm, 4th Generation (4G) networks have a structure that is packet based termed 'all-IP'. This is where Internet Protocol (IP) packets traverse an access and backbone network without any protocol conversion [9]. Existing circuit switched cellular networks, consist of Base Station (BS) (or base transceiver stations), Base Station Controllers (BSC), Mobile Switching Centres (MSC), gateways, and so on. The Base Station (BS) is responsible for fast power control and wireless Scheduling. The Base Station Controller (BSC) executes the majority of the radio resource management. In contrast, the 4th Generation (4G) network has a simple structure where each Base Station

(BS) operates intelligently to perform radio resource management as well as physical transmission [9] [10].

With consideration to 4th Generation (4G) wireless networks and beyond, the evolution from circuit switch technology to packet switched technology inherently implied that the flaws of both wireless and IP networks functioned concurrently to degrade voice quality. It was therefore imperative to develop quality control structures and methodology to deal with voice over a packet switched network for efficient deployment over next generation wireless networks.

Currently, if an operator chooses to deploy a 4th Generation (4G) or later technology, that operator may be forced to deploy the 4th Generation (4G) or latter technology as a data only network. Alternatively the operator has the option to deploy a fully-fledged IP Multimedia Subsystem (IMS) however IP Multimedia Subsystem (IMS) has been slow to adoption by operators due to its architectural and signalling complexities. Operators have also considered options such as Circuit Switch Fall Back (CSFB), that reverts to the previous generation network in the event that a voice call is made [11] [12]. These interim options however defeat the purpose of convergence and diminish the cost related advantages.

1.3 Research Objectives and Methodology

In order to carry voice of an acceptable quality, physical network resources need to be made available such that the voice traffic class is prioritised above all other traffic classes [13]. The objective of this research is to provide a quality control framework or methodology that will enable quality voice to be carried over a packet switched wireless network. This is accomplished through developing, proposing or utilising any low latency and / or Quality of Service (QoS) features available to ensure an acceptable level of voice quality is maintained in comparison to 2nd Generation (2G) and 3rd Generation (3G) wireless networks. The following focus areas were investigated to determine their bearing on voice quality:

1. Admission Control (AC) and Scheduling techniques in packet switched wireless networks such as WiMAX and LTE.
2. Media Access Control (MAC) and Radio Resource Control (RRC) were examined to determine MAC configurations for low latency, jitter and packet loss.
3. Based on the Media Access Control (MAC) and Radio Link Control (RLC) the standardised Quality of Service (QoS) Bearer or Service Flow framework for WiMAX and LTE was investigated.
4. Invoking of resource allocation and reservation of physical resource blocks across the air interface, backhaul and core networks.
5. Other research initiatives developed for the maintenance of Quality of Service (QoS) for voice.

1.3.1 Research Evaluation Criteria

For the evaluation of voice quality the following criteria can be used to measure the extent of voice quality [6]:

1. Network quality metrics.
2. Objective quality metrics.
3. Subjective quality metric.

Voice quality has a direct relationship to network quality metrics such as available bandwidth, packet loss, packet delay and jitter. Subjective quality metrics relate to a user's perceived voice quality, and objective quality metrics include aspects such as signal to noise ratio, spectral distance metrics and central distance metrics which may be correlated to a Mean Opinion Score (MOS) that is evaluated against an acceptable Mean Opinion Score (MOS) score [6].

For the purposes of this research network quality metrics are used because network quality metrics can be adequately measured and as mentioned have a suitably quantifiable relationship to voice quality. The following network quality metrics are investigated [14]:

1. Bandwidth: a defined quantity of data a transmission link is capable of carrying or is the rated limit defined by the respective standards bodies or manufacturers.

2. Latency or Packet Delay: the time elapsed between packet transmission and its reception at the destination.
3. Packet loss: the discarding of packets due to any possible reason whatsoever. Reasons could be related to network elements or environmental conditions.
4. Jitter: the variation in delay between consecutive packets.

The provisioning of satisfactory Quality of Service (QoS) for voice over an Internet Protocol (IP) network is inherently difficult because of the tight delay, jitter and packet loss requirements for voice traffic. For satisfactory voice transmission an Internet Protocol (IP) network should consist of sufficient bandwidth to carry the coded voice as well as relevant application, transmission and network protocol overheads. The network shall facilitate a less than 0.25% packet loss, a maximum jitter of 5 millisecond and less than 150 millisecond round trip packet delay [14]. These parameters have been determined by relating the network quality metric to objective and subjective voice quality metrics.

It has been established that a greater than 0.25% packet loss, 5 millisecond jitter and /or 150 millisecond delay significantly contributed to speech delay, stutter and break up [14]. Although efficient Packet Loss Concealment algorithms have been created such that voice traffic can withstand a greater than 0.25% packet loss, for the purposes of this research the above listed parameters are used.

The packet based requirements listed above, combined with the nature of radio physical resources make the provisioning of voice over a packet switched wireless network with a fair to perfect Mean Opinion Score (MOS) a challenge [11] [15].

1.3.2 Research Solution

The research solution proposes the following methodology:

1. The priority admittance of voice at the Admission Control (AC) component.
2. Voice quality control on the air interface utilising the most appropriate Radio Link Control (RLC) and Media Access Control (MAC) configuration represented by a Bearer or Service Flow.

3. Logical fixed mapping of the Admission Control (AC) Component to the applicable Bearer or Service Flow component.
4. Voice quality control on the backhaul via logical mapping of the Bearer or Service Flow component to the transport or backhaul network.

The structure of the overall solution allows for Over the Top (OTT) voice to be recognised via Deep Packet Inspection (DPI) at the eNodeB or WiMAX Base Station (BS) level as well as priority admittance of the voice stream at the Admission Control (AC) and Scheduling component of the wireless system. In conjunction with the above, the solution allows for a logical mapping of the admitted voice stream to LTE Quality Class Identifier (QCI) 2 or WiMAX Service Flow enhanced real time Polling Service (ertPS). Furthermore the solution permits the initiation or establishment or the voice specific Bearer or Service Flow via the Heuristic Analysis trigger and finally logical mapping of the Bearer or Service Flow to the transport network. The control aspect of this research is highlighted by the fact that the network operator has a choice or options to shape the Admission Control (AC), Bearer or Service Flow selection to determine the level of Quality of Service (QoS) required for voice traffic. However in terms of this research solution, the control component relates to the evolution from lacklustre voice quality to unsurpassed voice quality in a packet switched wireless network by all potential means attainable within the standards framework.

The advantage of the overall solution is that it emphasises low latency, jitter and packet loss across the entire system. The solution also leverages the proposed introduction of Deep Packet Inspection (DPI) at the eNodeB or WiMAX Base Station (BS) level to provide Quality of Service (QoS) for Over the Top (OTT) voice applications as long as the construct of the voice application can be successfully patterned and blue printed for recognition.

The creation and simulation of a voice specific Admission Control (AC) model illustrated that a voice specific Admission Control (AC) model can be successfully created as long as voice can be successfully prioritised as it enters into the packet switched wireless network. The key to the Admission Control (AC) model is to tailor the model around voice traffic yet not severely affect resource allocation to other types of traffic traversing the wireless network. The selection of the Bearer or Service Flow to be used for voice traffic was of importance because this Bearer or Service was required to support the best network metrics for voice. The Bearer or Service Flow selection was attained via experimental work conducted. The mapping of the Bearer or Service Flow to the transport network was depicted as more of a policy statement based on best practices where best practices dictated that the highest priority classifier be used for voice traffic because such categorisations ensured that voice traffic always received preferential treatment at the relevant transportation or backhaul queuing structure.

Further experimental work also demonstrated that Quality of Service (QoS) Bearer initiation is possible on LTE networks based on the proposed procedures for ordering and initiation of a Quality of Service (QoS) Bearer for voice traffic. These procedures are also available for integration with the Heuristic Analysis voice Admission Control (AC) scheme. This practical solution exhibited quality voice over an LTE network without the addition of an IP Multimedia Subsystem (IMS).

With regard to the testing of the solution, both the Heuristic Analysis approach, the Bearer or Service Flow framework and initiation of these were tested individually because it had to be proven that the best possible configuration for voice across each of the components was found and successfully implementable. It can be noted with confidence that the pairing of each of the components is successful because they deal with separate constituents such as, the Admission Control (AC) component deals with probability of blocking voice circuits yet, the Bearer, Service Flow or transport network deal with network metrics, yet it all has an influence on voice quality control in the packet switched wireless network.

1.4 Contribution to the field

The uniqueness of such an approach is found in the Heuristic Analysis approach to Admission Control (AC) and the introduction of Deep Packet Inspection (DPI) into the access of the wireless network. These options provide operators with greater control over voice applications and the Quality of Service (QoS). Voice quality control in a packet switch wireless network is viable and successful as long as the Quality of Service (QoS) framework proposed is adhered to and successfully implemented.

New information and contribution to the field of electrical engineering, specifically wireless communications systems are as follows:

1. Voice is currently the largest contributor to the revenues of all major operators. The research provides insight to operators for the deployment of voice over packet switched wireless networks at an adequate Quality of Service (QoS).
2. The research provides operators with insight into interim options available for the deployment of voice over LTE and WiMAX networks without overhauling their core voice network as well as maximising their existing investment into 2nd Generation (2G) or 3rd Generation (3G) networks.
3. The contribution from this research will directly relate to how operators provision voice on packet switched wireless networks yet ensuring that quality is maintained. This is done by providing a unique framework illustrating how Quality of Service (QoS) components in the wireless system can be configured. The framework consisted on focus to the recognition of voice traffic, an original voice specific Admission Control (AC) scheme combined with the voice recognition structure.
4. The solution provides a view to the provisioning of a new Heuristic Analysis recognition algorithm at the Admission Control (AC) component. This proposed new scheme utilises Deep Packet Inspection (DPI) filters to recognise Over the Top (OTT) and other voice traffic, and has an input into the admission control structure as well as trigger a Quality of Service Bearer or Service Flow for voice traffic.
5. An original voice specific Admission Control model was specified in combination with the Heuristic Analysis recognition algorithm, but also included was a newly introduced factor *p*. That represented a reserve margin that created a balance for admittance between voice traffic and all other services. Since voice was classified with priority in the new Admission Control model it had the ability to degrade all other services. *p* provided operators with the ability to apply traffic engineering to the Admission Control model.
6. A comprehensive analysis and trial network testing in respect of the Quality of Service (QoS) framework offered by current standard bodies was

conducted resulting in a recommendation that the enhanced real time Polling Service (ertPS) for WiMAX and QoS Class Identifier (QCI) 2 for LTE be used for voice traffic as they exhibited the best latency, jitter and packet loss performance compared to all other Service Flows.

7. In respect of voice traffic traversing multiple network domains a new voice network parameter factor *f* was proposed for introduction where each of the vendor designed system Bearers or Service Flows are categorised with this factor *f*. *f* is classified as the network quality metric parameter for an individual vendor designed system or network domain in relation to the overall voice system network quality metric threshold. f_p represents the sum of the maximum packet loss for a network domain in relation to the overall maximum packet loss, f_j represents the sum of the maximum packet jitter for a network domain in relation to the overall maximum packet loss and f_d represents the sum of the maximum packet delay for a network domain in relation to the overall maximum packet loss. Such a factor *f* shall be very beneficial to system integrators.

8. Several original voice Bearer initiation procedures were presented each utilising the Policy and Charging Rules Function (PCRF) of the LTE system. A part of the original contribution of this work included proposing the assignment of an attribute value by the voice recognition algorithm or any of the voice Bearer initiation procedures above to the Policy and Charging Rules Function (PCRF) via its Rx interface to initiate the Guaranteed Bit Rate Bearer (GBR) with QoS Class Identifier (QCI) 1 for voice traffic.

9. Other original contributions included the manipulation of the Traffic Flow Template (TFT) for the voice Guaranteed Bit Rate Bearer (GBR) such that the filtering mechanism used to classify traffic into the voice Guaranteed Bit Rate Bearer (GBR) was allocated only the Softswitch IP during the voice Guaranteed Bit Rate Bearer (GBR) attach process. This in turn meant that only traffic flowing to the Softswitch IP address would traverse the voice Guaranteed Bit Rate Bearer (GBR).

10. Finally the LTE voice Bearer initiation scheme was tested demonstrating that the philosophy listed above was functional.

1.5 Evidence of Original Contribution

The following papers are submitted as evidence in support of original contributions made by this research:

1. Nageshar N, Van Olst R, "A Heuristic Analysis Approach to Admission Control (AC) for Voice in Packet Switched Wireless Networks," in IEEE Africon, Livingstone, Zambia, September 2011, pp. 1-6.
2. Nageshar N, Van Olst R, "Maintenance of Voice Quality Control in the Evolution to Packet Switched Wireless Networks," in South African Telecommunication Networks and Applications Conference (SATNAC), East London, South Africa, September, 2011.
3. Nageshar N, Van Olst R, "Regulation of Bearer / Service Flow Selection between Network Domains for Voice over Packet Switched Wireless Networks," in ITU Kaleidoscope, Cape Town, South Africa, December, 2011, pp. 175-180.
4. Nageshar N, Van Olst R, "Deep Packet Inspection (DPI) Initiated Quality of Service (QoS) for Over the Top (OTT) Voice in LTE Network," to be submitted for publication.

1.6 Thesis Breakdown

The breakdown of the thesis is as follows; Chapter 2 provides an investigation into both the LTE and WiMAX standards commencing with an explanation of the network architecture of each of the technologies. The network architectures of both standards are deliberated with emphasis on the Radio Link Control (RLC), Medium Access Control, (MAC), Scheduling schemes and Admission Control (AC) components defined in the standards as well as its relationship to voice traffic. Chapter 3 provides a description of Circuit Switch Fallback (CSFB), Voice over LTE Generic Access (VoLGA), Over the Top (OTT) voice and IP Multimedia Subsystem (IMS) including the advantages and disadvantages of each option. Quality of Service (QoS) aware Admission Control (AC) and Scheduling is also described. In Chapter 4 a generic voice Quality of Service (QoS) Framework is presented in conjunction with Admission Control (AC), Admission Control (AC) to Bearer or Service Flow mapping and Bearer or Service Flow to Transportation mapping.

Chapter 5 proposes the establishment of an Admission Control (AC) scheme utilising Heuristic Analysis for the recognition and priority admittance of voice traffic at various levels of the Open Systems Interconnection (OSI) stack [16]. This is further combined with a voice specific variable Admission Control (AC) policy so as to provide a combined Heuristic Analysis and Admission Control (AC) scheme. Chapter 6 highlights the testing of the QoS Class Identifiers (QCI) and Service Flows specified for both LTE and WiMAX networks so as to determine the most appropriate to carry voice traffic. Chapter 7 proposes procedures for the ordering and commencement of a Quality of Service (QoS) Bearer for voice traffic over LTE considering Heuristic Analysis and other methods as well as testing of the voice Bearer initiation procedure. Chapter 8 presents a critical assessment of results obtained, and finally the summary conclusions and further work are presented in Chapter 9.

2 A Review of Packet Switched Wireless Networks

2.1 Introduction

It is essential to understand the organisational design of packet switched wireless networks in order to implement Quality of Service (QoS) for voice over a packet switched wireless network. In the following chapter both the LTE and WiMAX standards are investigated, commencing with an explanation of the network architecture for each of the technological standards. The implementation of Quality of Service (QoS) in a packet switched wireless network is dependent on the configuration of the physical resource layer and the degree to which control of this layer can be exercised. The network architectures of both standards are conversed with emphasis on the Radio Link Control (RLC), Medium Access Control, (MAC), Scheduling schemes and Admission Control (AC) as defined in the standard.

The Radio Link Control (RLC) and Media Access Control (MAC) configuration determine the Quality of Service (QoS) framework in a packet switched wireless network. In the LTE Radio Link Control (RLC) the Radio Link Control (RLC) entities such as acknowledged mode or unacknowledged mode determine the latency, jitter or packet loss thresholds for a specified Bearer similarly the WiMAX classification and classification rules fulfil such a function for a specified Service Flow. In the LTE Media Access Control (MAC) and Scheduling framework, dynamic, persistent or semi-persistent Scheduling differentiates Quality of Service (QoS) configuration, and in WiMAX various Scheduling polling methods contribute to the differentiation in Quality of Service (QoS) configuration.

The Quality of Service (QoS) framework based on the Radio Link Control (RLC), Media Access Control (MAC) and Scheduling for each of the standards are contrasted with a view to investigate and highlight the most viable physical layer configuration that satisfy the network quality metrics for voice traffic.

2.2 Long Term Evolution (LTE) Network Architecture

The LTE network architecture is a culmination of the evolution of radio access via the Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) in conjunction with the evolution of non-radio aspects termed the System Architecture Evolution (SAE). Together the Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) and System Architecture Evolution (SAE) lends itself to the Evolved Packet System (EPS), which uses the concept of Bearers to route IP traffic from a Gateway in the Packet Data Network (PDN) to the User Equipment (UE) [15] [17].

An Evolved Packet System (EPS) Bearer is a flow of IP packets with a defined Quality of Service (QoS) between the Packet Data Network Gateway (P-GW) and the User Equipment (UE). The Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) and Evolved Packet Core (EPC) which is part of the System Architecture Evolution (SAE) together set up and release Bearers as required by the User Equipment (UE) or user application. The Evolved Packet Core (EPC) is responsible for the overall control of the User Equipment (UE) as well as establishment of Bearers [17].

The main logical nodes of the Evolved Packet Core (EPC) are listed as follows:

1. Packet Data Network Gateway (P-GW): is responsible for IP address allocation to the User Equipment (UE), implementation of Quality of Service (QoS) and flow-based charging in accordance with the rules from the Policy Control and Charging Rules Function (PCRF) which is described further below. Based on Traffic Flow Templates (TFT) the Packet Data Network Gateway (P-GW) filters downlink user IP packets into the different Quality of Service (QoS) Bearers. The Packet Data Network Gateway (P-GW) also performs Quality of Service (QoS) enforcement for Guaranteed Bit Rate (GBR) Bearers, and also serves as the mobility anchor for inter-working with non-3rd Generation Partnership Project (non-3GPP) technologies [17] [18].
2. Serving Gateway (S-GW): is located in front of the Packet Data Network Gateway (P-GW) where all user IP packets are transmitted via the Serving Gateway (S-GW). The Serving Gateway (S-GW) serves as the local mobility anchor for the various data Bearers when a User Equipment (UE) moves

between eNodeBs. It also retains the information about Bearers when the User Equipment (UE) is in idle state as well as temporarily buffers downlink data while the Mobility Management Entity (MME) pages the User Equipment (UE) for restoration of Bearers. In addition, the Serving Gateway (S-GW) performs administrative functions for visited networks such as collecting information for charging (e.g. the volume of data sent to or received from the user), and lawful interception [15] [17].

3. Mobility Management Entity (MME): is the control node which processes signalling between the User Equipment (UE) and the core network. The protocols running between the User Equipment (UE) and the core network are known as the Non-Access Stratum (NAS) protocols [17].

In addition to the nodes listed above the Evolved Packet Core (EPC) includes other logical nodes such as the Home Subscriber Server (HSS) and the Policy Control and Charging Rules Function (PCRF) [17].

4. Policy Control and Charging Rules Function (PCRF): is responsible for policy control decision-making, as well as for the control of the flow based charging functionalities which reside in the Packet Data Network Gateway (P-GW). The Policy Control and Charging Rules Function (PCRF) provides the Quality of Service (QoS) authorization (QoS Class Identifier and bitrates) that decides how a certain data flow will be treated and ensures that this is in alignment with the user's profile [17].
5. Home Subscriber Server (HSS): is a combination of the Home Location Register (HLR) and authentication server. The Home Subscriber Server (HSS) is a database that contains subscription data such as the Quality of Service (QoS) profile on a per user basis. It holds information regarding the Packet Data Network Gateways (P-GW) that a user can connect to via an Access Point Name (APN) or any other form. In addition to the information above, the Home Subscriber Server (HSS) also holds dynamic information associated with mobility and authentication such as the identity of the Mobility Management Entity (MME) to which a user is presently attached or registered [17] [18].

Figure 2-1 illustrates the Evolved Packet System (EPS) network elements as described in the definitions above and expresses the interconnection protocol interfaces between the functional elements.

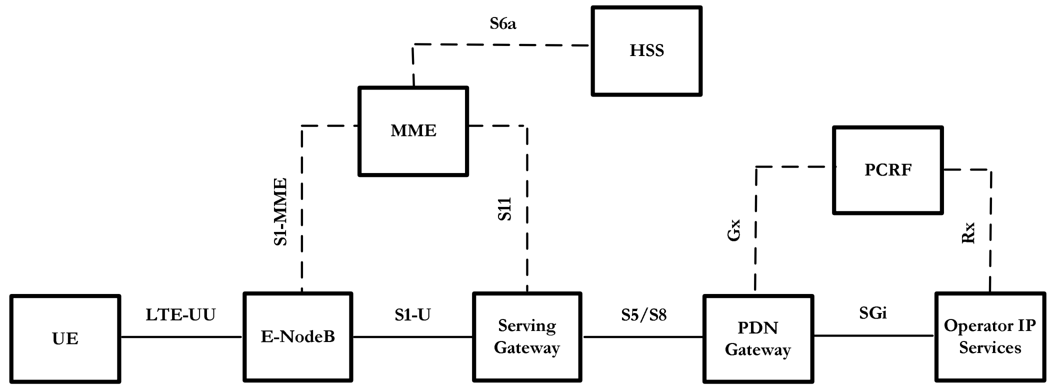

Figure 2-1. Evolved Packet System (EPS) network elements [10]

2.2.1 The LTE Access Network

As LTE is a flat architecture it consists of singular nodes known as the Evolved Node B (eNodeB). In contrast to previous generations of wireless architectures LTE integrates the radio controller function into the eNodeB. The Access Network of the LTE Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) consists of a mesh of eNodeBs interconnected with each other as sketched in Figure 2-2.

The eNodeBs are connected to the Evolved Packet Core (EPC) by means of an interface known as the S1 interface. More specifically to the Mobility Management Entity (MME) via an S1-MME interface and to the Serving Gateway (S-GW) via S1-U interface. The eNodeBs are connected to each other by means of an interface known as the X2 interface [17] [10].

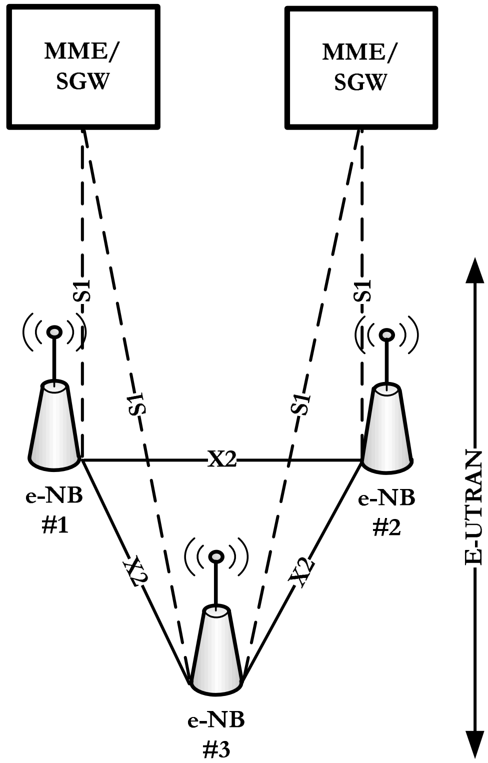

Figure 2-2. LTE Access Network [17]

The 3rd Generation Partnership Project (3GPP) defined protocols on the access layer i.e. between the eNodeBs and the User Equipment (UE) termed Access Stratum protocols. These protocols in conjunction with the Evolved - UMTS Terrestrial Radio Access Network (E-UTRAN) are responsible for all radio-related functions that can be summarized as follows [17]:

1. Radio Resource Management: covers functions related to the radio Bearers, such as Admission Control (AC), Bearer control, radio mobility control, Scheduling and active resource allocation to User Equipment (UE) in both the uplink and downlink directions [17].
2. Header Compression: ensures efficiency use of the radio interface via Robust Header Compression (RoHC) thereby significantly reducing overheads especially for small packets such as Voice over IP (VoIP) packets [17].
3. Security: all data sent over the radio interface is encrypted [17].

4. Connectivity to the Evolved Packet Core: consists of signalling in the direction of the Mobility Management Entity (MME) and the Bearer path towards the Serving Gateway (S-GW) [17].

2.2.2 The LTE Radio Link Control (RLC)

The Radio Link Control (RLC) layer is part of the LTE Access Layer and located between the Packet Data Convergence Protocol (PCDP) layer and the Media Access Control (MAC) layer. The Packet Data Convergence Protocol (PDCP) performs IP header compression and decompression, transfer of user data and maintenance of sequence numbers for radio Bearers. The Radio Link Control (RLC) layer reformats Packet Data Convergence Protocol - Protocol Data Units (PDU) (PDUs) so as to fit them into the size indicated by the MAC layer. The Radio Link Control (RLC) layer communicates with the Packet Data Convergence Protocol (PDCP) layer through a service access point, and with the MAC layer via logical channels [17].

In addition, the Radio Link Control (RLC) reorders the Radio Link Control (RLC) Protocol Data Units (PDU) if they are received out of sequence due to operation of the Hybrid Automatic Repeat Request (HARQ) performed at the MAC layer. This is one of the key differences from previous generation Universal Mobile Telecommunications Systems (UMTS), where the Hybrid Automatic Repeat Request (HARQ) reordering is performed at the MAC layer [17]. The advantage of Hybrid Automatic Repeat Request (HARQ) reordering in the Radio Link Control (RLC) layer is that no additional sequence numbers and reception buffer are needed for Hybrid Automatic Repeat Request (HARQ) reordering because in LTE the Radio Link Control (RLC) sequence number and the Radio Link Control (RLC) reception buffer are used for both Hybrid Automatic Repeat Request (HARQ) reordering and Radio Link Control (RLC) level automatic repeat request operations [17].

It is important to understand that each of the LTE user plane protocol stacks receives a Service Data Unit (SDU) from a higher layer, for which the layer provides a service, and outputs a Protocol Data Unit (PDU) to the layer below. As an example, the Radio Link Control (RLC) layer creates packets which are provided to the layer below, i.e. the MAC layer. The packets which the Radio Link Control (RLC) layer provides to the MAC layer are Radio Link Control (RLC) Protocol Data Units (PDU) from a Radio Link Control (RLC) viewpoint and MAC Service Data Units (MAC-SDU) from a MAC perspective. At the receiving side, the process is reversed, with each layer passing Service Data Units (SDU) to the layer above, where they are received as Protocol Data Units (PDU) [17].

All Radio Link Control (RLC) functions are performed by what is termed as Radio Link Control (RLC) entities which are configured in one of three data transmission modes namely [17]:

1. Transparent Mode.
2. Unacknowledged Mode (UM).
3. Acknowledged Mode (AM).

The eNodeB makes a choice between two modes, namely the Unacknowledged Mode or the Acknowledged Mode based on the Quality of Service (QoS) requirements of the Evolved Packet System (EPS) Bearer [17].

2.2.2.1 Transparent Mode Radio Link Control (RLC) Entity

As is indicated by the name, the Transparent Mode Radio Link Control (RLC) entity is transparent to the Protocol Data Units (PDU) that pass through it i.e. no functions are performed and no Radio Link Control (RLC) overhead is added. Since no overhead is added, a Radio Link Control (RLC) Service Data Unit (SDU) is directly mapped to a Radio Link Control (RLC) Protocol Data Unit (PDU) and vice versa. Therefore, the use of Transparent Mode Radio Link Control (RLC) is very restricted. Only radio resource control messages which do not need Radio Link Control (RLC) configuration can utilise the Transparent Mode Radio Link Control (RLC) such as broadcast system information messages, paging messages and radio resource control messages. Transparent Mode Radio Link Control (RLC) is not used for user plane data transmission in LTE [17].

Each Transparent Mode Radio Link Control (RLC) entity is configured either as a transmitting Transparent Mode Radio Link Control (RLC) entity or as a receiving Transparent Mode Radio Link Control (RLC) entity [17].

2.2.2.2 *Unacknowledged Mode Radio Link Control (RLC) Entity*

Similar to the Transparent Mode Radio Link Control (RLC) entity the Unacknowledged Mode Radio Link Control (RLC) entity provides a unidirectional data transfer service. The Unacknowledged Mode Radio Link Control (RLC) entity is primarily employed by error-tolerant and delay-sensitive real-time applications such as Voice over IP (VoIP). Point-to-multipoint services also use Unacknowledged Mode Radio Link Control (RLC) since no feedback path is required for point-to-multipoint services [17].

The main tasks of Unacknowledged Mode Radio Link Control (RLC) can be summarised as follows [17]:

1. Segmentation and concatenation of Radio Link Control (RLC) Service Data Unit (SDU).
2. Reordering of Radio Link Control (RLC) Protocol Data Units (PDU).
3. Duplicate detection of Radio Link Control (RLC) Service Data Units (SDU).
4. Reassembly of Radio Link Control (RLC) Service Data Units (SDU).

Segmentation and concatenation encompasses the transmitting Unacknowledged Mode Radio Link Control (RLC) entity performing packet segmentation and concatenation on the Radio Link Control (RLC) Service Data Unit (SDU) received from upper layers to form Radio Link Control (RLC) Protocol Data Units (PDU) for the lower layer. The size of the Radio Link Control (RLC) Protocol Data Unit at each transmission opportunity is decided and notified by the MAC layer depending on the radio channel conditions and the available transmission resources, therefore each transmitted Radio Link Control (RLC) Protocol Data Unit can be of differing sizes [17].

As the receiving Unacknowledged Mode Radio Link Control (RLC) entity receives Radio Link Control (RLC) Protocol Data Units (PDU), it first reorders them if received out of sequence. Out of sequence reception is unavoidable due to the Hybrid Automatic Repeat Request (HARQ) operation in the MAC layer making use of multiple Hybrid Automatic Repeat Request (HARQ) processes. Radio Link Control (RLC) Protocol Data Units (PDU) that are received out of sequence are stored in the reception buffer until all the previous Radio Link Control (RLC) Protocol Data Units (PDU) are received, once the correct sequenced Protocol Data Units (PDU) are delivered to the upper layer, the stored Protocol Data Units (PDU) are delivered thereafter. For the duration of the reordering process, any duplicate Radio Link Control (RLC) Protocol Data Units (PDU) received are detected by checking the sequence numbers and are instantaneously discarded [17].

2.2.2.3 Acknowledged Mode Radio Link Control (RLC) Entity

In contrast to the other Radio Link Control (RLC) transmission modes, Acknowledged Mode Radio Link Control (RLC) provides a bidirectional data transfer service hence a single Acknowledged Mode Radio Link Control (RLC) entity is configured with the ability both to transmit and receive. The corresponding parts of the Acknowledged Mode Radio Link Control (RLC) entity are referred to as the transmitting side and the receiving side respectively. An important feature of Acknowledged Mode Radio Link Control (RLC) is retransmission. An Automatic Repeat reQuest (ARQ) operation is performed to support error free transmission. From a Quality of Service (QoS) perspective this function reduces packet loss since transmission errors are corrected by retransmissions. Acknowledged Mode Radio Link Control (RLC) is mainly utilised by error sensitive and delay tolerant non-real time applications [17].

The main functions of Acknowledged Mode Radio Link Control (RLC) can be summarized as follows [17]:

1. Retransmission of Radio Link Control (RLC) Protocol Data Units (PDU).
2. Re-segmentation of retransmitted Radio Link Control (RLC) Protocol Data Units (PDU).
3. Polling of status reports.
4. Status reporting.
5. Status prohibiting.

As previously mentioned, an important function of Acknowledged Mode Radio Link Control (RLC) is retransmission. In order that the transmitting side retransmits only the missing Radio Link Control (RLC) Protocol Data Units (PDU), the receiving side provides a status report to the transmitting side indicating either Acknowledgement (ACK) and / or Negative Acknowledgement (NACK) information for the Radio Link

Control (RLC) Protocol Data Units (PDU) transmitted. Status reports are sent by the transmitting side of the Acknowledged Mode Radio Link Control (RLC) entity to the receiving side [17].

The Acknowledged Mode Radio Link Control (RLC) transmitting side is able to transmit two types of Radio Link Control (RLC) Protocol Data Units (PDU), namely Radio Link Control (RLC) - Data Protocol Data Units (PDU) containing data received from upper layers and Radio Link Control (RLC) – Control Protocol Data Units (PDU) generated in the Acknowledged Mode Radio Link Control (RLC) entity itself. In order to differentiate between Data and Control Protocol Data Units (PDU), a 1-bit flag is included in the Acknowledged Mode Radio Link Control (RLC) header [17].

The transmitting side of the Acknowledged Mode Radio Link Control (RLC) entity can however request a status report from the receiving side, by means of the 1-bit polling indicator included in the Acknowledged Mode Radio Link Control (RLC) header. This function is called Polling as it allows the transmitting side to get the receiver status report. The transmitting side can then use the status reports to select the Radio Link Control (RLC) - Data Protocol Data to be retransmitted, and manage transmission and retransmission buffers efficiently. A typical example where the transmitting side may initiate a Poll is after transmission of the last Protocol Data Unit having been transmitted, or a predefined number of Protocol Data Units (PDU) or data bytes having been transmitted [17].

After the receiving side of the Acknowledged Mode Radio Link Control (RLC) entity obtains a Poll from the peer transmitting side, it checks the reception buffer status and transmits a status report at the first transmission opportunity [17].

2.2.3 Medium Access Control (MAC)

The MAC layer is the lowermost sub-layer in the Layer 2 architecture of the LTE radio protocol stack. The connection to the physical layer below is through transport channels, and the connection to the Radio Link Control (RLC) layer above is through logical channels [17]. The MAC layer performs multiplexing and de-multiplexing between logical channels and transport channels. The MAC layer in the transmitting side constructs

MAC Protocol Data Units (PDU) known as transport blocks from MAC Service Data Unit (SDU) received through logical channels and the MAC layer in the receiving side recovers MAC Service Data Unit (SDU) from MAC Protocol Data Units (PDU) received through transport channels [17].

The MAC layer consists of a Hybrid Automatic Repeat reQuest (HARQ) entity, a multiplexing / de-multiplexing entity and a controller. The Hybrid Automatic Repeat Request entity is responsible for transmit Hybrid Automatic Repeat Request (HARQ) operations that include transmission and retransmission of transport blocks. The Hybrid Automatic Repeat Request (HARQ) entity is also responsible for reception and processing of ACK or NACK signalling. The receive Hybrid Automatic Repeat reQuest (HARQ) operation includes reception of transport blocks, combining of the received data and generation of ACK or NACK signalling [17].

2.2.4 Scheduling

Scheduling allows for the appropriate allocation of physical resource blocks for on-going calls in a packet switched wireless network. The scheduler in the eNodeB distributes available radio resources in a cell amongst attached User Equipment (UE) and the radio Bearers of each of the User Equipment (UE) [17]. The selection of the Scheduling algorithm is left to the eNodeB based on well-defined Scheduling schemes. In principle, the eNodeB allocates downlink or uplink radio resources to User Equipment (UE) based respectively on the downlink data buffered in the eNodeB and on buffer status reports that the eNodeB receives from the User Equipment (UE). In this process, the eNodeB makes reference to the Quality of Service (QoS) requirements of each configured Bearer and from this selects the size of the MAC Protocol Data Unit required [10] [17].

The Scheduling schemes that are proposed by the 3rd Generation Partnership Project (3GPP) in LTE networks are listed as follows [13]:

1. Dynamic Scheduling.
2. Persistent Scheduling.
3. Semi-persistent Scheduling.

The normal mode of Scheduling is Dynamic Scheduling, which is selected by means of downlink assignment messages for the allocation of downlink transmission resources and uplink grant messages for the allocation of uplink transmission resources. Considering Dynamic Scheduling, the User Equipment (UE) sends a resource request for every packet or retransmission and the Base Station (BS) allocates uplink resources for every packet or retransmission separately. The drawback of Dynamic Scheduling is the large amount of signalling required [9] [13]. They are transmitted on the Physical Downlink Control Channel (PDCCH) using a Cell Radio Network Temporary Identifier (C-RNTI) to identify the intended User Equipment (UE) [17].

In addition to the Dynamic Scheduling, Persistent Scheduling is also defined. Persistent Scheduling supports radio resources to be semi-statically configured and allocated to a User Equipment (UE) for a longer time period than one sub-frame thus avoiding the need for specific downlink assignment messages or uplink grant messages over the Physical Downlink Control Channel (PDCCH) for each and every sub-frame [17]. Persistent Scheduling may be useful for services such as Voice over IP (VoIP) for which the data packets are small, periodic and semi-static in size however wasteful because of the voice activity factor. The voice activity factor is the ratio of actual speech in comparison to silence in a conversation. For this kind of service the timing and amount of radio resources needed are predictable. Thus the overhead of the Physical Downlink Control Channel (PDCCH) is significantly reduced compared to the case of Dynamic Scheduling however Persistent Scheduling is radio resource inefficient [17].

The principle of Semi-persistent Scheduling consists of Persistent Scheduling for initial transmissions and Dynamic Scheduling for retransmissions [9] [13]. Semi-persistent Scheduling reduces the total amount of signalling required however it does not perform efficiently in high packet loss scenarios.

2.2.4.1 Scheduling Information Transfer

The assumption underpinning Scheduling in LTE networks is that radio resources are only allocated for communication to or from User Equipment (UE) if data is available to be sent or received. Buffer status reports from the User Equipment (UE) to the eNodeB are used to assist the allocation of uplink radio resources by the eNodeB. In the downlink direction the scheduler in the eNodeB is aware of the amount of data to be delivered to each User Equipment (UE) however, in the uplink direction because the Scheduling decisions are performed in the eNodeB and the buffer for the data is located in the User Equipment (UE), buffer status reports have to be sent from the User Equipment (UE) to the eNodeB indicating the amount of data at the User Equipment (UE) that needs to be transmitted over the uplink scheduler [17].

There are two types of buffer status reports defined in LTE, namely:

1. A long buffer status report.
2. A short buffer status report.

Either is transmitted depending on the amount of uplink transmission resources available for distribution of the buffer status reports, the number of groups of logical channels that consist of non-empty buffers or whether a specific event is triggered at the User Equipment (UE) [17].

The long buffer status report notifies the amount of data for four logical channel groups whereas the short buffer status report notifies the amount of data for only one logical channel group [17]. Although the User Equipment (UE) might actually have more than four logical channels configured, the overhead would be large if the amount of data at the User Equipment (UE) were to be reported for every logical channel individually hence grouping the logical channels into four groups represents a compromise between efficiency and accuracy in terms of reporting [17].

A buffer status report can be triggered in the following situations [17]:

1. When data arrives for a logical channel which has a higher priority than the logical channels whose buffers previously contained data.
2. When a certain time has elapsed since the last transmission of a buffer status report.
3. When the serving cell changes.

If a User Equipment (UE) is not allocated with enough uplink Scheduling resources to send a buffer status report, either a single bit Scheduling request is sent over the Physical Uplink Control Channel (PUCCH), or the random access procedure is performed to request an allocation of an uplink radio resource in order to send a buffer status report. Thus LTE provides suitable signalling to ensure that the eNodeB has sufficient information about the data waiting at each User Equipment (UE) uplink transmission buffer so as to allocate corresponding uplink transmission resources in a well-timed manner [17].

2.3 LTE Quality of Service (QoS) Framework

The configurations listed above in the Radio Link Control (RLC), MAC and Scheduling all contribute to the creation of the LTE Quality of Service (QoS) Framework. These configurations have a direct relationship to the Evolved Packet System (EPS) Bearer listed below.

The LTE Evolved Packet System (EPS) is constructed on a packet flow that is established between the Packet Data Network Gateway (P-GW) and the User Equipment (UE). LTE uses separate service data flows that are mapped to corresponding Bearers with a common Quality of Service (QoS) treatment [12]. A service data flow within a Bearer is assigned a QoS Class Identifier (QCI) that refers to a set of packet forwarding treatments (e.g. Scheduling weights, admission thresholds, queue management thresholds, radio link layer protocol configuration and MAC configuration as previously described) preconfigured for each network element [12]. The mapping of a service data flow to a Bearer is currently classified by IP five-tuple based packet filter as illustrated in Figure 2-3. This is either provisioned by the Policy and Charging Rules Function (PCRF) or defined by certain application layer signalling [12].

Bearers can be classified into two categories based on the nature of the Quality of Service (QoS) they provide [10] [17]:

1. Minimum Guaranteed Bit Rate (GBR) Bearers: can be used for applications such as Voice over IP (VoIP). These have a related Guaranteed Bit Rate (GBR) value

for which dedicated transmission resources are permanently allocated at Bearer establishment or modification [10] [17].

2. Non - Guaranteed Bit Rate (Non-GBR) Bearers: does not guarantee any particular bit rate and can be used for applications such as web browsing or File Transfer Protocol (FTP) transference. For these Bearers no bandwidth resources are assigned on a permanent basis to the Bearer [10] [17].

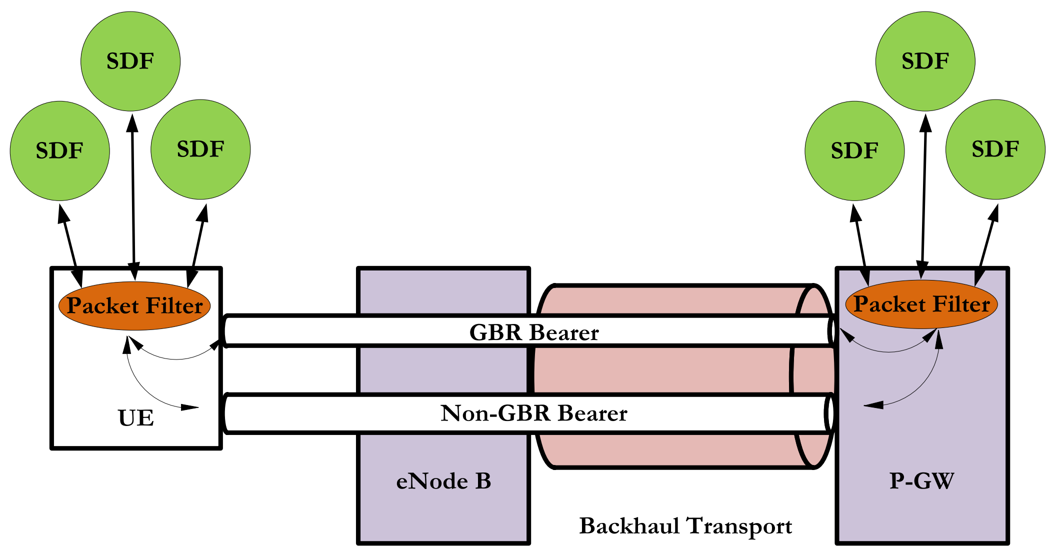

Figure 2-3. LTE Service Data Flow to Bearer Mapping

In the Access Network it is the responsibility of the eNodeB to ensure the preservation of necessary Quality of Service (QoS) for a Bearer over the radio interface. Each Bearer has an associated QoS Class Identifier (QCI) and an Allocation and Retention Priority (ARP). Each QoS Class Identifier (QCI) is characterized by the priority handling, packet delay budget and acceptable packet loss rate as described in Table 2-1. A small range of QoS Class Identifiers (QCI) has been standardised such that vendors have a uniform understanding of the underlying service characteristics [17]. This ensures that an LTE operator can expect uniform traffic-handling behaviour throughout the network regardless of the manufacturer of the eNodeB equipment [17].

The priority and packet delay budget (and to some extent the acceptable packet loss rate) from the QoS Class Identifier (QCI) label inherently determine the Radio Link Control (RLC) mode configuration chosen as well as the treatment of packets sent over the

Bearer by the MAC scheduler in terms of Scheduling policy, queue management policy and rate shaping policy. As an example, a packet with a higher priority can be expected to be scheduled before a packet with lower priority. Considering Bearers with a low acceptable loss rate an Acknowledged Mode (AM) entity can be used within the Radio Link Control (RLC) layer to ensure that packets are delivered successfully across the radio interface [10] [17].

Table 2-1. Description of QoS Class Identifier (QCI) characteristics [12] [15]

QCI	Resource Type	Priority	Packet Delay Budget	Packet Error Loss Rate
1	GBR	2	100ms	10^{-2}
2	GBR	4	150ms	10^{-3}
3	GBR	3	50ms	10^{-3}
4	GBR	5	300ms	10^{-6}
5	Non-GBR	1	100ms	10^{-6}
6	Non-GBR	6	300ms	10^{-6}
7	Non-GBR	7	100ms	10^{-3}
8	Non-GBR	8	300ms	10^{-6}
9	Non-GBR	9	300ms	10^{-6}

Aside from the QoS Class Identifier (QCI) and the LTE Bearers listed, the following Quality of Service (QoS) attributes are also associated with the LTE Bearer [12] [17]:

1. Allocation and Retention Priority (ARP): is used by call Admission Control (AC) and overload control for control plane treatment of a Bearer. The Allocation and Retention Priority (ARP) of a Bearer is used to decide whether or not a requested Bearer should be established in case of radio congestion. It also governs the prioritization of the Bearer with respect to a new Bearer establishment request. Once a Bearer has been successfully established the Bearers Allocation and

Retention Priority (ARP) value does not have any impact on the Bearer level packet treatment such as Scheduling and rate control [12] [17].

2. Maximum Bit Rate (MBR): is associated with the Guaranteed Bit Rate (GBR) Bearer and sets up an upper limit on the bit rate that can be anticipated from a Guaranteed Bit Rate (GBR) Bearer. The reason for such a parameter is that bit rates higher than the Guaranteed Bit Rate (GBR) may be allowed for a Guaranteed Bit Rate (GBR) Bearer if resources are available [12] [17].
3. Aggregate Maximum Bit Rate (AMBR): makes reference to the aggregate sum of the bit rates for a group of Non - Guaranteed Bit Rate (Non-GBR) Bearers [12] [17].

2.3.1 LTE Bearer Interface Mapping

An Evolved Packet System (EPS) Bearer has to cross multiple interfaces as presented in Figure 2-4 namely; the radio interface (also known as the LTE-Uu interface) from the User Equipment (UE) to the eNodeB, the S1 interface from the eNodeB to the Serving Gateway (S-GW) and the S5/S8 interface from the Serving Gateway (S-GW) to the Packet Data Network Gateway (P-GW).

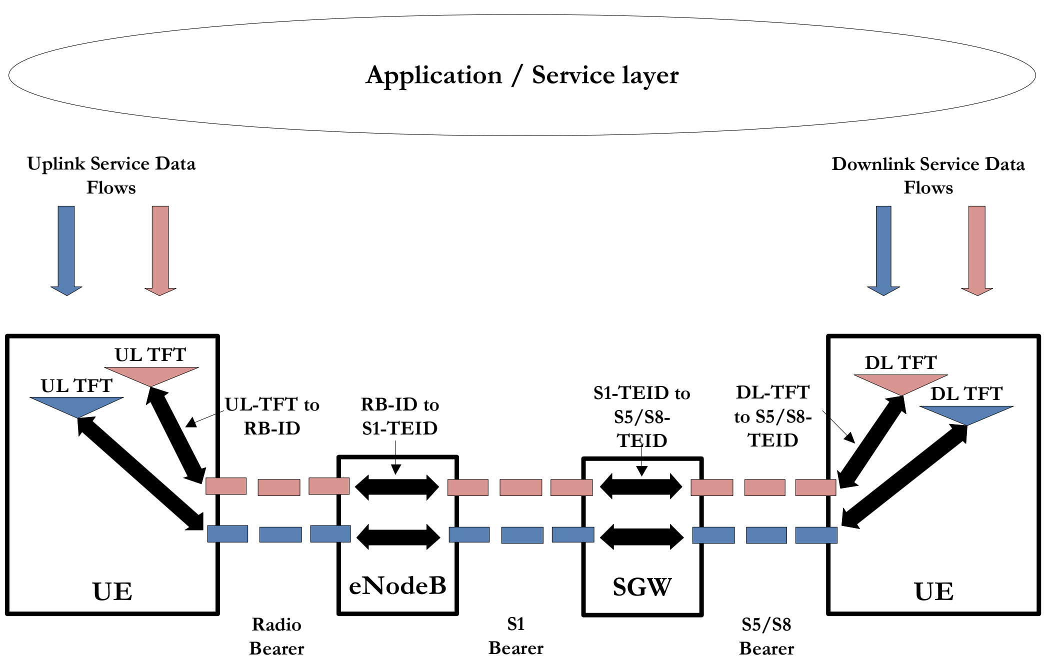

Figure 2-4. LTE/SAE Bearers across the different interfaces [10]

Across each of the interfaces the Evolved Packet System (EPS) Bearer is mapped onto a lower layer Bearer, each with its own Bearer identification. Each node keeps track of the binding between the Bearer identifications across its different interfaces [17].

2.3.2 LTE Default Bearer Establishment

The attaching of a User Equipment (UE) to the LTE network is defined by the User Equipment (UE) Attach procedure. This is where the User Equipment (UE) is assigned an IP address by the Packet Data Network Gateway (P-GW) and at least one Bearer is established. This Bearer is called the Default Bearer and remains established throughout the connection lifespan of the User Equipment (UE) to the Packet Data Network Gateway (P-GW) so as to provide the User Equipment (UE) with constant IP connectivity to that Packet Data Network Gateway (P-GW) [17]. Based on subscription data that is retrieved from the Home Subscriber Server (HSS) the default Bearer Quality of Service (QoS) is assigned by the Mobility Management Entity (MME). The Packet Data Network Gateway (P-GW) may change these values after interaction with the Policy and Charging Rules Function (PCRF) or to match local configuration. The Default Bearer is always a non-Guaranteed Bit Rate (GBR) Bearer as it is permanently established.

Dedicated Bearers can be established at any time during or after the attach procedure. A dedicated Bearer can be either a Guaranteed Bit Rate (GBR) or a non-Guaranteed Bit Rate (GBR) Bearer [17]. In the case of multiple Bearers, packet filtering is applied such that traffic is filtered into different Bearers based on a pre-set Traffic Flow Template (TFT). The Traffic Flow Templates (TFT) use IP header information such as source and destination IP addresses and port numbers to filter packets such as Voice over IP (VoIP) from web browsing traffic so that each can be sent down the respective Bearers with the appropriate Quality of Service (QoS) treatment applied.

An uplink Traffic Flow Template (TFT) associated with each Bearer in the User Equipment (UE) filters IP packets to Evolved Packet System (EPS) Bearers in the uplink direction. A downlink Traffic Flow Template (TFT) in the Packet Data Network Gateway (P-GW) is a similar set of downlink packet filters. Figure 2-5 illustrates a Traffic Flow Template (TFT) for the FTP protocol [10] [17].

□ EPS Bearer Level Traffic Flow Templates

type: EPS Bearer Level Traffic Flow Templates (Bearer TFT) (0x54)

length: 11

0000 …. = spare: 0x00

…. 0000 = instance: 0x00

Traffic Flow Templates IEI: 0x00

Length of Traffic Flow Templates IE: 11

001. …. = TFT operation code: Create new TFT (0x01)

…0 …. = Is there a parameter list? no

…. 0001 = Number of packet filters: 1

□ Packet Filter

…. 0000 = Packet filter identifier: 0

.. 00 …. = Packet filter direction: pre Rel-7 TFT filter (0)

Packet filter evaluation precedence: 0

Length of packet filter contents: 5

Packet filter component type identifier: Remote port range type (0x51)

Remote range-low port number: 20

Remote range-high port number: 21

Figure 2-5. LTE Traffic Flow Templates (TFT) Example [19]

A dedicated Bearer is always associated with a Traffic Flow Template (TFT) however a default Bearer may or may not be associated with a Traffic Flow Template (TFT). There exists an evaluation packet precedence index in the packet filter which is used by User Equipment (UE) to search for a match (to map the application traffic). Once the User Equipment (UE) finds a match it uses that particular packet filter to transmit the data. If there is no match the User Equipment (UE) transmits the data on a Bearer to which no Traffic Flow Templates (TFT) have been assigned [17] [19].

2.3.3 Dedicated Bearer Establishment Procedure

This subsection provides a description of the Bearer establishment technique across the LTE network nodes as presented in Figure 2-4. In LTE the decision to establish or modify a dedicated Bearer is taken by Evolved Packet System (EPS) after which Quality of Service (QoS) parameters are assigned by the Evolved Packet System (EPS) Bearer. These values are not modified by Mobility Management Entity (MME) but rather forwarded transparently to the Evolved - UMTS Terrestrial Radio Access Network (E-

UTRAN). The Mobility Management Entity (MME) may however reject the establishment of dedicated Bearer if there are any discrepancies identified [17]. Figure 2-6 provides an explanation of the LTE dedicated Bearer establishment procedure.

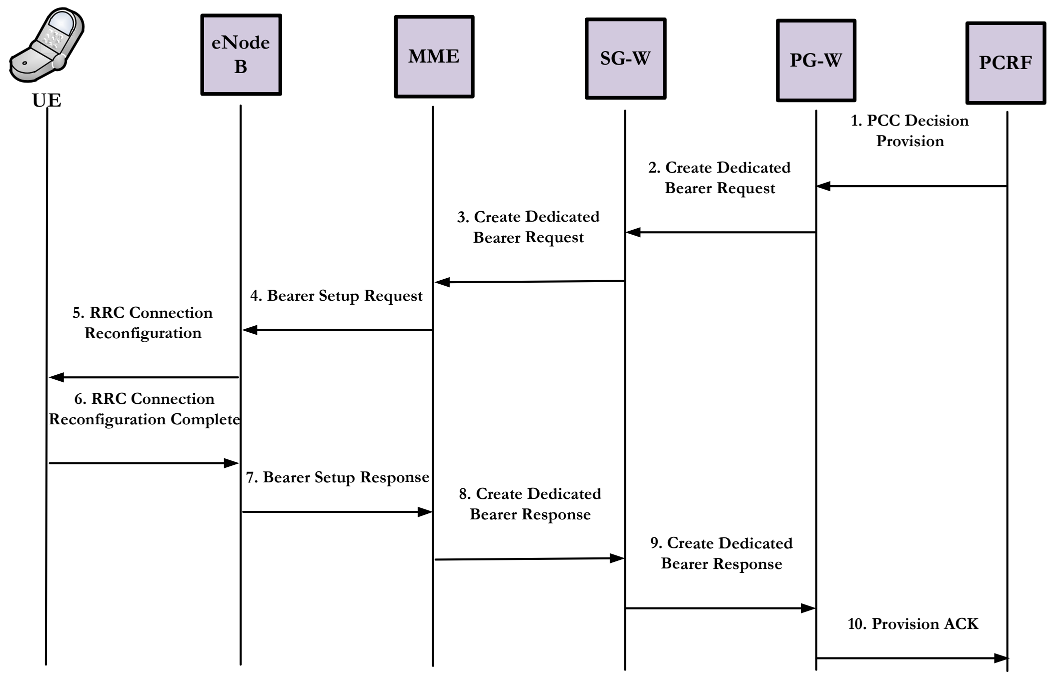

Figure 2-6. Message Flow for LTE Dedicate Bearer Establishment

1. The Policy and Charging Rules Function (PCRF) forwards a Policy Control and Charging (PCC) Decision Provision message to the Packet Data Network Gateway (P-GW) specifying the required Quality of Service (QoS) for the Bearer terminating on the Packet Data Network Gateway (P-GW).
2. The Packet Data Network Gateway (P-GW) uses this Quality of Service (QoS) policy to assign the Bearer Quality of Service (QoS) parameters. The Packet Data Network Gateway (P-GW) then sends a Create Dedicated Bearer Request message including the Quality of Service (QoS) and Uplink Traffic Flow Templates (TFT) to be used by the User Equipment (UE) towards the Serving Gateway (S-GW) [17].
3. The Serving Gateway (S-GW) receives the Create Dedicated Bearer Request message, including Bearer Quality of Service (QoS), Uplink Traffic Flow

Templates (TFT) and S1-Bearer identification and thereafter forwards this to the Mobility Management Entity (MME) [17].

4. The Mobility Management Entity (MME) constructs a set of session management configuration information, including the Uplink Traffic Flow Templates (TFT) and the Evolved Packet System (EPS) Bearer identity, all of which are included it in the Bearer Set-up Request message that it forwards to the eNodeB [17].
5. The Bearer Set-up Request passes the Quality of Service (QoS) indicators of the Bearer to the eNodeB. This information is used by the eNodeB for call Admission Control (AC), as well as to ensure the appropriate Scheduling of the User Equipment's (UE) IP packets. The eNodeB maps the Evolved Packet System (EPS) Bearer Quality of Service (QoS) to the radio Bearer Quality of Service (QoS).
6. The eNodeB then signals a RRC Connection Reconfiguration message (which includes the radio Bearer Quality of Service (QoS) indicators, session management configuration and Evolved Packet System (EPS) radio Bearer identity) to the User Equipment (UE) to set up the radio Bearer [17].
7. The RRC Connection Reconfiguration message contains all the configuration parameters for the radio interface. This is primarily for the configuration of the Layer 2 parameters such as the Packet Data Convergence Protocol, Radio Link Control (RLC) and Media Access Control (MAC) parameters. Messages 6 to 10 are the corresponding response messages to confirm that the Bearers have been set up correctly [17].

This subsection concludes the Quality of Service (QoS) mechanisms available to the LTE standard as defined and standardised by the 3rd Generation Partnership Project (3GPP) body.

2.4 Worldwide Interoperability for Microwave Access (WiMAX) Network Architecture

The WiMAX architecture as developed by the WiMAX forum is based on the IEEE 802.16 family of standards. This portrays a fully converged, unified network architecture created to support fixed and nomadic services. The WiMAX reference model is shown in Figure 2-7.

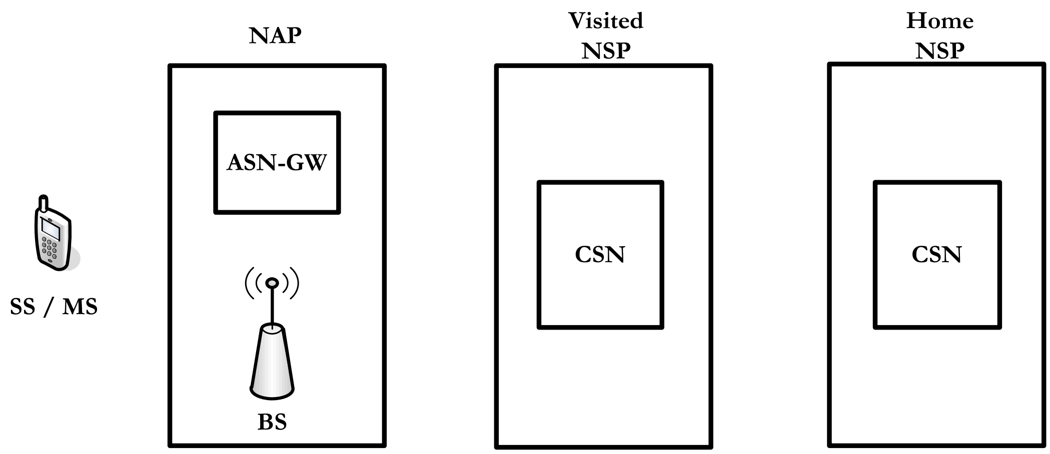

Figure 2-7. WiMAX Reference Model [8]

The WiMAX network architecture is based on a flat 'all-IP' model and comprises of the following major network elements [8]:

1. Subscriber Station (SS) or Mobile Station (MS). This comprises of all devices, such as cell phones, wireless laptops and software needed for communication with a WiMAX network [8].
2. Network Access Provider (NAP). This offers radio access functionality such as, Access Service Network Gateway (ASN-GW), IEEE 802.16 interface with network admission and handover and Base Station (BS) [8].
    a. The Base Station (BS) is responsible for providing the air interface to the Mobile Station (MS). Additional functions that may be part of the Base Station (BS) are handoff triggering, tunnel establishment, radio resource

management, Quality of Service (QoS) policy enforcement, traffic classification, Dynamic Host Control Protocol (DHCP) proxy, key management, session management and multicast group management [8].

b. The Access Service Network Gateway (ASN-GW) typically acts as a layer 2 traffic aggregation point. Additional functions that may be part of the Access Service Network Gateway (ASN-GW) include; intra access service network location management and paging, radio resource management, Admission Control (AC), caching of subscriber profiles, encryption keys, Authentication Authorization Accounting (AAA) functionality, management of mobility tunnels with Base Station (BS), Quality of Service (QoS) policy enforcement and foreign agent functionality for mobile IP [8].

3. Network Service Provider (NSP) provides IP connectivity services. This contains the logical representation of functions such as; Connectivity Service Network (CSN), Home Agent (HA), AAA servers, internet connectivity, IP address management, mobility and roaming between Access Service Networks. A Network Service Provider (NSP) may have a contract with another Network Service Provider (NSP) and may also have contracts amongst multiple Network Service Providers (NSP) [8].

    a. Connectivity Service Network (CSN). The Connectivity Service Network (CSN) provides connectivity to the Internet, Access Service Providers (ASP), other public networks and corporate networks. The Connectivity Service Network (CSN) is owned by the Network Service Provider (NSP) and includes AAA servers that support authentication for the devices, users and specific services. The Connectivity Service Network (CSN) provides per user policy management of Quality of Service (QoS) and security. The Connectivity Service Network (CSN) is also responsible for IP address management, support for roaming between different Network Service Providers (NSP), location management between Access Networks and mobility and roaming between Access Networks.

    b. Home Agent (HA). This works in conjunction with a Foreign Agent (FA) such as the Access Service Network Gateway (ASN-GW) to provide an efficient end-to-end Mobile IP solution. The Home Agent (HA) serves as an anchor point for subscribers providing secure roaming with Quality of Service (QoS) capabilities.

c. Authentication, Authorisation and Accounting Server (AAA). This is included within the Connectivity Service Network (CSN) for wireless system subscription services requiring Authentication, Authorisation and Accounting (AAA).

2.4.1 WiMAX Radio Link Control (RLC)

The WiMAX Radio Link Control (RLC) component consists of the service specific Convergence Sublayer and the MAC Layer. The Convergence Sublayer accepts Protocol Data Units (PDU) from the higher layers and transmits them to the MAC Common Part Sublayer where classical type MAC procedures are applied [8]. A diagrammatical representation of the WiMAX protocol stack is listed in Figure 2-11 below.

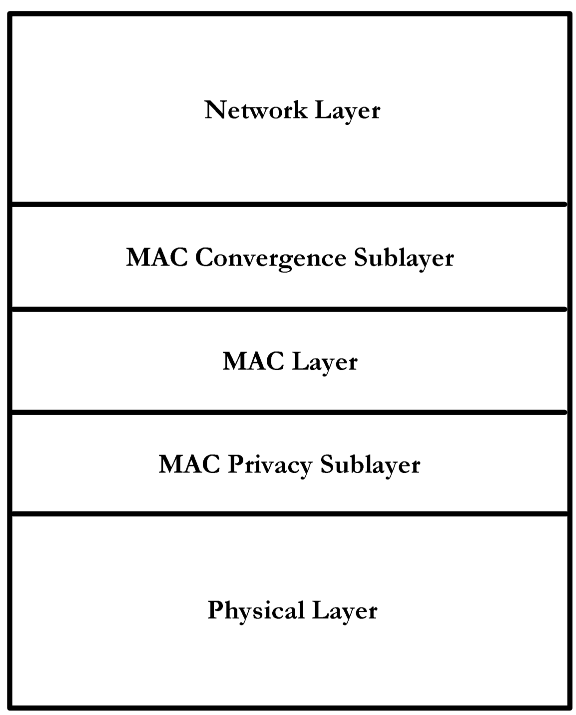

Figure 2-8. WiMAX Protocol Stack [8] [20]

The Convergence Sublayer specification illustrated in the WiMAX standard is also called the Packet Convergence Sublayer. The Packet Convergence Sublayer is used for the transport of all packet based protocols such as the Internet Protocol (IP), Point-to-Point Protocol (PPP) and Ethernet. The Packet Convergence Sublayer resides on top of the IEEE 802.16 MAC and performs the following functions utilizing the services of the MAC [8] [20]:

1. Classification of Protocol Data Units (PDU) from the higher layers into an appropriate transport connection.
2. Delivery of resulting Convergence Sublayer Protocol Data Unit (PDU) to the MAC service access point associated with the Service Flow for transport to the peer MAC service access point.
3. Receipt of the Convergence Sublayer Protocol Data Unit (PDU) from the peer MAC service access point.
4. Suppression of payload header information (optional).
5. Rebuilding of any suppressed payload header information (optional).

The sending Convergence Sublayer is responsible for delivering the MAC Service Data Unit (SDU) to the MAC service access point.

2.4.2 Media Access Control

The WiMAX MAC layer is classified as an adaptation layer between the physical layer and the upper layers of the Open Systems Interconnection (OSI) stack. The MAC layer receives MAC Service Data Units (SDU) from the layer above, aggregates and encapsulates these into MAC Protocol Data Units (PDU). The MAC layer then passes the MAC Protocol Data Units (PDU) to the physical layer [8]. Similarly, the WiMAX MAC layer takes MAC Protocol Data Units (PDU) from the physical layer, removes encapsulation and reorders the MAC Protocol Data Units (PDU) into MAC Service Data Unit (SDU) and passes the MAC Service Data Unit (SDU) to the protocol layers above [8].

The WiMAX MAC layer has been designed and optimised for the enablement of point to multipoint wireless applications. The IEEE designed the WiMAX MAC to subscribe to the following requirements [21]:

1. Point to multipoint. One of the requirements for WiMAX is that a Base Station (BS) is able to communicate concurrently to numerous fixed, fixed-mobile or mobile users. Similarly to Ethernet, the WiMAX MAC layer is based on Collision Sense Multiple Access with Collision Avoidance (CSMA / CA) to provide the point to multipoint (PMP) capability [8].
2. Connection orientated. The WiMAX MAC layer shall be able to support communication in all conditions as well as in various traffic profiles. As WiMAX is a packet switched wireless network, the system must be able to support both continuous and burstable traffic. Considering data traffic across a generic access network most data traffic is burstable in nature having short times of high data rates, then remaining dormant for a period of time. The IEEE aimed to support numerous users at high throughput rates [8].
3. Efficient spectrum use. The WiMAX MAC shall be capable of supporting methods that translate to an efficient use of allocated and or available spectrum [8].
4. Multiple Quality of Service (QoS) options. Support for an assortment of Quality of Service (QoS) classes and forms were needed for communication of various traffic conditions. Quality of Service (QoS) is a fundamental part of the WiMAX MAC layer and is defined by Service Flow or Scheduling mechanisms that are described later [8].
5. Multiple WiMAX / IEEE 802.16 physical layers. The WiMAX MAC layer shall be able to provide support for variations in the IEEE 802.16 standard physical layers [8].
6. Techniques for power management and security. The WiMAX MAC layer shall also incorporate features including power management techniques and security features [20].

With regard to Quality of Service (QoS), the MAC layer is responsible for delivery of the MAC Service Data Unit (SDU) to the peer MAC service access point in conformity to Quality of Service (QoS), fragmentation, concatenation and other transport functions associated with a particular connection's Service Flow characteristics [8].

A Service Flow is a unidirectional flow of packets that conforms to a particular Quality of Service (QoS). The Subscriber Station (SS) and Base Station (BS) provide this Quality of Service (QoS) according to the Quality of Service (QoS) parameters set defined for the Service Flow [8]. An illustration of the WiMAX classifier location as well as Service Flow is depicted in Figure 2-9 [12].

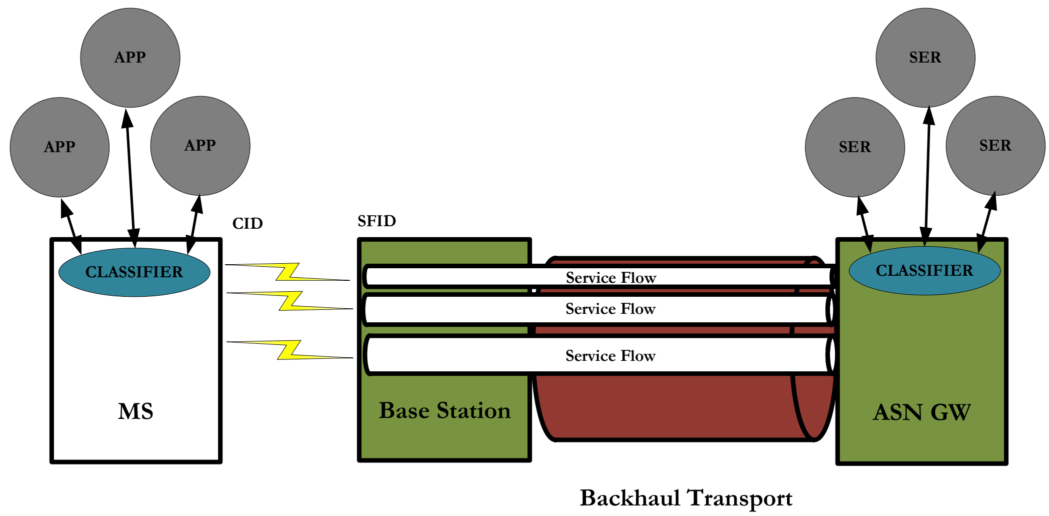

Figure 2-9. Classifier and Service Flow in the WiMAX Quality of Service (QoS) Framework [12]

2.4.2.1 WiMAX MAC Connection Identifier (CID)

In order for data to be transferred over a WiMAX link the User Equipment (UE) or Mobile Station (MS) and the Base Station (BS) need to firstly create a connection between the WiMAX MAC layers of the two communicating stations. An identifier known as a Connection Identifier (CID) is generated and assigned to each uplink / downlink connection. The Connection Identifier (CID) serves as an intermediate address for the data packets transmitted over the WiMAX link [8].

Another identifier also used within the WiMAX MAC layer is known as the Service Flow Identifier (SFID). This is assigned to unidirectional packet data traffic by the Base Station (BS) as depicted in Figure 2-9. It should also be deliberated that the Base Station (BS) WiMAX MAC layer handles the mapping of the Service Flow Identifiers (SFID) to

Connection Identifiers (CID) in order to facilitate the required Quality of Service (QoS) [8].

2.4.2.2 WiMAX Classification and Classification Rules

The process by which a MAC Service Data Unit (SDU) is mapped onto a particular transport connection for transmission between MAC peers is termed Classification. Classification associates a MAC Service Data Unit (SDU) with a transport connection which in turn creates an association with the Service Flow characteristics of that connection. This association with the Service Flow characteristics facilitates the delivery of MAC Service Data Unit (SDU) with the appropriate Quality of Service (QoS) constraints [8].

A classification rule is a set of identifying criteria applied to each packet entering a WiMAX network. It consists of protocol specific packet matching criteria, a classification rule priority and a reference to a Connection Identifier (CID). If a packet matches the specified packet matching criteria it is then passed on for delivery on the connection defined by the Connection Identifier (CID) [8]. Several classification rules may each refer to the same Service Flow. The classification rule priority is used for ordering the application of classification rules to packets. Explicit ordering is necessary because the patterns used by classification rules may overlap [8]. Downlink classification rules are applied by the Base Station (BS), and uplink classification rules are applied at the Subscriber Station (SS) [8]. It is possible for a packet to fail to match the set of defined classification rules. In this case the Convergence Sublayer shall discard the packet.

2.4.3 WiMAX Scheduling

The air interface scheduler at the MAC Sublayer determines how radio resources are assigned among multiple Service Flows based on Quality of Service (QoS) attributes. Resources assigned to a Mobile Station (MS) enable it to receive traffic over the downlink and transmit traffic over the uplink. [12]. WiMAX Scheduling services represent the data handling mechanisms that are supported by the MAC scheduler for data transport on a connection. Each connection is associated with a singular Scheduling service [8].

The outbound transmission scheduler selects data for transmission from a particular frame or bandwidth allocation alternatively known as the Scheduling queue. The Scheduling is performed by the Base Station (BS) for the downlink, and the Subscriber Station (SS) for the uplink.

The WiMAX scheduler also considers the following items for each active Service Flow [8]:

1. The Scheduling service specified for the Service Flow.
2. The values assigned to the Service Flows Quality of Service (QoS) parameters.
3. The availability of data for transmission.
4. The capacity of the allowed bandwidth.

By specifying a Scheduling type and its associated Quality of Service (QoS) parameters the Base Station (BS) scheduler can anticipate the throughput and latency needs of the uplink traffic and provide polls and / or grants at appropriate times [8]. WiMAX supports a variety of polling mechanisms that both vendors and carriers can select for use.

In each WiMAX sector, users follow a transmission protocol that control contention between users as well as enables services to be tailored to delay and bandwidth requirements of each user application. This is accomplished through four different types of uplink Scheduling mechanisms that are implemented using options such as unsolicited bandwidth grants, polling, and contention procedures [8].

Table 2-2 provides a summary of the Scheduling types and the poll or grant options available for each Scheduling type.

Table 2-2. Scheduling Services and Usage Rules

Scheduling Type	Piggy Back Request	Bandwidth Stealing	Polling
UGS	Not Allowed	Not Allowed	PM bit is used to request a unicast poll for bandwidth needs of non-UGS connections.
rtPS	Allowed	Allowed	Scheduling only allows unicast polling.
ertPS	Allowed	Allowed	Scheduling only allows unicast polling.
nrtPS	Allowed	Allowed	Scheduling may restrict a Service Flow to unicast polling via the transmission / request policy, otherwise all forms of polling are allowed.
BE	Allowed	Allowed	All forms of polling allowed.

Where:
UGS = Unsolicited Grant Service.
rtPS = real time Polling Service.
ertPS = enhanced real time Polling Service.
nrtPS = non real time Polling Service.
BE = Best Effort

2.4.4 WiMAX Service Flow Description

As initially indicated, a Service Flow is a unidirectional flow of packets that facilitate a particular Quality of Service (QoS). The Subscriber Station (SS) and Base Station (BS) provide Quality of Service (QoS) to an application according to the Quality of Service (QoS) parameter set out in the Service Flow [8]. Service Flows are provisioned during the installation phase when a Subscriber Station (SS) is installed and configured for attachment to the WiMAX network.

In order to map applications on a Subscriber Station (SS) to its required levels of Quality of Service (QoS), all data communications are presented in the context of a Transport

Connection [8]. Transport connections are associated with Service Flows (one connection per Service Flow) in order to provide a reference against which to request bandwidth. A transport connection defines the mapping between peer convergence processes that utilise the MAC as well as a Service Flow [8].

Service Flows are integral to the bandwidth allocation process for a Subscriber Station (SS) because Service Flows provide the mechanism for uplink and downlink Quality of Service (QoS) management. A Subscriber Station (SS) requests uplink bandwidth on a per connection basis thereby implicitly identifying the Service Flow. However bandwidth is granted by the Base Station (BS) to a Subscriber Station (SS) as an aggregate of grants in response to per connection requests from the Subscriber Station (SS) [8].

2.5 WiMAX (IEEE 802.16e/m) Quality of Service (QoS) Framework

The Quality of Service (QoS) framework for the WiMAX standard is based on Service Flows that exist between the Access Service Network Gateway (ASN-GW) and the Mobile Station (MS) or Subscriber Station (SS). As previously defined a Service Flow is marked with a Connection Identifier thereby inserting an indicator of Quality of Service (QoS) attributes such as packet loss, latency, jitter and throughput to be adhered to by the MAC layer [12].

The primary drive of the WiMAX Quality of Service (QoS) framework is to define transmission ordering and scheduling on the air interface, such that certain packets are classified and prioritised amidst a shared radio interface [8]. The IEEE has defined the following requirements for the enablement of Quality of Service (QoS) in the WiMAX framework design [8]:

1. The framework shall consist of a configuration and registration function for preconfiguring Subscriber Station (SS) based Service Flows and traffic parameters.
2. There shall be a function for the establishment of Service Flows and traffic parameters.

3. MAC Scheduling and Quality of Service (QoS) traffic parameters shall be utilised for the uplink Service Flows.
4. Quality of Service (QoS) traffic parameters shall exist for the downlink Service Flows.
5. Service Flow properties shall be grouped into named service classes such that upper layer entities and external applications (at the Subscriber Station (SS) and the Base Station (BS)) can request Service Flows in a globally consistent manner [8].

As packets enter the MAC interface of the Mobile Station (MS), Subscriber Station (SS) or Access Service Network Gateway (ASN-GW) they are associated with specific Service Flows determined by the predefined classifier rules [12] at the Convergence Sublayer. This is based on protocol specific packet matching criteria such as source IP address, destination IP addresses, source port, destination port, protocol and Differentiated Services Code Point (DSCP), all dependent on the direction of the traffic flow [12] .

The WiMAX (IEEE 802.16e) standard supports the following five Service Flow types. These are associated with the Scheduling schemes listed above [8] [12]:

1. Unsolicited Grant Service (UGS). This Service Flow is dedicated to real-time traffic with fixed-size data packets on a periodic basis.
2. real-time Polling Service (rtPS). This Service Flow is intended for real-time traffic with variable-size data packets on a periodic basis.
3. extended rtPS (ertPS). This Service Flow is intended for real-time traffic that generates variable-size data packets on a periodic basis with a sequence of active and silence intervals.
4. non-real-time polling service (nrtPS). This Service Flow is for delay tolerant traffic that requires a minimum reserved rate.
5. Best Effort (BE) service. General internet traffic is catered for by the Best Effort Service Flow.

Table 2-3. Description of Service Flow Quality of Service (QoS) characteristics for WiMAX [8] [12]

Service Flow Type	MRTR	MSTR	Maximum Latency	Maximum Jitter	Traffic Priority
UGS		X	X	X	
rtPS	X	X	X	X	X
ertPS	X	X	X		X
nrtPS	X	X			X
BE		X			X

Where: MTRT = Minimum Reserved Traffic Rate.
MSTR = Maximum Sustained Traffic Rate.

Each of the Service Flows above has the resulting Quality of Service (QoS) attributes associated; each determined by the applicable Convergence Sublayer and MAC layer configuration highlighted earlier. These attributes are highlighted in Table 2-3 as well as described further below.

2.5.1 Minimum Reserved Traffic Rate (MRTR) Parameter

The Minimum Reserved Traffic Rate (MRTR) parameter specifies the minimum bandwidth rate reserved for this Service Flow type and is classified in bits per second. The value of this parameter is calculated excluding the MAC overhead. The Base Station (BS) and Access Service Network Gateway (ASN-GW) are able to satisfy the bandwidth request for a connection up to its minimum reserved traffic rate based on the available Scheduling and allocation mechanisms. If less bandwidth than its minimum reserved traffic rate is requested then the Base Station (BS) may reallocate the excess reserved bandwidth for other purposes [8].

2.5.2 Maximum Sustained Traffic Rate (MSTR) Parameter

The Maximum Sustained Traffic Rate (MSTR) parameter defines the peak information rate of a requested Service Flow. The rate is expressed in bits per second and pertains to the Service Data Unit (SDU) at the entrance to the system. This parameter does not include transport, protocol, or network overhead such as MAC headers or Cyclic Redundancy Checks or non-payload session maintenance overhead like Session Initiation Protocol (SIP), Media Gateway Control Protocol (MGCP) and H.323 administration, etc. [8].

This parameter does not limit the instantaneous rate of the service, however the service is regulated to conform to this parameter at the destination network interface in the uplink direction [8]. This is done on an average basis over time and not instantaneously. On the downlink direction it is assumed that the service has already been regulated at the ingress of the network. If this parameter is set to zero then there is no explicitly mandated maximum rate [8].

2.5.3 Maximum Latency Parameter

This Maximum Latency parameter specifies the maximum wait between a packet entry of the Base Station (BS) or the Subscriber Station (SS) and the forwarding of the Service Data Unit (SDU) to the air interface [8]. This parameter signifies a service commitment at the Base Station (BS) or Subscriber Station (SS) guaranteed by the Base Station (BS) or Subscriber Station (SS). If a Service Flow exceeds its minimum reserve rate then the Base Station (BS) or Subscriber Station (SS) does not need to meet this service commitment for that specific Service Flow [8].

2.5.4 Tolerated Jitter Parameter

Similar to jitter in any packet switched wireless network, the Tolerated Jitter Parameter defines the maximum variation in delay between packets for a connection [8].

2.5.5 Traffic Priority Parameter

The Traffic Priority Parameter specifies priority assigned to a Service Flow. If two Service Flows are identical in all Quality of Service (QoS) parameters besides priority then the Service Flow with the higher priority is given lower delay and a higher buffering inclination [8]. The priority parameter does not take precedence over any conflicting Service Flow Quality of Service (QoS) parameter in terms of non-identical Service Flows [8].

2.6 Quality of Service (QoS) Differentiation between IEEE 802.16e and IEEE 802.16m

In a similar fashion to the IEEE 802.16e standard the Quality of Service (QoS) mechanisms in the IEEE 802.16m standard is achieved by the interpretation of flows of packets as Service Flows. The packet flows occur in a singular direction and are mapped to a singular transport connection [22].

The differences between IEEE 802.16e and IEEE 802.16m are as follows where the following have been added to the IEEE 802.16m standard [12] [22]:

1. Adaptive Granting and Polling service: was added to the framework to facilitate various packet sizes with variable arrival intervals [12].
2. Quick Access: is a shortened random access delay was introduced in the form of quick access which significantly improves the quality of experience of delay-sensitive and interactive applications. A quick access message is carried in the first step of the random access bandwidth request procedure [12].
3. Delayed Bandwidth Request: was introduced to request bandwidth proactively in order to avoid the random access bandwidth request procedure and thus reduce the access delay [12].
4. Priority controlled access: is where a priority controlled access class is introduced to prioritise contention based random access [12].

2.7 Conclusion

The preceding Chapter has emphasised that the understanding of the architecture and design of packet switched wireless networks is essential to understanding the Quality of the Service (QoS) framework that will be most applicable for voice traffic. In the chapter above both the LTE and WiMAX standards were investigated with prominence on the Radio Link Control (RLC), Media Access Control (MAC), Scheduling schemes and Admission Control (AC) defined by the standards bodies. The standards have indicated that varying Scheduling structures and Radio Link Control (RLC) configuration determine the latency, jitter and packet loss of a Service Flow or Bearer.

Quality voice can be facilitated using the Quality of Service (QoS) Framework based on a QoS Class Identifier (QCI) applied on a Bearer or a priority Scheduling and polling scheme applied on a Service Flow. Although Quality of Service (QoS) can be applied on the Protocol Data Units (PDU) and Service Data Unit (SDU) via the Radio Link Control (RLC) and Media Access Control (MAC) layers, there is no specification that has been explicitly quantified for voice. It is inherently up to the vendor and operator to decide if UGS, rtPS, ertPS, QCI 1, QCI 2 or QCI 3 can be used for voice. There are no voice traffic specifications in terms of Admission Control (AC), the mapping of voice traffic to the backhaul network or the initiation of an applicable Bearer or Service Flow.

In the latter chapters these questions shall be answered, however in the following chapter related work in terms of voice over packet switched wireless networks is investigated with emphasis on industry proposals and research solutions.

3 Related Work and Current Initiatives

3.1 Introduction

Legacy networks have been designed and optimised for voice communications. With the technological and standardisation move towards packet switched wireless networks comes the challenge of optimal resource allocation for voice. Based on this conundrum the research question to operators becomes how best to efficiently deliver legacy switched voice over packet switched wireless networks.

The 3rd Generation Partnership Project (3GPP) envisaged that voice would be carried using the IP Multimedia Subsystem (IMS) architecture, however this has been slow to deployment by majority of the mobile operators due to its implementation complexity. Further to this, the majority of wireless operators possess expansive legacy networks to which they have made significant investments. These investments need to be fully leveraged until the end of their lifespan.

For LTE operators wanting to utilise their existing 2nd Generation or 3rd Generation networks for voice, the standards bodies have embraced Circuit Switched Fallback (CSFB). Circuit Switched Fallback (CSFB) is a dual radio solution that caters for the LTE User equipment (UE) to fall back onto existing 2nd Generation or 3rd Generation networks for voice and does not traverse the 4th Generation Radio Access Network (RAN). Other prominent solutions that have been proposed are, Voice over LTE via Generic Access (VoLGA) and Over the Top (OTT) voice [23].

In respect of the ideal scenario for voice over packet switched wireless networks, every area from the air interface of the dialled party to the medium of reception of the received party needs to be sufficiently managed to ensure an acceptable Quality of Service (QoS). IP Multimedia Subsystem (IMS) has been punted as the technological solution for voice over Long Term Evolution (LTE) by the standards bodies. Although listed below are interim options that are in direct violation of this recommendation, the objective is not to undermine this technology, but rather provide carriers with further options such that carriers are not held ransom to a singular technological option in terms of price or deployment requirements [11].

Other research initiatives include integration between legacy systems and packet switched wireless systems, as well as initiatives based on resource allocation models for the different layers of the Open Systems Interconnection (OSI) stack which broker or negotiate resource allocation for voice.

In the following chapter a description of Circuit Switch Fallback (CSFB), Voice over LTE Generic Access (VoLGA), Over the Top (OTT) voice and IP Multimedia Subsystem (IMS) is provided as well as the advantages and disadvantages of each option. Quality of Service (QoS) aware Admission Control (AC) and Scheduling is also described.

3.2 Circuit Switch Fallback (CSFB)

Circuit Switch Fallback (CSFB) utilises the existing 2nd Generation or 3rd Generation networks to deliver voice over an LTE network, this option does not exist for WiMAX. Circuit Switch Fallback (CSFB) permits the User Equipment (UE) to vacate the LTE network and go over to the 2nd Generation or 3rd Generation network to make and receive voice calls [24]. The LTE network utilises the paging channel to redirect a voice call from the LTE network to the existing 2nd Generation or 3rd Generation network. Circuit Switch Fallback (CSFB) requires a new Mobile Switching Centre (MSC) and Serving GPRS Support Node (SGSN) interface called SGs in order to signal incoming calls and Short Message Service (SMS) messages to the Mobility Management Entity (MME) as illustrated in Figure 3-1 [23].

A Circuit Switch Fallback (CSFB) enabled User Equipment (UE) is registered by the Mobility Management Entity (MME) in an LTE network on both the LTE core and 2nd Generation or 3rd Generation networks alike. When a voice call is initiated, the User Equipment (UE) sends a request to the Mobility Management Entity (MME) instructing the Mobility Management Entity (MME) to perform Circuit Switch Fallback (CSFB). The Mobility Management Entity (MME) then requests the eNodeB to redirect the User Equipment (UE) to the 2nd Generation or 3rd Generation network [23]. The limitations of such a solution include user terminals requiring the ability to access both the LTE network and the 2nd Generation or 3rd Generation networks, hence these terminals must support the combined Evolved Packet System (EPS) / International Mobile Subscriber Identity (IMSI), attach, detach, location update procedures and Circuit Switch Fallback (CSFB) [23] [24].

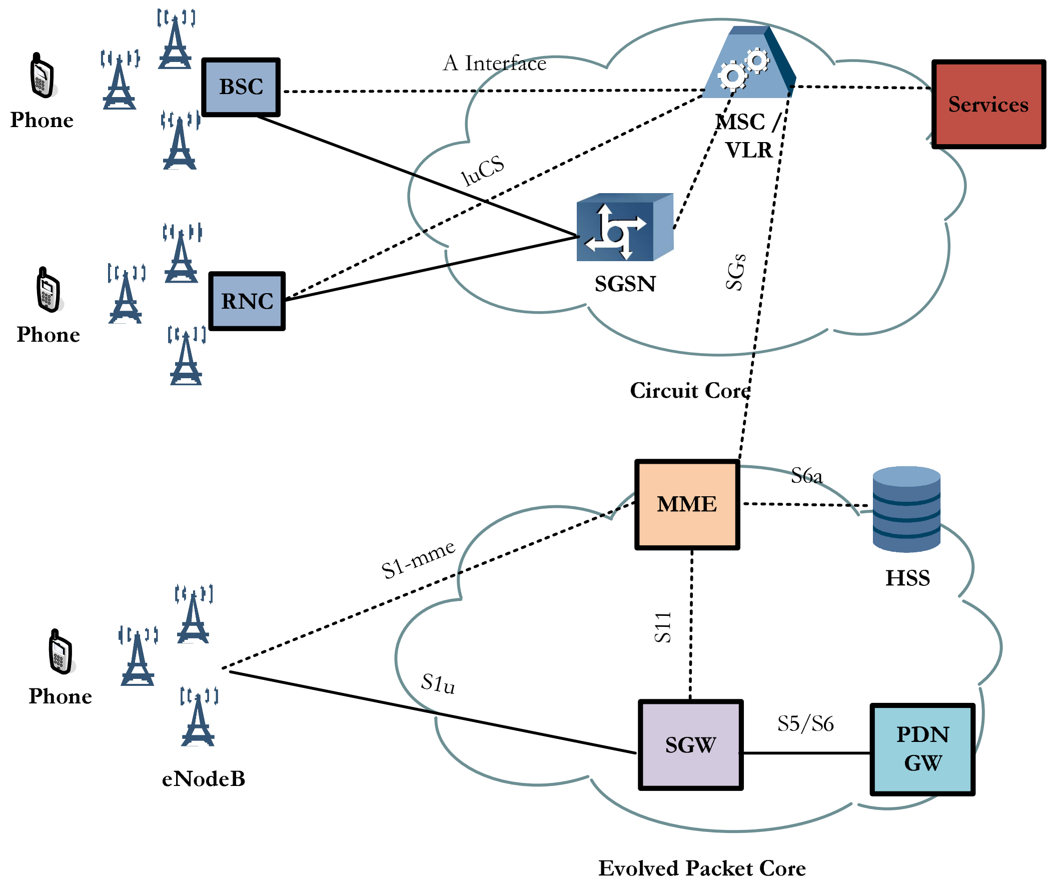

Figure 3-1. Circuit Switch Fallback (CSFB) Architecture [11]

Advantages of Circuit Switch Fallback (CSFB) include [23]:

1. No new hardware network components are required because Circuit Switch Fallback (CSFB) reuses the Mobile Switching Centre's (MSC), circuit switch platforms, Operational Support Systems (OSS), Business Support Systems (BSS), prepaid and post-paid billing systems.
2. Does not force SMS to fall back.

Disadvantages of Circuit Switch Fallback (CSFB) include [23]:

1. Circuit Switch Fallback (CSFB) requires enhancements to the Mobility Management Entity (MME), Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) and Mobile Switching Centre (MSC). The Mobile Switching Centre (MSC) needs to incorporate changes such as the addition of the SGs interface to the Mobility Management Entity (MME), simultaneous paging on the A / Iu / SGs interfaces and sending and receiving SMS over the SGs interface. These may be costly for operators that are looking to reduce investment in legacy infrastructure.
2. Does not support simultaneous voice and data services for 2nd Generation or 3rd Generation networks without 2nd Generation or 3rd Generation networks Dual Transfer Mode protocol.
3. Circuit Switch Fallback (CSFB) is signalling intensive, hence induces an extended post dial delay due to the mobile handset being forced to fall back and register with the 2nd Generation or 3rd Generation network to make or receive a call.
4. Requires additional network engineering considerations in terms of coverage. The LTE coverage areas must be engineered to overlap with the coverage of the 2nd Generation or 3rd Generation network in order for LTE subscribers to handover to the 2nd Generation or 3rd Generation network for voice services.

3.3 Voice over LTE Generic Access (VoLGA)

Voice over LTE Generic Access (VoLGA) makes use of a VoLGA Access Network Controller (VANC) as illustrated in Figure 3-2 to tunnel circuit switched traffic across an LTE network. Contrary to Circuit Switch Fallback (CSFB), Voice over LTE Generic Access (VoLGA) packetizes and delivers voice calls natively over LTE. The voice call set-up signalling as well as the circuit switch voice stream are transported transparently over the LTE data Bearer to the VoLGA Access Network Controller (VANC) [23].

Voice over LTE Generic Access (VoLGA) is based on the existing 3GPP Generic Access Standard. The Generic Access Standard was introduced in 3GPP Release 6 and was extended in 3GPP Release 8 (2008) to contain provision for the 3rd Generation (3G) core Iu network interface [23].

If a voice call originates from the LTE Radio Access Network (RAN), the User Equipment (UE) invokes a voice service by sending a request to the VoLGA Access Network Controller. The VoLGA Access Network Controller (VANC) forwards this request to the Mobile Switching Centre (MSC) using either the A or Iu interface. After the User Equipment (UE) authentication / authorisation with the Mobile Switching Centre (MSC) the User Equipment (UE) forwards a set-up message to the Mobile Switching Centre (MSC) via the VoLGA Access Network Controller (VANC) thereby ordering the Mobile Switching Centre (MSC) to originate a call. On receipt of the message the Mobile Switching Centre (MSC) instructs the VoLGA Access Network Controller (VANC) to establish a Bearer connection. The VoLGA Access Network Controller (VANC) then establishes an uplink and downlink connection from the Mobile Switching Centre (MSC) to the User Equipment (UE) [23].

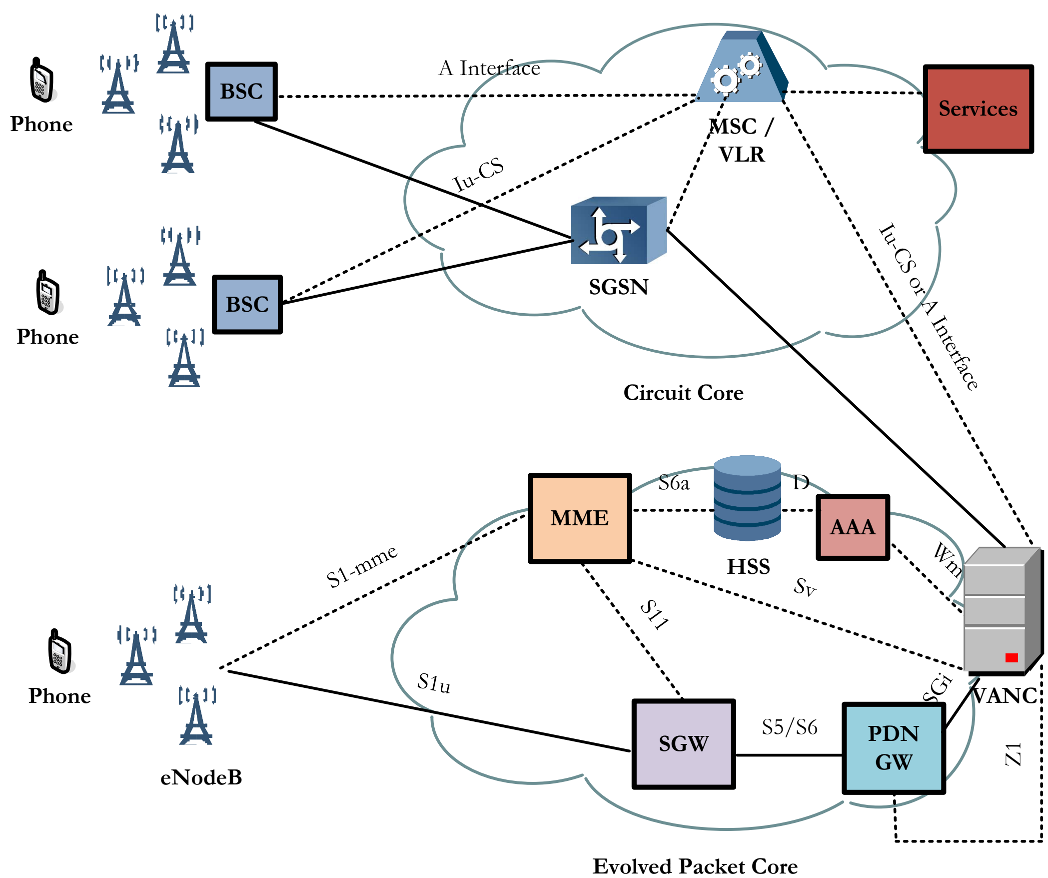

Figure 3-2. Voice over LTE Generic Access (VoLGA) Architecture [11]

If a voice call requires termination on an LTE network via the VoLGA Access Network Controller (VANC) then the User Equipment (UE) on the LTE network is paged via the VoLGA Access Network Controller (VANC) in order to begin the dedicated connection establishment. The Mobile Switching Centre (MSC) authenticates the User Equipment (UE) and then initiates the call set-up. As with call origination, the VoLGA Access Network Controller (VANC) sets up the uplink and downlink connection path. Once the voice Bearer has been established the User Equipment (UE) summons the subscriber and sends back a message to the Mobile Switching Centre (MSC) via the VoLGA Access Network Controller (VANC). The Mobile Switching Centre (MSC) sends an alert message notification to the calling party, on answering the alert, before establishing the connection the User Equipment (UE) sends a connect message via the VoLGA Access Network Controller (VANC) to the Mobile Switching Centre (MSC) [23].

Advantages of Voice over LTE Generic Access (VoLGA) include [23]:

1. No major network architectural changes required to the 2nd Generation or 3rd Generation Mobile Switching Centre (MSC) or modifications to the LTE radio or core network. All enhancements are implemented on the User Equipment (UE) and the VoLGA Access Network Controller (VANC).
2. There is no additional signalling load on the on the Radio Access Network side in order to initiate a voice call.
3. Voice over LTE Generic Access (VoLGA) supports combinational services such as, IP Multimedia Subsystem (IMS) Rich Communication Services on LTE and has no impact on any value-added network-supported services and emergency services because all telephony services are supported by the VoLGA Access Network Controller (VANC).
4. Supports SMS.
5. Avoids the need to make changes to back-office support systems, such as billing, subscriber management and customer care.
6. Supports LTE to 2nd Generation or 3rd Generation handovers as it is circuit switched based.
7. Enables simultaneous LTE data and circuit switched voice.
8. Enables faster call set-up than Circuit Switched Fallback because the User Equipment (UE) remains within LTE domain.

Disadvantages of Voice over LTE Generic Access (VoLGA) include [23]:

1. Voice over LTE Generic Access (VoLGA) requires the set-up and deployment of VoLGA Access Network Controller (VANC); this is a new element in the network architecture.
2. Operators need to upgrade both the Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) and Mobility Management Entity (MME) to handle Voice over LTE Generic Access (VoLGA).
3. Voice over LTE Generic Access (VoLGA) is not standardised by the 3rd Generation Partnership Project (3GPP) body. There is no guarantee that Voice over LTE Generic Access (VoLGA) will be adapted by the 3rd Generation Partnership Project (3GPP) body and without the support of 3rd Generation Partnership Project (3GPP) body, Voice over LTE Generic Access (VoLGA) enabled User Equipment (UE) and network upgrades may be slow to market.
4. Requires terminal modifications that will facilitate tunnel enablement. Generic Access chips sets are known to be of significant impact to User Equipment (UE) battery life.
5. Voice over LTE Generic Access (VoLGA) is only suitable for operators coming from the 3rd Generation Partnership Project (3GPP) body and not from the CDMA standards.

3.4 Over-the-Top (OTT) Voice over IP (VoIP)

The traditional definition of Over the Top (OTT) voice is the distribution of voice without going through a mobile operator or telecoms operator, that is, voice carried via an application over the internet or a private network via IP. Voice over IP (VoIP) has an inherent advantage, that being its ability to easily integrate with various systems. This advantage has been leveraged by the likes of Skype and Google Talk to provide free or cheap voice services to customers with a basic internet connection.

Numerous Over the Top (OTT) providers provide voice as a free service from their custom applications and draw revenue from avenues such as advertising, which goes against the traditional telecommunications operator revenue model. This in turn has spurred the mobile and fixed providers to broaden their service base into Over the Top (OTT) voice applications and services. Operators have thereby taken the strategic direction to explore services such as voice-enhanced instant messaging and voice plugins catering for internet communities or social networks over and above their existing base of active Voice over IP (VoIP) users [23].

In the mobile space one of the key catalysts for Over the Top (OTT) voice is the increased penetration of smartphones and the emergent availability of attractive flat-rate mobile data plans. Mobile consumers are showing greater interest in Over the Top (OTT) voice as such applications become easier to use, save consumers money and can be integrated with their existing social media applications.

The delivery approach of Over the Top (OTT) voice may consist of the following options:

1. User defined application that enables a Voice over IP (VoIP) session between other users of the same application.
2. Session based call initiation using protocols such as Session Initiation Protocol (SIP) or H323 / H248 to initiate call flow between users.

Advantages of Over the Top (OTT) voice include [23]:

1. Easy to install. A user typically downloads an application such as Skype, Google Talk or X-Lite, completes the registration process and is ready to commence with a voice call to other users with the same application.
2. Is relatively inexpensive or free to use.

Disadvantages of Over the Top (OTT) voice include [23]:

1. Voice performance can be severely compromised in a congested mobile or fixed access network.
2. Over the Top (OTT) voice has debatable continuity performance after handover to 2nd Generation or 3rd Generation networks since it cannot rely on Single Radio Voice Call Continuity (SRVCC). Single Radio Voice Call Continuity (SRVCC) is

an LTE functionality that allows an IP Multimedia Subsystem (IMS) VoIP call in the LTE packet domain to be moved to a legacy voice domain, i.e. 2nd Generation or 3rd Generation or 1x CDMA networks.

3. Performance is reliant on the Quality of Service (QoS) offered by the transmission network provider. This may come at a price as carriers see this area as a way of recouping lost voice revenue.
4. Requires support from handset vendors and operating system developers in terms of optimal application performance on the handset.

3.5 Quality of Service (QoS) Aware Combined Admission Control (AC) and Scheduling for Voice

If LTE is considered as an example in order to provide quality of service control considering physical resource block allocation, the Admission Control (AC) and packet Scheduling need to be Quality of Service (QoS) aware [25] [26] [27]. The Quality of Service (QoS) aware Admission Control (AC) will grant or deny access to a new radio resource Bearer depending whether it will be able to fulfil the new Bearer or Service Flow Quality of Service (QoS) requirement while guaranteeing Quality of Service (QoS) on existing Bearers or Service Flows. The Quality of Service (QoS) aware packet Scheduling allocates the dynamically shared data channel to active radio Bearers based on Quality of Service (QoS) requirements [28].

With reference to Admission Control (AC), the Admission Control (AC) algorithm decides to admit a new user, Bearer or Service Flow if the sum of the Guaranteed Bit Rate (GBR) Bearers of new and existing users is less than a predetermined value [28]. With reference to packet Scheduling, the packet Scheduling algorithm consists of priority Scheduling given to packets which are farthest below its stated or contracted Guaranteed Bit Rate (GBR) metric requirement. This is done with consideration to the estimated achievable throughput on available physical resource blocks; hence the Admission Control (AC) and packet Scheduling give rise to a proportional fair and Guaranteed Bit Rate (GBR) aware metric that is used for Quality of Service (QoS) [28].

3.6 Voice Quality of Service Negotiation in IP Multimedia Subsystem (IMS)

IP Multimedia Subsystem (IMS) has been defined as a multimedia session-control subsystem encompassing core network elements for the provision of multimedia services based on a horizontally-layered architecture [29] [30]. IP Multimedia Subsystem (IMS) can be used to support Quality of Service (QoS) for IP multimedia sessions offering users enhanced service quality for varying service applications [30] [31]. In terms of voice over packet switched wireless networks, IP Multimedia Subsystem (IMS) is envisaged by the 3rd Generation Partnership Project (3GPP) as the successor to the 2nd Generation or 3rd Generation circuit-switched voice architecture.

The IP Multimedia Subsystem (IMS) procedures for negotiating multimedia session characteristics are specified by the 3rd Generation Partnership Project and are based on the Internet Engineering Task Force (IETF) Session Initiation Protocol (SIP) [30] [32]. For the deployment of IP Multimedia Subsystem (IMS) voice the IP Multimedia Subsystem (IMS) requires that a client be deployed on the User Equipment (UE) in the packet switched wireless network [23].

Irrespective of the network type available, all IP Multimedia Subsystem (IMS) enabled User Equipment (UE) are obliged to register on the IP Multimedia Subsystem (IMS) platform in order to receive data and voice services. When the IP Multimedia Subsystem (IMS) platform originates or terminates a voice session utilising LTE or 3rd Generation packet switch access, the session is configured according to the IP Multimedia Subsystem (IMS) standards based origination or termination procedures [23] [33].

The IP Multimedia Subsystem (IMS) defines a Quality of Service (QoS) framework that incorporates a policy based management resource control function to provide interaction between applications and resources [34] [35]. Extensions to IP Multimedia Subsystem (IMS) have included inter-domain resource allocation via topology discovery mechanisms that allow the Policy Decision Function to map out neighbouring Quality of Service (QoS) control elements [36]. Client applications are usually designed in a manner such that they originate with predefined configuration parameters. Possible parameters to be negotiated include: type; quality; encoding of media; terminal capabilities to be used; and desired Quality of Service (QoS) per media stream (e.g., guaranteed Quality of Service (QoS) or Best Effort) [30].

3.6.1 IP Multimedia Subsystem (IMS) Architecture for Voice

The IP Multimedia Subsystem (IMS) architecture is expansive in its design but not all entities and interfaces are required for all use cases. As the current research is focused on voice over packet switched wireless networks [37], only those entities in the IP Multimedia Subsystem (IMS) architecture that are immediately required for the implementation of voice are presented [38]. The simplified IP Multimedia Subsystem (IMS) architecture is described below and illustrated in Figure 3-3 [29] [38].

1. The Call Session Control Function (CSCF) operates as the core components of IP Multimedia Subsystem (IMS) architecture and consists of the following entities [29] [38]:
    a. Proxy - Call Session Control Function (P-CSCF). The Proxy - Call Session Control Function (P-CSCF) is the initial point of contact for a User Equipment (UE) and behaves as a proxy entity, thereby accepting Session Initiation Protocol requests and onward forwarding the requests. The User Equipment (UE) attaches to the Proxy - Call Session Control Function (P-CSCF) prior to performing IP Multimedia Subsystem (IMS) registrations and initiating Session Initiation Protocol sessions. The Proxy - Call Session Control Function (P-CSCF) may be in the home domain of the IP Multimedia Subsystem (IMS) operator, or it may be in the visiting domain where the User Equipment (UE) may be roaming. For attachment to a given Proxy - Call Session Control Function (P-CSCF), the User Equipment (UE) performs the Proxy - Call Session Control Function (P-CSCF) discovery procedures. Attachment to the Proxy - Call Session Control Function (P-CSCF) is required in order for the User Equipment (UE) to initiate IP Multimedia Subsystem (IMS) registrations and session enablement [38].
    b. Interrogating - Call Session Control Function (I-CSCF). The Interrogating - Call Session Control Function (I-CSCF) is the entry contact within an operator's network for all connections destined to a

user. While the Proxy - Call Session Control Function (P-CSCF) is the first point of contact for users regardless of the network, the Interrogating - Call Session Control Function (I-CSCF) is the point of contact for users in their home network, that is the Interrogating - Call Session Control Function (I-CSCF) facilitates all connections for that user [38].

c. Serving - Call Session Control Function (S-CSCF). The Serving - Call Session Control Function (S-CSCF) is the heart of the IP Multimedia Subsystem (IMS) core network and is responsible for handling the registration process, making routing decisions, maintaining sessions and downloading user information and service profiles from the Home Subscriber Server (HSS). The Serving - Call Session Control Function (S-CSCF) is the most processing resource demanding node of the IP Multimedia Subsystem (IMS) core network due to its initial filter criteria processing logic which enables IP Multimedia Subsystem (IMS) service control [38].

2. The Home Subscriber Server (HSS) is the master database for a user. It is comparable to the Home Location Register (HLR) in a legacy mobile radio network. The Home Subscriber Server contains subscription related information required by the network entities handling sessions. The Home Subscriber Server support call control servers by solving authentication, authorisation, addressing resolution and location dependencies, etc. The Home Subscriber Server (HSS) is common to both the LTE core and IP Multimedia Subsystem (IMS) architectures [10] [29] [38].
3. The Application Server (AS) undertakes the control of the end services required by the User Equipment (UE). Applications implemented on the application server can make use of a set of network capabilities such as call control, location, Short Message Service (SMS) and Multimedia Message Service (MMS). Each set of capabilities is supported by a dedicated network server (e.g. switch, location server) and through a specific network to network interface.

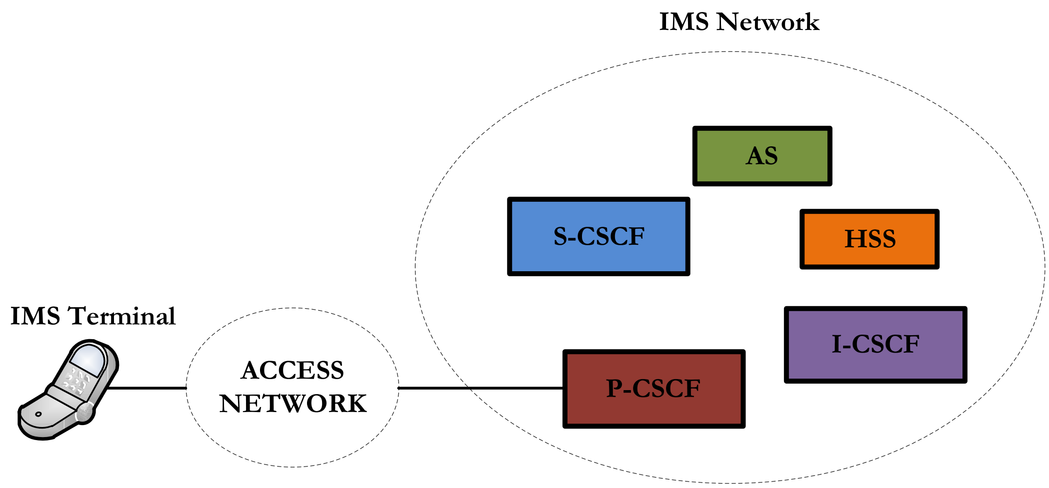

Figure 3-3. Simplified IP Multimedia Subsystem (IMS) Network [38]

The generalised procedure for IP Multimedia Subsystem (IMS) voice session establishment is illustrated in Figure 3-4 and described as follows [30]:

1. The User Equipment (UE) establishes a connection to the IP Multimedia Subsystem (IMS) core through an Access Network.
2. The Proxy - Call Session Control Function (P-CSCF) is allocated to serve as an inbound or outbound SIP proxy.
3. SIP application-level registration to the IP Multimedia Subsystem (IMS) network is established via the Proxy - Call Session Control Function (P-CSCF) which may be located at either the home network or at the visiting network.
4. The Proxy - Call Session Control Function (P-CSCF) interfaces with a Policy Decision Function (PDF), that is, the Policy and Charging Rules Function (PCRF) and PDN - Gateway (P-GW) that authorises and initiates the use of Bearer and Quality of Service (QoS) resources for IP Multimedia Subsystem (IMS) services in the Access Network.
5. The IP Multimedia Subsystem (IMS) session establishment, modification, and release is facilitated by the Serving - Call Session Control Function (S-CSCF) acting as a SIP termination server.

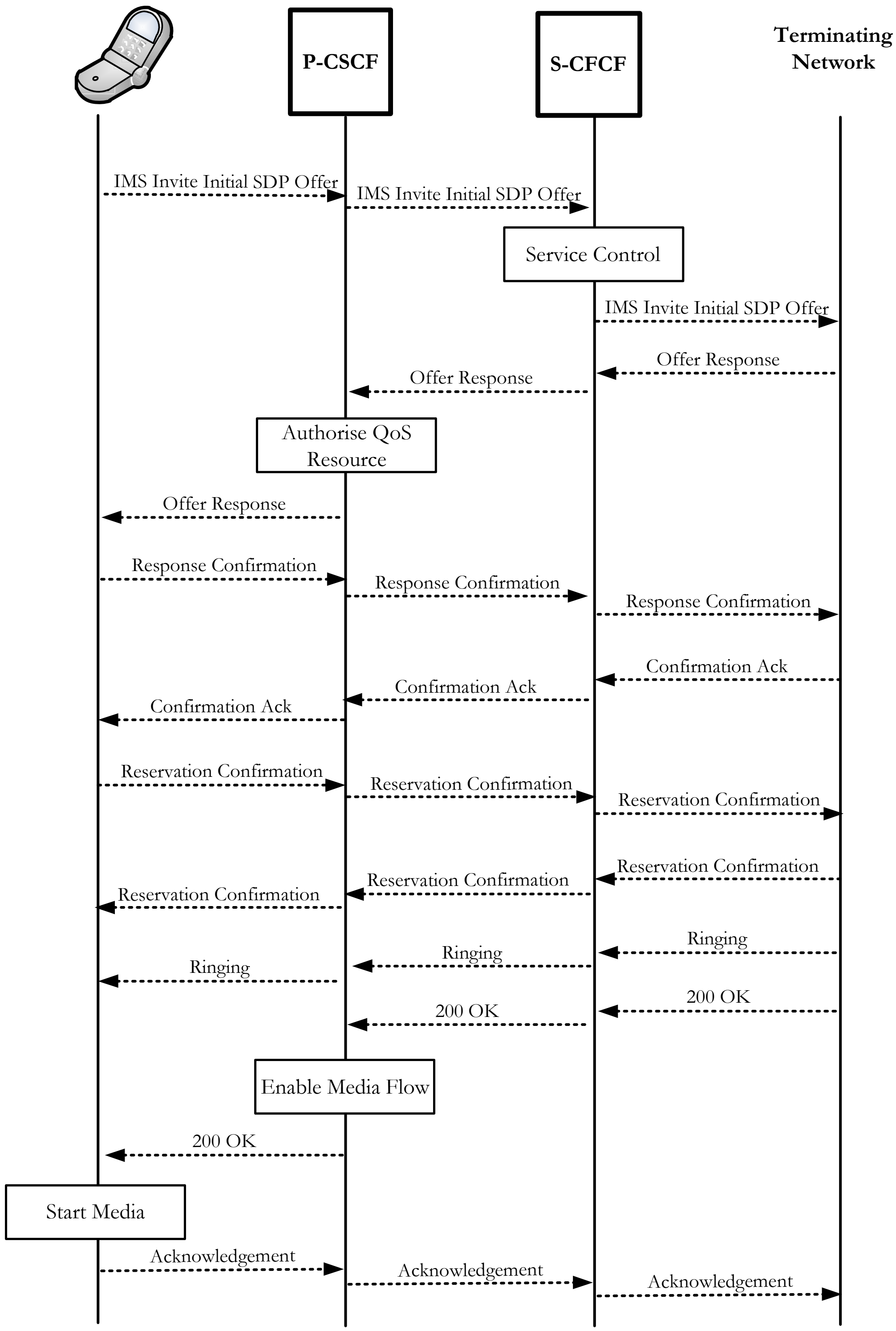

Figure 3-4. IP Multimedia Subsystem (IMS) Session Establishment for Users in the Home Network

6. All SIP signalling to and from the User Equipment (UE) traverses the allocated Serving - Call Session Control Function (S-CSCF). The Serving - Call Session Control Function (S-CSCF) interrogates the Home Subscriber Server (HSS) to access user profile information and fetch subscription data for authentication, authorization and accounting purposes. The Serving - Call Session Control Function (S-CSCF) facilitates various service provision components by invoking one or more application servers as required.
7. Session negotiation procedure commences and is based on the Session Description Protocol (SDP) offer / answer model, which provides a mechanism for User Equipment (UE) to use Session Description Protocol (SDP) to arrive at a common view of a multimedia session between them. The model involves one entity offering the other a description of desired session parameters and the other entity answering with the desired session parameters from its perspective.
8. After negotiation the IP multimedia session commences.

Advantages of IP Multimedia Subsystem (IMS) voice include [23]:

1. The enablement of advanced voice and data-blended services.
2. Support for concurrent voice and data handovers from LTE to 3rd Generation circuit-switched and packet-switched networks.
3. Offers the convergence of fixed and mobile services on wireline and wireless networks.

Disadvantages of IP Multimedia Subsystem (IMS) voice include [23]:

1. A sizable investment is required due to the need to deploy the IP Multimedia Subsystem (IMS) core, including Call Session Control Function (CSCF), Telephony Application Server (TAS) as well as the IP Short Message Gateway (IP-SM-GW).
2. Further investment is required in Session Continuity Control (SCC) and Single Radio Voice Call Continuity (SRVCC) application servers in order to provide hand-down to 3rd Generation circuit-switched voice networks.
3. The upgrades of all Mobile Switching Centre (MSC) servers are required in order to support Single Radio Voice Call Continuity (SRVCC) signalling.

4. Information Technology (IT) systems and processes changes are required in terms of provisioning, billing and support.
5. IP Multimedia Subsystem (IMS) has no ecosystem for roaming or interconnection.
6. Intense usage of battery power as the network constantly keeps a communication channel open to the User Equipment (UE) also Voice over IP (VoIP) calls cause a much higher processor load during a call.

3.7 Conclusion

The chapter above listed the possibilities that operators may have to successfully carry quality voice over a next generation packet switched wireless network. A description of Circuit Switch Fallback (CSFB), Voice over LTE Generic Access (VoLGA), Over the Top (OTT) voice and IP Multimedia Subsystem (IMS) was highlighted as well as the advantages and disadvantages of these options.

The interim options listed are valuable for operators with a need to deploy a next generation wireless network and yet maximise on their investment into existing 2nd Generation or 3rd Generation networks in terms of voice deployment. Circuit Switch Fallback (CSFB) and Voice over LTE Generic Access (VoLGA) are the key enablers for these operators but these options only diverge from the inevitable move to a converged IP solution.

The interim options are also fundamental to currently enable voice over an LTE enabled device without the complexity of deploying the IP Multimedia Subsystem (IMS). As has been listed the IP Multimedia Subsystem (IMS) is the standards bodies preferred option for network evolution however it is a fundamental shift from traditional circuit switched platforms or to some extent Softswitch platforms. For an operator to implement the IP Multimedia Subsystem (IMS) their entire switching core needs to be replaced, operators find it more acceptable for their Access Network (AN) to be replaced than for their core network to be supplanted. The reasons are that operators do not want to disturb existing wholesale and enterprise customers unless they categorically need to. IP Multimedia Subsystem (IMS) however has the advantage of enabling multimedia services to all customers. Operators need to find a point of evolution when multimedia services become important to their wholesale and enterprise customers.

Many operators view the increased infiltration of Over the Top (OTT) voice as a threat to existing revenues, especially voice applications on smartphones. Operators need to rather view the current social media revolution as a larger threat and participate in this space with an integrated Over the Top (OTT) voice solution merged with existing social media applications. The biggest disadvantage of Over the Top (OTT) voice is its lack of Quality of Service (QoS). However, control of Quality of Service (QoS) is a major benefit that operators possess in their tool-kit that they can exploit. Other than the IP Multimedia Subsystem (IMS) none of the other interim voice options listed above sufficiently cater for Quality of Service (QoS) for Over the Top (OTT) voice. The new information that this research has provided is the ability to provide Quality of Service (QoS) for Over the Top (OTT) voice by sufficiently recognising and admitting Over the Top (OTT) voice through a packet switched wireless network.

In light of operators' flexibility in evolving their Radio Access Network (RAN) as opposed to an evolution of their core voice network, the next chapter seeks to provide a new methodology of prompting Quality of service over the radio access and core portions of the LTE network without significantly influencing the core voice network. This is all presented in line with enabling a Quality of Service (QoS) framework for voice considering; the Admission Control (AC), the Bearer or Service Flow selection and the resource mapping of the of the Bearer or Service Flow to the transport network.

4 Proposed Method used in Deriving an Overall QoS Resource Management Framework for Voice

4.1 Introduction

In any packet switched network, Quality of Service (QoS) offered for traffic is determined by the ingress and egress conditioning of packets at each and every node in the network. Considering Differentiated Services (Diffserv) defined by the Internet Engineering Task Force (IETF) RFC 2475 [39] and Multiprotocol Label Switching (MPLS) [40] defined by the Internet Engineering Task Force (IETF) RFC 3031 [40], each protocol expresses a packet forwarding treatment or decision applied at the ingress and / or egress points of routers through which Service Flows or Label Switched Paths traverse. When a voice call is made, the applied conditioning to the voice packets at the ingress and egress of each node of the packet switched network determine the Quality of Service (QoS) that is applied to that specific voice call.

Based on this philosophy the following nodes are considered:

1. The Admission Control (AC) point.
2. The Bearer or Service Flow which consist of the Radio Link Control (RLC) configuration, MAC and packet Scheduling of the physical resource blocks over the air interface.
3. The transport or backhaul network.

Considering the ingress and egress philosophy listed above, Admission Control (AC) becomes important for the recognition of voice traffic or the prioritisation of an incoming voice call for traffic conditioning purposes at the next node. The action of the next node is mapping of the admitted voice stream into a real-time or equivalent Bearer or Service Flow as well as the summoning of such a Bearer or Service Flow. The final node is the mapping of the voice Bearer or Service Flow to the appropriate Transport resources.

In the flowing chapter a generic voice Quality of Service (QoS) framework based on static inter-domain mapping, consisting of the Admission Control (AC) component, Admission Control (AC) to Bearer or Service Flow mapping and Bearer or Service Flow to transportation mapping is presented.

4.2 Generic Voice Quality of Service (QoS) Framework

In a generic Quality of Service (QoS) framework, a network is divided into different administrative domains with each network domain consisting of handshake entities (service negotiating entities) such as a negotiating manager and a resource manager. The negotiating manager and resource manager are specific to each domain; where the former negotiates with its subscribers and the latter checks the availability of network resources [41].

An innovative resource management framework shall comprise of the following attributes [41] [42]:

1. The ability to control and obtain resources based on requests from a User Equipment (UE).
2. The resource management framework shall be compatible with various Quality of Service (QoS) architectures across differing service provider platforms.

In the development of a resource management framework for voice, a voice pointer approach as illustrated in Figure 4-1 is proposed. In this approach it is considered effective to simply use Quality of Service (QoS) attribute mapping between differing network nodes for voice: where 1; 2; 3 or 4 is representative of a pointer or label that has a direct relationship to a specified Quality of Service (QoS) attribute [42] [43].

Each pointer would identify to a given traffic handling capability which is relevant to a particular user plane network element. The specification of a traffic handling behaviour provides sufficient information that allows the realisation of a particular Quality of Service (QoS) Bearer or Service Flow via that network element [42] [43]. The aim is that a set of traffic handling behaviours with predefined attributes could be prearranged at the entry point of each of the individual nodes, such that Quality of Service (QoS) for voice traffic is maintained across the pointers as traffic traverses each of the nodal elements, thereby also ensuring Quality of Service (QoS) at the ingress nodal point and invoking Quality of Service (QoS) at the egress nodal points [42] [43].

A typical network set-up will consists of a User Equipment (UE) invoking the Admission Control (AC) pointer demarcated for voice traffic. The Admission Control (AC) will in turn align to the Bearer or Service Flow pointer and the Bearer or Service Flow pointer would align to the transport network pointer. It is anticipated that this reservation will be held in state until the User Equipment (UE) hangs up or is terminated [42].

The above methodology deviates from the bandwidth or session negotiation model as it does not hold session information but rather trusts the source of the information and applies the relevant Quality of Service (QoS) forwarding treatment to the source traffic flow. It is recommended that such a methodology only become applicable to voice traffic [42].

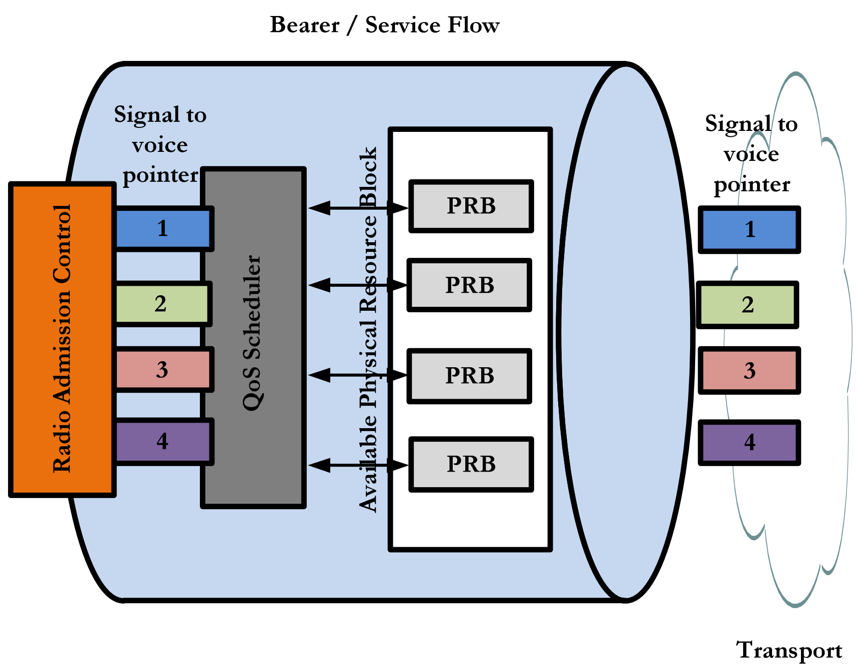

Figure 4-1. Pointer Approach to Voice Quality of Service (QoS) Co-operation [42]

4.3 Admission Control (AC)

Radio Admission Control (AC) admits or rejects requests for new connections between the Base Station (BS) and User Equipment (UE) depending on if the network will be able to fulfil the Quality of Service (QoS) criteria of a new connection request without compromising active sessions [44]. If LTE is considered as an example, in order to provide quality control on the physical resource block allocation, the Admission Control (AC) and packet Scheduling need to be Quality of Service (QoS) aware. The Quality of Service (QoS) aware packet Scheduling allocates the dynamically shared data channel to active radio Bearers based on predefined Quality of Service (QoS) requirements [28].

In order to sufficiently admit voice traffic in a packet switched network the first step is to ensure accurate identification of voice at the eNodeB or WiMAX Base Station (BS) of the network. The recognition of voice can be facilitated by means of shallow or Deep Packet Inspection (DPI), voice codec sniffing or any other methodology [42].

4.4 Admission Control (AC) to Bearer or Service Flow Resource Mapping

Considering LTE as an example, the necessary voice Traffic Flow Templates (TFT) shall be used to discriminate between different payloads using the IP header, such as IP address or Port numbers etc. It is anticipated that a repository shall be built relating the IP addresses of the User Equipment (UE) to the Bearer or Service Flow for voice. At this location the voice pointer is proposed. This voice pointer shall represent the most relevant Bearer or Service Flow applicable to voice [42].

As illustrated in Figure 4-2: when voice traffic is recognised at the eNodeB or WiMAX Base Station (BS) component of the network it is expected that the call Admission Control (AC) component will invoke the Radio Access Network (RAN) Bearer or Service Flow. On analysis of WiMAX and LTE it is initially mentioned that Guaranteed Bit Rate (GBR) QoS Class Identifier (QCI) 1, 2 or 3 and Service Flow Unsolicited Grant Service or Real-Time Polling Service (rtPS) or Extended Real-Time Polling Service (ertPS) are the most preferable Bearers or Service Flows to carry voice on the Radio

Access Network (RAN). This is due to the stringent latency, jitter and error loss requirements of voice [42]. The various LTE QoS Class Identifiers and WiMAX Service Flows have been tested to determine the most appropriate for voice traffic with the results presented in the later chapters.

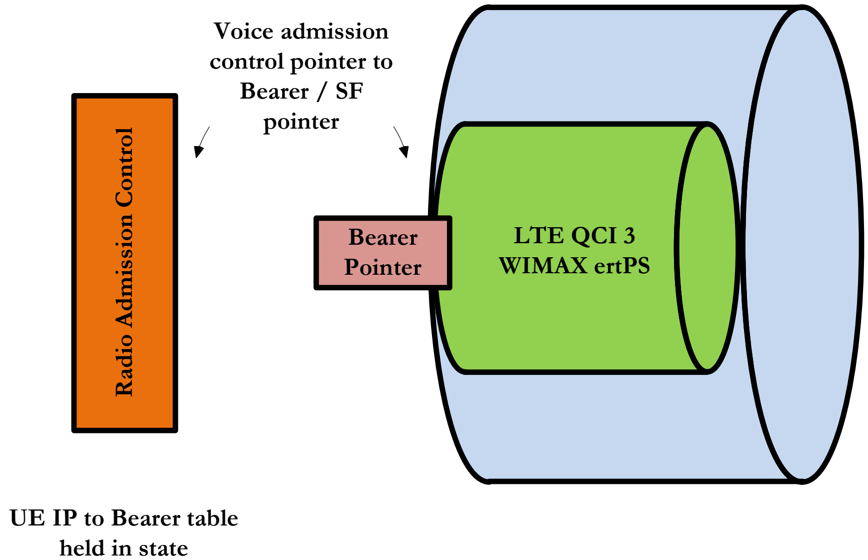

Figure 4-2. Radio Access Control to Bearer or Service Flow Pointer [42]

4.5 Bearer or Service Flow to Transport Mapping

Many providers face the situation where the constraint of resources does not only occur at the Radio Access Network (RAN) layer but also on the transport or backhaul layer. Taking this into account, there remains a need to provide voice quality control in the transport network [42]. It is anticipated that the transport network will be an all IP network.

With the advent of Metro Ethernet systems being deployed directly to the eNodeB or WiMAX Base Station (BS), the transport limitation issue shifts focus from limited resources to appropriate queuing needed to be applied on the edge of the transport network [42]. In order to provide sufficient voice quality control, route control and packet queuing parameters need to be coordinated with the radio network [45]. Route control refers to the Quality of Service (QoS) management function that is in charge of selecting the optimum routes in the transport network to guarantee the efficient use of transport resources. Route control also makes reference to link utilisation and buffer control on the backhaul network [45]. Packet queuing is in charge of implementing the appropriate Quality of Service (QoS) queuing decisions at the IP transport network nodes so that different flows receive the appropriate Quality of Service (QoS) treatment at every node.

Considering that Quality of Service (QoS) is required over the transport network there is a recommendation that Integrated Services (IntServ) and Differentiated Services (DiffServ) be utilised in the transport of 'all-IP' 4^{th} Generation networks. Integrated Services (IntServ) uses Resource Reservation Protocol (RSVP) to reserve bandwidth during a session set-up. If the sender receives a resource reservation confirmation returned from the receiver as an indication of Quality of Service (QoS) guarantee, it initiates the session [9] [46]. Integrated Services (IntServ) ensures strict Quality of Service (QoS), but each router in the transport network must implement Resource Reservation Protocol (RSVP) and maintain a per-flow state, which can cause difficulties in a large-scale network [9] [46].

Differentiated Services (DiffServ), on the other hand, does not require a signalling protocol and cooperation among nodes as the Quality of Service (QoS) level of a packet is indicated by the Differentiated Services (DiffServ) Differentiated Service (DS) field of the IP header [9] [46]. The Internet Engineering Taskforce (IETF) RFC 2474 and 2475 define the fundamental framework of the Differentiated Services (DiffServ) scheme. The Differentiated Services (DiffServ) architectural framework is such that each packet's header is marked with one of the standardised code points. Each packet containing the same code point receives identical forwarding treatment by routers and switches in the path. This obviates the need for state or complex forwarding decisions in core routers based on per flow bases [39] [47].

The ingress boundary router is normally required to classify traffic based on TCP/IP header fields. Differentiated Services (DiffServ) micro flows are subjected to regulation and marking at the ingress boundary router according to a contracted service level. Depending on the particular Differentiated Services (DiffServ) model, out-of-profile packets are either dropped at the boundary or marked with a different priority level, such as best-effort [39] [47]. These functions are termed as traffic conditioning in Differentiated Services (DiffServ) language. A Differentiated Services (DiffServ) flow along with similar Differentiated Services (DiffServ) traffic forms an aggregate. All subsequent forwarding and policing are performed on aggregates by Differentiated Services (DiffServ) interior nodes. As the interior nodes are not expected to perform an expensive classification function, their ability to process packets at high speeds becomes possible. Enforcement of the aggregate traffic contracts between Differentiated Services (DiffServ) domains is vital to providing Quality of Service (QoS) [39] [47]. However, the Admission Control (AC) modules must ensure that new reservations do not exceed the aggregate traffic capacity. These features make it possible to provide end-to-end services using Differentiated Services (DiffServ) architecture [39]. Table 4-1 below illustrates the mapping rules that may be applied to 4th Generation wireless network traffic flows [42].

Table 4-1. Differentiated Services (DiffServ) DHCP to LTE QoS Class Identifier (QCI) and WiMAX Service Flow Mapping

DHCP	LTE QCI	WiMAX SF
EF	1, 2	UGS
AF4	3, 4	ertPS
AF3	5, 7	rtPS
AF2	6	nrtPS
AF1	8	nrtPS
BE	9	BE

As illustrated in Figure 4-3, when the Bearer or Service Flow pointer has been invoked it is expected that the necessary transport pointer will be aligned, i.e. the LTE Bearer pointer will align to the appropriate transport Quality of Service (QoS) Service Flow as indicated by the pointer. On the transport side it is expected that the transport Bearer will take the form of Differentiated Services (DiffServ) Expedited Forwarding (EF) or

Multiprotocol Label Switching Experimental Bit (EXP) 5 which are the highest priority Service Flows [42] [48].

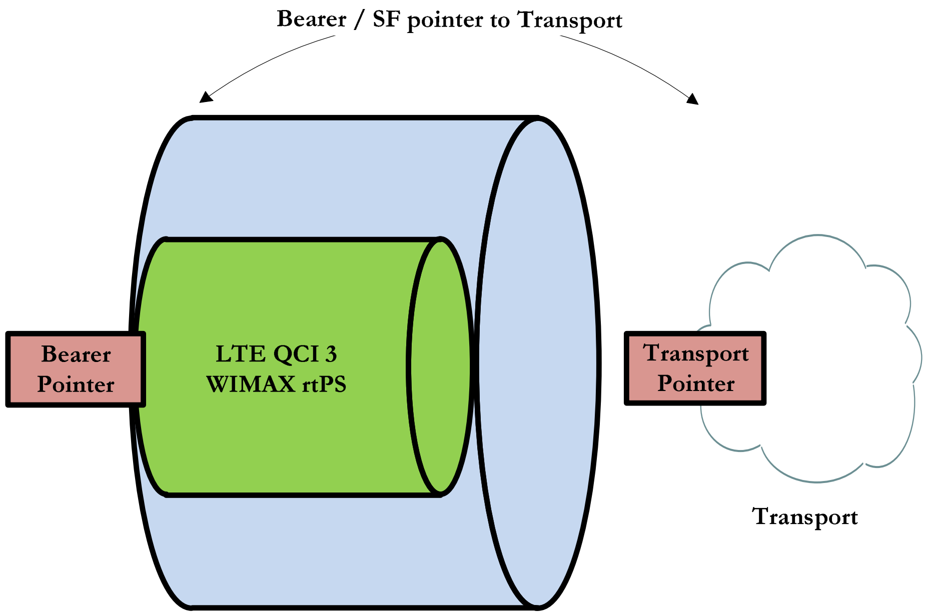

Figure 4-3. Bearer / Service Flow Pointer to Transport Pointer [42]

4.6 Conclusion

In this chapter a generic resource management framework for voice is presented, highlighting the voice pointer approach. Each pointer represents a specific traffic handling capability relevant to a particular user plane network element. Voice pointers representing the ingress of the Admission Control (AC), Admission Control (AC) to Bearer or Service Flow mapping and Bearer or Service Flow to Transportation mapping are illustrated, describing an alignment of voice traffic conditioning across each of the nodal elements.

The Admission Control (AC) element in a packet switched wireless network is investigated, and a recommendation that voice be recognised at this Admission Control (AC) component be made as well as trigger the Quality of Service (QoS) framework indicated by the pointer approach. An important part of the framework is the mapping of the Bearer or Service Flow to the backhaul network where it is recommended that

Differentiated Services (DiffServ) is used with a Differentiated Services Code Point (DSCP) of Expedited Forwarding (EF) or Multiprotocol Label Switching Experimental Bit (EXP) 5 is used for voice. These have been specified for real time traffic hence will result in the lowest possible latency, jitter and packet loss statistics for the transport network. Selection of Bearer or Service Flow for voice shall be presented in subsequent chapters.

In the following chapter the Admission Control (AC) component is highlighted with specific emphasis on recognising voice at the Admission Control (AC) component using Heuristic Analysis. This is the pattern recognition of voice protocols with the intention to recognise Over the Top voice traffic and trigger a Quality of Service (QoS) Service Flow or Bearer for admitted voice stream.

5 A Heuristic Analysis Approach to Admission Control (AC) for Voice in Packet Switched Wireless Networks

5.1 Introduction

As previously stated, radio Admission Control (AC) admits or rejects requests for new connections between the Base Station (BS) and User Equipment (UE) depending whether it will be able to fulfil the Quality of Service (QoS) criteria of a new connection request without compromising already active sessions [44]. Hence radio Admission Control (AC) plays an important role in determining the overall Quality of Service (QoS) for voice over a packet switch wireless network. If WiMAX and LTE are considered as examples, for a WiMAX Service Flow, Quality of Service (QoS) parameters such as Minimum Reserve Traffic Rate (MRTR) and Maximum Sustained Traffic Rate (MSTR) are consulted in making Admission Control (AC) decisions [12] [44]. For LTE, Bearer level parameters such as Allocation and Retention Priority (ARP), Guaranteed Bit Rate (GBR) attributes and Aggregate Maximum Bit Rate (AMBR) are consulted in making Admission Control (AC) decisions [12] [44].

The parameters listed above make reference to resources that need to be allocated on a per request basis however Allocation and Retention Priority (ARP) in LTE specifies the relative importance of an Evolved Packet System (EPS) Bearer as compared to other Evolved Packet System (EPS) Bearers for the allocation and retention of that Bearer. In situations where resources are scarce, network elements can use the Allocation and Retention Priority (ARP) to prioritize Bearers with a high retention priority over Bearers with a low retention priority when performing Admission Control (AC) [44]. Allocation and Retention Priority (ARP) is essential for Bearer retention however does not assist in the initial recognition and admission of a voice Bearer. It has been noted that prioritisation of voice is key in allowing Quality of Service (QoS) for voice packets; however this has to be placed in the context of not compromising the performance of other services [44].

The following chapter proposes the establishment of a new Admission Control (AC) scheme utilising Heuristic Analysis for the recognition and priority admittance of voice calls at various levels of the Open Systems Interconnection (OSI) stack [16]. The new scheme is considered in order to provide improved voice quality by accurately recognising and admitting voice traffic over all other services in the packet switched wireless network [16].

In the innovative solution presented, Deep Packet Inspection (DPI) at the eNodeB or Access Service Network Gateway (ASN-GW) level is merged with a mathematically modelled Heuristic Objects and Axioms Algorithm. The model proposed is based on the analysis of voice protocol set-up messages for the successful recognition of voice traffic. This structure is further combined with a voice specific variable Admission Control (AC) arrangement in order to provide a combined Heuristic Analysis and Admission Control (AC) scheme.

Session Initiation Protocol (SIP) and Google Talk are analysed to derive a pattern of recognition. The probability of recognition in conjunction with the combined Heuristic Analysis and Admission Control (AC) scheme is simulated to determine the blocking probability for voice calls in a packet switched wireless network [16].

5.2 Heuristic Analysis Approach to Classification of Voice

In order to sufficiently identify and admit voice calls in a packet switched network the first step is to ensure intelligent packet recognition at the eNodeB or WiMAX Base Station (BS) level. The recognition of voice traffic can be conducted using Shallow or Deep Packet Inspection (DPI). With reference to the Open Systems Interconnection (OSI) model as indicated in Figure 5-2, shallow packet inspection inspects headers at layer 3 and ports at layer 4, while Deep Packet Inspection (DPI) inspects headers at layers 4 through 7 and the payload [49].

Depending on the manufacturer's choice, voice can be recognised in the following manner:

1. IP header on the network layer.
2. Port on the transport layer.
3. Session Initiation Protocol (SIP), Real Time Protocol (RTP) on the session layer.
4. Over The Top (OTT) voice application on the application layer.

In the research conducted it is proposed that recognition of voice can be done using Deep Packet Inspection (DPI) at the eNodeB or WiMAX Base Station (BS) level. Dolgonow et al [50] specify a Deep Packet Inspection (DPI) engine on the LTE Serving Gateway (S-GW) at a centralised point however with this option the signalling and traffic flow overhead increases between the access and the core of the network before any further processing takes place [16].

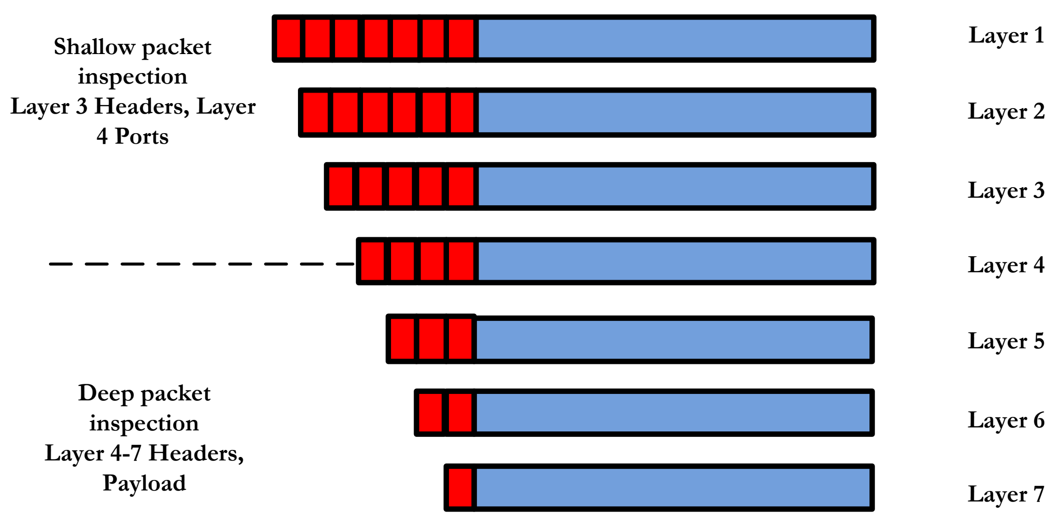

Figure 5-1. Shallow and Deep Packet Inspection (DPI)

The newly proposed Heuristic Analysis scheme requires the addition of a Deep Packet Inspection (DPI) engine because the scheme shall use Deep packet Inspection (DPI) to inspect packets as part of the analysis process. The locality of the Deep Packet Inspection (DPI) engine is proposed to be positioned at the eNodeB or WiMAX Base Station (BS) level so as to recognise Over the Top (OTT) voice or any voice traffic at the access point of the wireless network. Figure 5-2 depicts the addition of the Deep Packet Inspection (DPI) module to the eNodeB or WiMAX Base Station (BS) [16].

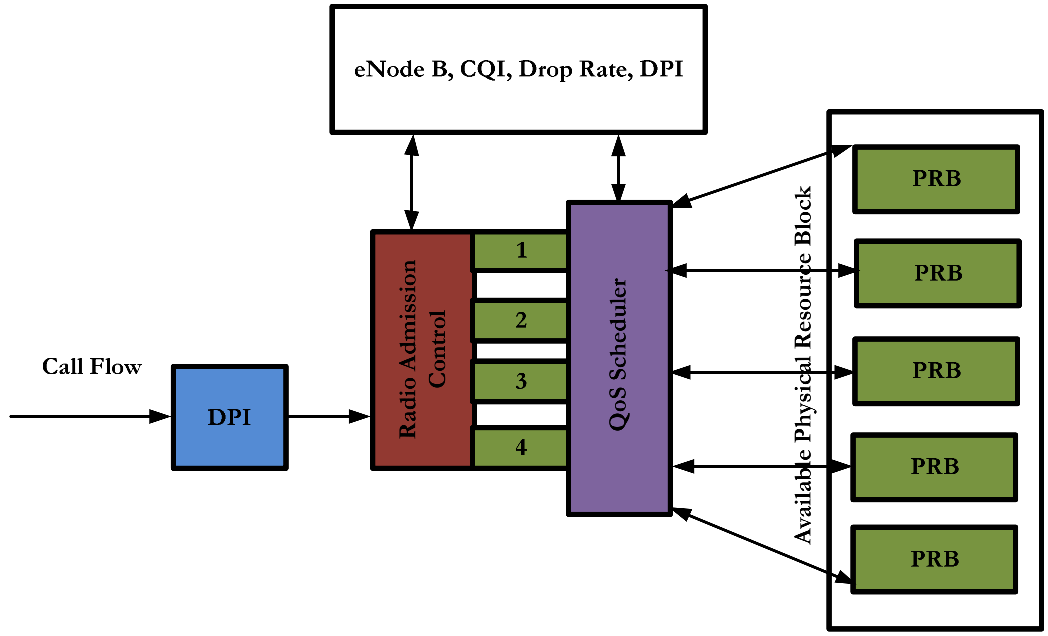

Figure 5-2. Incorporation of Deep Packet Inspection (DPI) in a Packet Switched Wireless Network [16]

5.2.1 Overview of Heuristic Analysis

Heuristic Analysis is a method used to adapt algorithms to cater for adaptive input parameters that will yield a desired output even if certain inputs parameters change or have a variance. Heuristic analysis is paramount to contemporary problems that tend to be intricate and relate to analysis of large data sets. Even if an exact algorithm can be developed for these problems its time or space complexity may turn out to be challenging, yet in reality it is often sufficient to find an approximate or partial solution [51].

A generic problem-solving strategy for deriving Heuristics that is based on a set of definitions and assumptions is presented below. These constitute the accepted basis for a generalised, self-consistent algorithm derivation. The axioms presented are generalisations, which allow for deductions of useful methodology. Derivation is done by constructing and analysing graphic symbols of interacting objects without identifying precise objects. In the work presented below Heuristic algorithms are presented that lead to some approximations to the solution of protocol pattern recognition for voice traffic [52].

The pattern recognition process is visualised as being composed of interacting objects. Conceptually, it is seen that these objects interact or inter-relate to each other. Interaction of objects is defined as a set of recognition pattern modifiers or a set of recognition patterns. A well-defined pattern contains objects, attributes, a wanted or unwanted effect and root causes. Root causes are defined as causal attributes that can be linked to an unwanted effect [52].

The strategy for deriving Heuristics is based on the following assumptions (Ax_1 – Ax_6) that arise from self-evident truths, experience and intuition. These are all selected to support simplification of analysis [52]:

Ax_1 Problems can be analysed in terms of interacting objects.

Ax_2 Interacting objects can be simplified to pairs of objects.

Ax_3 Interaction of objects can be reduced to one attribute from each object supporting an effect that is acting on a third attribute (of an initial object or of a third object).

Ax_4 Attributes require no metrics in a conceptual analysis.

Ax_5 Effective simplification for problem analysis and solution can be achieved with a minimum set of objects.

Ax_6 Problem situations must be reduced to unwanted effects of which is to be solved at a time.

In terms of voice pattern recognition, the following is an example of a possible recognition model where an algorithmic pattern exists. This recognition model consists of a 4 sequence recognition process (objects) with variances in inter-arrival times of each of the attributes.

Where: O_1 = Object 1.
O_2 = Object 2.
O_3 = Object 3.
O_4 = Object 4.

For an accurate response from the recognition process it can be said that the probability *P(E)* of the sum of positive acknowledgments from each of the objects will equal to a true result represented by the number 1 as illustrated in Equation 5-1.

$$\mathrm{P(E)} = \sum_{x \in E}^{n} f(O_x) \qquad 5\text{-}1$$

A schematic of the Heuristic recognition algorithm is illustrated in Equation 5-2 as follows;

$$\begin{array}{llllll}
O_1 — A_1 & & & & & \\
& \rangle — W_1 — A_{12} & & & & \\
O_2 — A_2 & & \rangle — W_2 — A_{123} & & & 5\text{-}2 \\
& O_3 — A_3 & & \rangle \ 1 & & \\
& & O_4 — A_4 & & &
\end{array}$$

Where: A_1 = Assumed ideal arrival time for Object 1.

A_2 = Assumed ideal arrival time for Object 2.

A_3 = Assumed ideal arrival time for Object 3.

A_4 = Assumed ideal arrival time for Object 4.

W_1 = Assumed ideal arrival time for combined Object 1 and 2.

W_2 = Assumed ideal arrival time for combined Object 1, 2 and 3.

As is illustrated in Equation 5-2 the recognition process can be represented by a Heuristic algorithm that takes into consideration variances in inter-arrival times of each of the Objects. The methodology listed is applied to the voice system model below so as to create a Heuristic Analysis algorithm for the recognition of voice traffic (especially Over the Top (OTT) voice) in an all IP network.

5.3 Heuristic Analysis Voice System Model Derivation and Associated Experimental Technique

In order to sufficiently analyse voice traffic the system model as per Figure 5-3 was set-up [16]. The purpose of this experiment was to derive the Objects for the new Heuristic algorithm to be used for the recognition of voice traffic in a packet switched wireless network; hereby known as the experimental variables. Discovery of the experimental variables consisted of the analysis Session Initiation Protocol (SIP) and Google Talk voice protocols. Session Initiation Protocol (SIP) and Google Talk were chosen because they each operated on different layers of the Open Systems Interconnection (OSI) model, i.e. the session layer and the application layer. This demonstrates the flexibility of the new Heuristic algorithm across the different layers of the Open Systems Interconnection (OSI) model.

5.3.1 Heuristic Analysis Voice System Model Experimental Design

The Heuristic Analysis voice system model test bed was constructed and configured with the following components:

1. A WiMAX Subscriber Station (SS)
2. A WiMAX Base Station (BS).
3. WiMAX Access Service Network Gateway (ASN-GW).
4. Access to the WiMAX network management element.
5. Laptop connected at the WiMAX SS / MS.
6. Laptop connected to the Ethernet backhaul.
7. Connectivity to an Ethernet backhaul network.
8. Connectivity to the Internet.
9. Wireshark Network Protocol Analyser.
10. A Softswitch.

The first test laptop was connected to the WiMAX link and terminated a Point-to-Point Protocol (PPP) session onto a service provider Broadband Radius Aggregation Server (BRAS) for IP address allocation and internet access. The service provider in turn routed traffic from the Point to Point Protocol (PPP) session to the internet which was then directed to the second test laptop [16]. This enabled direct connectivity from the laptop connected to the WiMAX link to the laptop connected in the laboratory, as well as connectivity to the Softswitch.

For the Session Initiation Protocol (SIP) experiment the carrier grade Softswitch was exposed to the internet to terminate Session Initiation Protocol (SIP) registrations. A Session Initiation Protocol (SIP) soft phone application was used on the WiMAX connected laptop to register a Session Initiation Protocol (SIP) call onto the Softswitch. The Softswitch would then route the call via a broadband network onto an IP phone in the laboratory. This experiment was repeated on several occasions with the Wireshark Network Protocol Analyser [53] capturing traces of the Session Initiation Protocol (SIP) registrations.

For the GoogleTalk experiment, the relevant GoogleTalk software was installed on both the laboratory laptop and the laptop connected to the WiMAX link. Several calls were terminated from one laptop to the other with the Wireshark Network Protocol Analyser [53] capturing traces of the GoogleTalk set-up procedure [16].

The Wireshark Network Protocol Analyser [54] was used as the measuring instrument as it is a network troubleshooting, analysis and communications protocol tool that accurately captures traces from an Ethernet port. Wireshark is cross-platform analyser, using the GTK+ widget toolkit to implement its user interface, and using pcap to capture packets. Wireshark runs on various Unix-like operating systems including Linux, Mac OS X, BSD, Solaris, and Microsoft Windows [54] [55].

The Wireshark Network Protocol Analyser is further discussed in Appendix A.

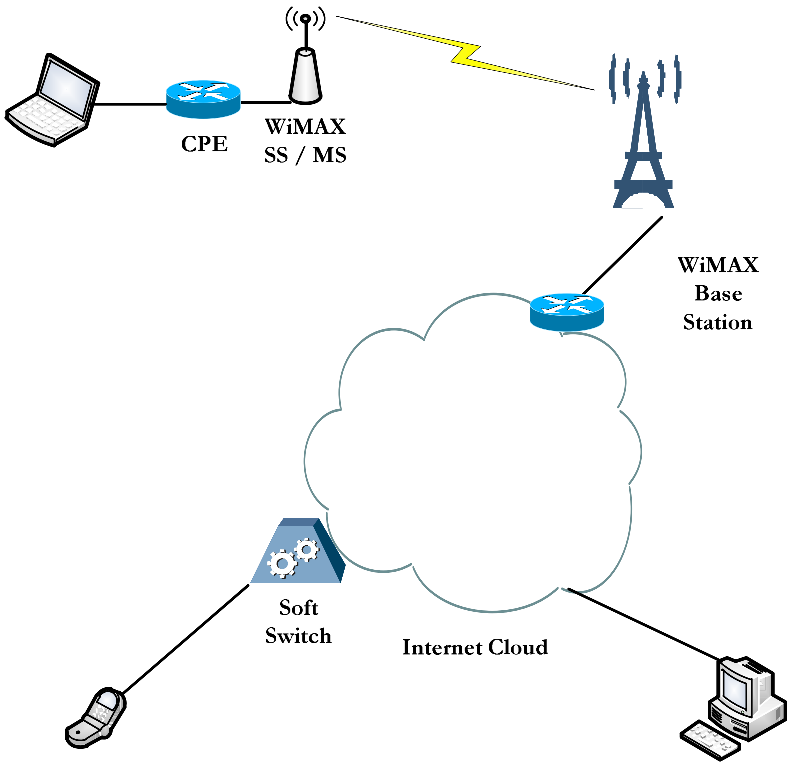

Figure 5-3. Lab set-up for Voice Packet Pattern Analysis [16]

5.3.2 Analysis of Session Initiation Protocol (SIP)

The experiments were conducted according to the experimental design outlined above. More than 20 individual traces were done in accordance with the experimental procedure. On analysis of the Wireshark Network Protocol Analyser [53] traces; for the Session Initiation Protocol (SIP) the following patterns as per Figure 5-4 were recognised during a call set-up (this is standard Session Initiation Protocol (SIP) invite):

1 *SIP/SDF request invite*
2 *SIP Status 100 trying*
3 *SIP Status 407 Proxy Authentication*
4 *SIP Request Acknowledgement*
5 *SIP Status 100 trying*

Figure 5-4. Session Initiation Protocol (SIP) call set-up pattern recognition

The call set-up request was initiated on several occasions with the above mentioned sequence occurring on every occasion; however new information discovered was that a hard rule could not be set for recognition pattern because the spanning tree protocol as well as Point-to-Point Protocol (PPP) Link Control Protocol requests occurred in majority of the test-sets thereby interrupting the recognition process by arriving in-between the sequence of protocol messages. Based on all the test-set results, the probability of recognising the Session Initiation Protocol (SIP) invite structure in a sequenced approach was derived to be 50%. This result which is deemed to be insufficient is illustrated in Figure 5-5 [16].

Equation 5.3 depicted in a schematic form illustrates the inclusion of all Objects for the experiment above which results in a probability of recognition of 0.5. Where, each of the Session Initiation Protocol (SIP) protocol characteristics as per Figure 5-4 represents an Object.

$$\left.\begin{array}{l} O_1 \\ O_2 \\ O_3 \\ O_4 \\ O_5 \end{array}\right\} 0.5 \tag{5-3}$$

If a non-numbered recognition categorisation is used, the probability of recognition significantly increases to 80% since this method allows the flexibility of matching and avoiding protocol messages that come in-between the Session Initiation Protocol (SIP) message structure, however multiple call set-ups from the laptop reduces the recognition pattern back to 50% [16].

The non-numbered categorisation alludes to the fact that the sequence of Figure 5-4 does not follow immediately after each other but rather has other protocol messages running in-between the sequence of messages. The phenomenon results in the addition of an Axiom representing the variance in inter-arrival times between Objects in receiving spanning tree protocol as well as Point-to-Point Protocol (PPP) Link Control Protocol requests between Objects.

During the experimental process it was discovered that a non-numbered categorisation used in combination with a fixed sample rate duration of 0.191ms (the time between the start of the measurement and the end of the measurement = 0.191ms) the recognition pattern increases to 90%. This was discovered by recognising that within this time period only a singular Session Initiation Protocol (SIP) call registration as per the Objects defined will complete, thereby eliminating multiple call set-ups from disrupting the recognition process. Equation 5.4 depicted in a schematic illustrates the inclusion of all Objects (O) and Axioms (A) representing the variations in inter-arrival times that result in a probability of recognition of 0.9.

$O_1 — A_1$
$\backslash$
$/$ — $W_1 — A_{12}$
$O_2 — A_2$ $\backslash$
$/$ — $W_2 — A_{123}$
$O_3 — A_3$ $\backslash$
$/$ — $W_3 — A_{1234}$ 5-4
$O_4 — A_4$ $\backslash$
$O_5 — A_5$ — 0.9
$/$
O_6 — —

Where: O_1 = SIP Protocol Message Figure 5-4 line 1.
O_2 = SIP Protocol Message Figure 5-4 line 2.
O_3 = SIP Protocol Message Figure 5-4 line 3.
O_4 = SIP Protocol Message Figure 5-4 line 4.
O_5 = SIP Protocol Message Figure 5-4 line 5.
O_6 = Fixed sample rate duration of 0.191ms.
A_1 = Assumed arrival time for Object 1.
A_2 = Assumed arrival time for Object 2.

A_3 = Assumed arrival time for Object 3.

A_4 = Assumed arrival time for Object 4.

A_5 = Assumed arrival time for Object 5.

W_1 = Assumed arrival time for combined Object 1 and 2.

W_2 = Assumed arrival time for combined Object 1, 2 and 3.

W_3 = Assumed ideal arrival time for combined Object 1, 2, 3 and 4.

The traces further revealed that if the realm IP address of the Softswitch is used in combination with the above methodology, the voice pattern was recognised in all of the test samples [16]. The realm IP address of the Softswitch is presented in all Session Initiation Protocol (SIP) messages because this is the termination point of the Session Initiation Protocol (SIP) session. It has to be noted that for a telecommunications operator this would be acceptable because the operator may have a centralised Softswitch, however for Session Initiation Protocol (SIP) calls between terminal devices that use a SIP agent this will not function correctly because the IP address of the terminal device is unique for each device and would be difficult to adjust in the algorithm for each user. Equation 5.5 depicted in a schematic illustrates the inclusion of the Softswitch IP Objects (O) resulting in a probability of recognition of 1.

$O_1 — A_1$

$\quad\quad\quad \rangle — W_1 — A_{12}$

$O_2 — A_2 \quad\quad\quad\quad \rangle — W_2 — A_{123}$

$\quad\quad\quad O_3 — A_3 \quad\quad\quad \rangle — W_3 — A_{1234}$ 5-5

$\quad\quad\quad\quad\quad\quad O_4 — A_4$

$\quad\quad\quad\quad\quad\quad\quad\quad O_5 — A_5 —$

$\quad\quad\quad\quad\quad\quad\quad\quad O_6 — — — \quad 1$

$\quad\quad\quad\quad\quad\quad\quad\quad O_7 — \;—$

Where: O_7 = Realm IP address of the Softswitch where SIP is terminating

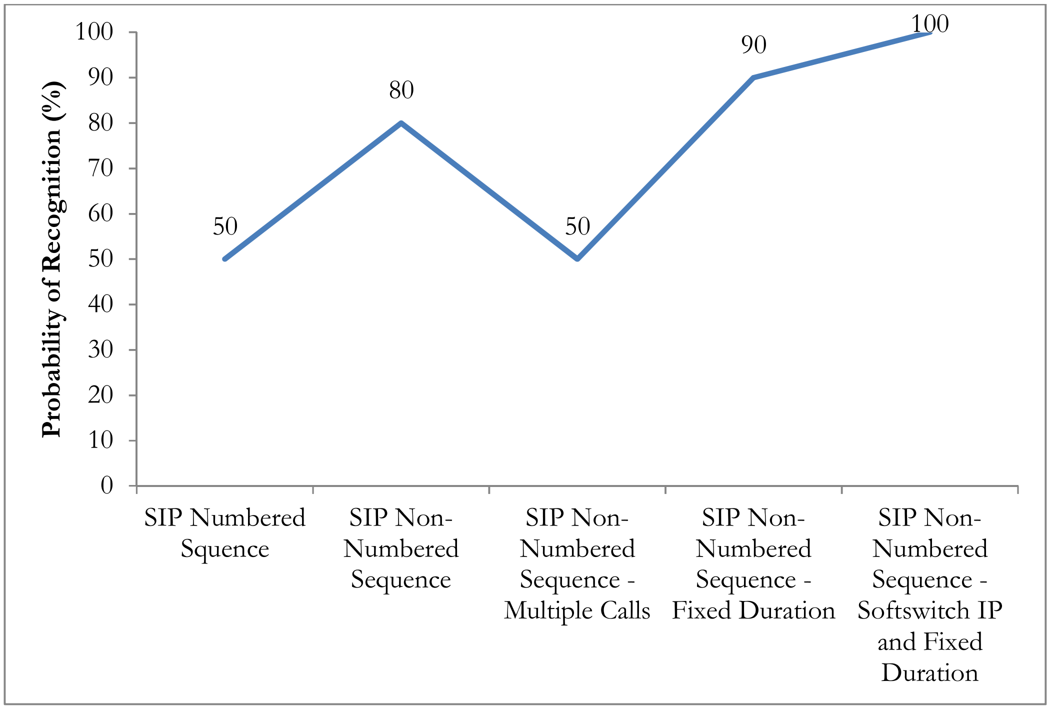

Figure 5-5. Analysis of Session Initiation Protocol (SIP) Packets [16]

5.3.3 Analysis of Google Talk

The experiments for the Google Talk application were carried out according to the experimental design portrayed earlier. More than 20 individual traces were conducted based on the experimental procedure. On analysis of the Wireshark Network Protocol Analyser [53] traces for the Google Talk application it was discovered that the ability to recognise patterns that were similar to Session Initiation Protocol (SIP) was challenging.

The predominant protocols highlighted in the traces include the Transmission Control Protocol (TCP) and Hypertext Transfer Protocol (HTTP). This in turn meant that the application within the HTTP protocol would need to be recognised.

On further analysis the following recognition sequence had emerged:

$_1$ *HTTP GET /create_session HTTP/1.1*

$_2$ *Host: relay.google.com\r\n*

$_3$ *User-Agent: gtalkplugin\r\n*

Figure 5-6. Analysis of Google Talk set-up Sequence [16]

With each of the Google Talk application call set-up characteristics representing an Object and the variance in inter-arrival times between Objects as well as the variance between Objects (Google Talk application messages) in receiving spanning tree protocol, as well as Point-to-Point Protocol (PPP) Link Control Protocol requests in-between Objects represented by Axioms. Based on this the following Heuristic algorithm can be derived as per Equation 5-6.

$$\begin{array}{lll} O_1 — A_1 & & \\ & \rangle — W_1 — A_{12} & \\ O_2 — A_2 & & \rangle \; 1 \\ & O_3 — A_3 & \end{array} \qquad \text{5-6}$$

Where: O_1 = Google Talk Application Message Figure 5-6 line 1.

O_2 = Google Talk Application Message Figure 5-6 line 2.

O_3 = Google Talk Application Message Figure 5-6 line 3.

A_1 = Assumed arrival time for Object 1.

A_2 = Assumed arrival time for Object 2.

A_3 = Assumed arrival time for Object 3.

W_1 = Assumed arrival time for combined Object 1 and 2.

The above voice set-up recognition methodology combined with gleaning of the source IP address of the User Equipment (UE) provides the opportunity to successfully recognise Over the Top (OTT) voice and enable priority admittance of this traffic. In the later chapters it shall be further illustrated that this priority admittance results in the enablement of a Quality of Service Bearer or Service Flow over a packet-switched wireless network.

Considering the findings above, an innovative approach would be to combine the Heuristic Analysis recognition algorithm with an appropriate Admission Control (AC) structure so as to provide priority admittance to voice traffic. The advantage of Deep Packet Inspection is that filtering rules within the engine can be created to trigger a desired output. As an example the filtering rule as per the Heuristic Analysis recognition algorithm becomes an element of the radio Admission Control (AC) structure.

Radio Admission Control (AC) in 4th Generation (4G) networks is presented below with a view to amalgamate the Heuristic Analysis recognition algorithm with a newly proposed voice Admission Control (AC) scheme.

5.4 Radio Admission Control (AC) in 4G Networks

In 4th Generation networks Bearer and Service Flow parameters are largely used in constructing Admission Control (AC) decisions. Based on the Admission Control (AC) decisions, resources are allocated accordingly. If we consider a multi-class system, the manner in which resources are allocated can be divided into three categories i.e. [44]:

1. Complete sharing scheme.
2. Complete partitioning scheme.
3. Virtual partitioning scheme.

Within the complete sharing scheme all users share the same resources and all calls are treated equally, therefore there is no Quality of Service (QoS) priority admittance class. This is contrary to the requirement for voice calls. With the complete partitioning scheme resources are split up into several parts, this enables different classes of service to have a set resource allocation, however if resources for a particular class are used up then the call is not admitted. With this method radio resource utilisation is decreased [44].

As illustrated in Figure 5-7 below; the virtual partitioning scheme divides traffic into two groups each allocated a nominal amount of physical resource; however underutilised resources can be used by an overloaded group. The scheme is more efficient than the complete partitioning scheme but also leads to decreased utilisation of the radio resource [44]. This is the optimal solution because it facilitates a priority admittance class for voice calls.

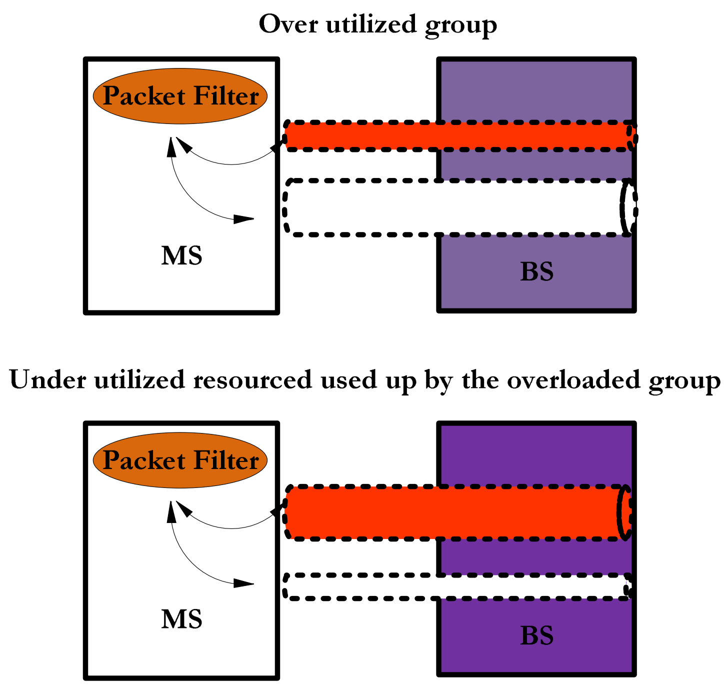

Figure 5-7. Radio Admission Control (AC) Virtual Partitioning Scheme

5.5 Voice Specific Admission Control (AC) Model

As the Heuristic Analysis recognition algorithm only deals with the initial recognition of voice traffic it is proposed that a new Admission Control (AC) scheme is developed to consider the priority admittance of voice traffic. In the proposed Admission Control (AC) scheme a voice service class is added to the virtual partitioning scheme. Inherently it is suggested that within the Admission Control (AC) mechanism only two traffic classes exist; namely voice and other, where other represents all other services. Other services can be split into different classes however for the purposes of this research this is not deliberated. It is proposed that Scheduling and not Admission Control (AC) be positioned as the differentiator between all other services. If LTE is considered as an example, a differing QoS Class Identifier (QCI) can be allocated to differing services as mapped by the various network operators [10]. If WiMAX is considered as an example, the differing Service Flows shall be allocated to differing services as mapped by the network operator [8].

In the development of the new voice specific Admission Control (AC) scheme system model the following methodology shall be taken into consideration [16]:

1. Total system capacity shall be as efficiently utilised as possible.
2. The voice service shall be recognised as per an effective Deep Packet Inspection (DPI) platform based on Heuristic Analysis methods.
3. The voice class shall have Admission Control (AC) priority but not consume overall resources so as to cause degradation of other services. For this purpose a system factor of p is introduced for operators to utilise such that the voice class shall not occupy more than $(1-p)$ of total system capacity.
4. The other services class shall have the ability to occupy total bandwidth resources if there are no voice calls made, hence maximum physical resources become available for other services.

The system model for the concept listed above is as follows:

Where:
Sv = voice services.
So = all other services.
Ct = total system capacity.
n = number of voice calls.
m = number of other calls.
B = instantaneous system bandwidth.
Bv = instantaneous bandwidth allocated for voice.
Bo = instantaneous bandwidth allocated for other services.
Br = bandwidth required by new call.
p = factor of reserve bandwidth.

Based on the model above the potential capacity allocation of the two differing Admission Control (AC) classes can be represented by Equation 5-7 for voice and Equation 5-8 for other [16];

$$0 \leq Sv \leq Ct(1-p) \tag{5-7}$$

$$0 \leq So \leq Ct \tag{5-8}$$

Therefore the new proposed Admission Control (AC) rules are listed as follows [16]:

1. For a recognised voice call admission is true if;

$$\sum_{i=1}^{n} Bv(i) + \sum_{j=1}^{m} Bo(j) + Ct(p) + Br \leq Ct \tag{5-9}$$

2. For other services, call admission is true if;

$$\sum_{i=1}^{n} Bv(i) + \sum_{j=1}^{m} Bo(j) + Br \leq Ct \tag{5-10}$$

The factor p can be determined by the network operator as to the traffic management criteria required by the network operator. For areas with high voice and low data demand this value can range from;

$$0.1 \leq p \leq 0.25 \tag{5-11}$$

Alternatively for high data areas;

$$0.25 \leq p \leq 0.75 \tag{5-12}$$

3. For the purposes of adding priority to specific voice calls such as emergency services, the Allocation and Retention Priority (ARP) mechanism in LTE and Priority mechanism in WiMAX shall be used.

The efficient utilisation of the overall network capacity in terms of voice calls can be further observed as there is a built-in reserve margin *p*; however the factor *p* trades-off this inefficiency for better quality of experience on other services. *p* can be changed to suit the operator requirements [16].

The number of calls possible through the proposed scheme is determined by the codec plus overheads used and the system capacity minus a reserve margin *p*. This figure will be positive because it is impractical for the codec rate to be less than the system capacity minus the system reserve margin, especially considering that on average the codec rate plus overheads shall be 50 Kilobits/s. The number of call calls possible shall be rounded down to the nearest integer in order to facilitate the Erlang B formula. According to the Erlang B formula the probability of blocking can be stated as per Equation 5-13 [16];

$$Pb = \frac{\frac{E^{\frac{Ct(1-p)}{R}}}{\left(\frac{Ct(1-p)}{R}\right)!}}{\sum_{i=0}^{\frac{Ct(1-p)}{R}} \frac{E^i}{i!}} \qquad \text{5-13}$$

Where: *Pb* = probability of blocking.

E = total amount of traffic offered in Erlang.

R = codec data rate used.

Considering that the Heuristic Analysis amalgamated methodology listed above is to offer admission for the purpose of ensuring Quality of Service (QoS) for voice calls, it can be further said that if a call is not recognised then the call cannot be admitted as a quality assured voice call. If a call is not recognised then for the purposes of this research it is classified as blocked.

Equation 5-14 illustrates the probability of blocking considering the deployment of Deep

Packet Inspection (DPI) with the voice specific Admission Control (AC) scheme. The probability of voice call blocking for the availability capacity on the system in relation to the Deep Packet Inspection (DPI) Heuristic Analysis recognition algorithm is a mutually exclusive event. However from a mathematical perspective as well as an overall system probability of blocking perspective it is accumulative and is represented as follows [16]:

$$Pb = \frac{\frac{E^{\frac{Ct(1-p)}{R}}}{\left(\frac{Ct(1-p)}{R}\right)!}}{\sum_{i=0}^{\frac{Ct(1-p)}{R}} \frac{E^i}{i!}} + Pb_{dpi} \qquad 5\text{-}14$$

Where: Pb_{dpi} = probability of the necessary DPI device not recognising a voice call.

5.5.1 Simulation of System Blocking Probability for the Session Initiation Protocol (SIP) Heuristic Analysis Recognition Algorithm

The purpose of the system blocking probability Session Initiation Protocol (SIP) Heuristic Analysis recognition algorithm simulation is to demonstrate how the probability of blocking for voice calls changes with different Heuristic Analysis recognition algorithms. The experimental variables include Equations 5-3, 5-4, 5-5 in combinations with the Erlang B probability of blocking formula.

The experimental design consisted of the simulation of overall system probability of blocking as per Equation 5-14 where the probability of call blocking in conjunction with the probability of non-recognition of voice is exemplified. A continuously increasing pattern of Erlang traffic from 0 to 20 Erlangs was used with fixed number of 20 lines (i.e. fixed system capacity, p and voice codec). This is done to show the effect of non-recognition of voice calls via the Heuristic Analysis methods. MATLAB [56] was used to simulate the experiment. MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualisation, data analysis, and numerical computation [57]. Further information on MATLAB and the simulations are provided in Appendix A.

The experiment was conducted in accordance with the experimental design. As is illustrated in Figure 5-8, significant results illustrate that the overall blocking probability increases with an increase of the probability of non-recognition of a voice call. The Session Initiation Protocol (SIP) numbered sequence approach performs worse than the Session Initiation Protocol (SIP) non-numbered sequence approach incorporated with a fixed time duration and Softswitch IP. As was anticipated the graphs tracked the Heuristic Analysis recognition algorithm and demonstrated that the probability of call blocking in addition to Heuristic Analysis recognition remained accumulative.

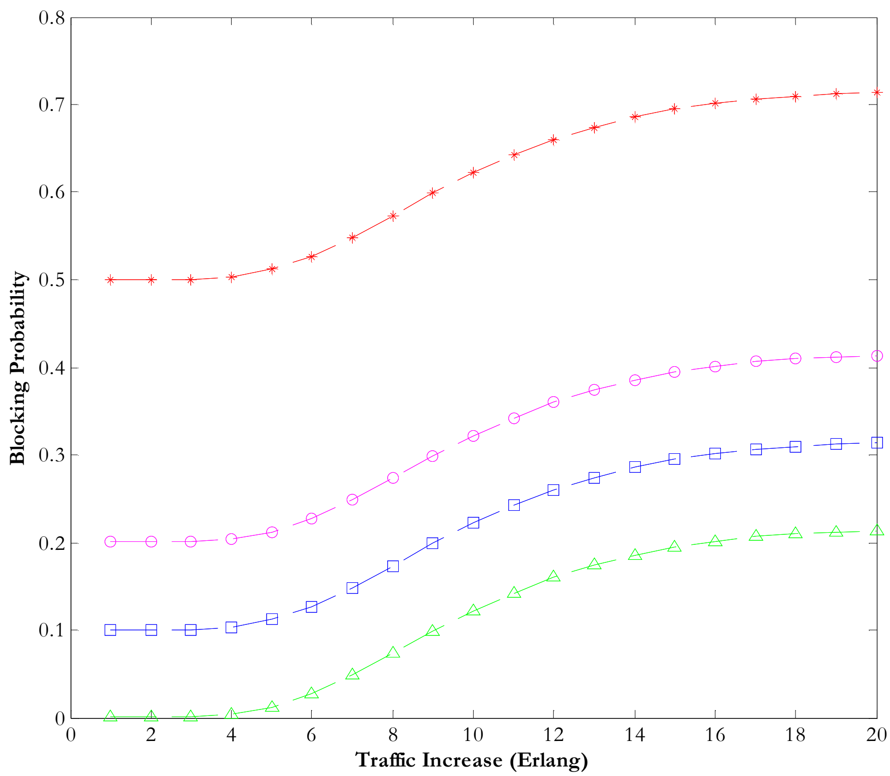

Figure 5-8. Analysis of Session Initiation Protocol (SIP) packets with Recognition Sequence

Where: --+--+-- = SIP numbered sequence,

--o--o-- = SIP non-numbered sequence,

--□--□-- = SIP non-numbered sequence with fixed duration, and,

--Δ--Δ-- = SIP non-numbered sequence with fixed Softswitch IP and time duration.

5.5.2 Simulation of a Changing p in the Voice Specific Admission Control Model

p is the factor of reserve bandwidth that operators can select in order to cater for high or low voice traffic scenarios on a per sector or cell basis. The purpose of the following experiment is the simulation of Equation 5-13 in order to determine the nature of p in relation to the probability of blocking for voice calls within the framework of the voice specific Admission Control (AC) scheme. The notion behind p is to create a balance between voice traffic and other traffic such that voice does not consume all available physical resources in a sector or cell. It was anticipated that as p increases the probability of voice call blocking will increase because the physical resource reserve margin for other traffic increases.

The experimental variables and experimental design consisted of the simulation of overall system probability of blocking as per Equation 5-13 where the probability of call blocking was measured whilst increasing the factor p from 0.1 to 0.75. The number of telephony or call lines in the simulation was based on modelling an LTE system with the following assumptions:

1. System has a single carrier with 1.4 MHz spectrum.
2. Voice codec rate including all possible overheads of 42 Kilobits/s.
3. Based on the above the number of possible lines equals to 120.
4. Erlang traffic rate is kept at a constant 100 Erlangs.

MATLAB [56] was used to simulate the experiment where the parameters listed above as well as Equation 5-13 are entered into the simulation system. Further information on MATLAB and the simulations are provided in Appendix A. The experiment was conducted in accordance with the experimental design.

The analysis of the results is depicted in a graphical overview as per Figure 5-9. The significant result that can be seen in the simulation is that when p increases the probability of blocking increases which was in accordance with the expected results. This phenomenon is due to the operator allocated bandwidth for voice traffic decreasing in line with an increasing reserve margin.

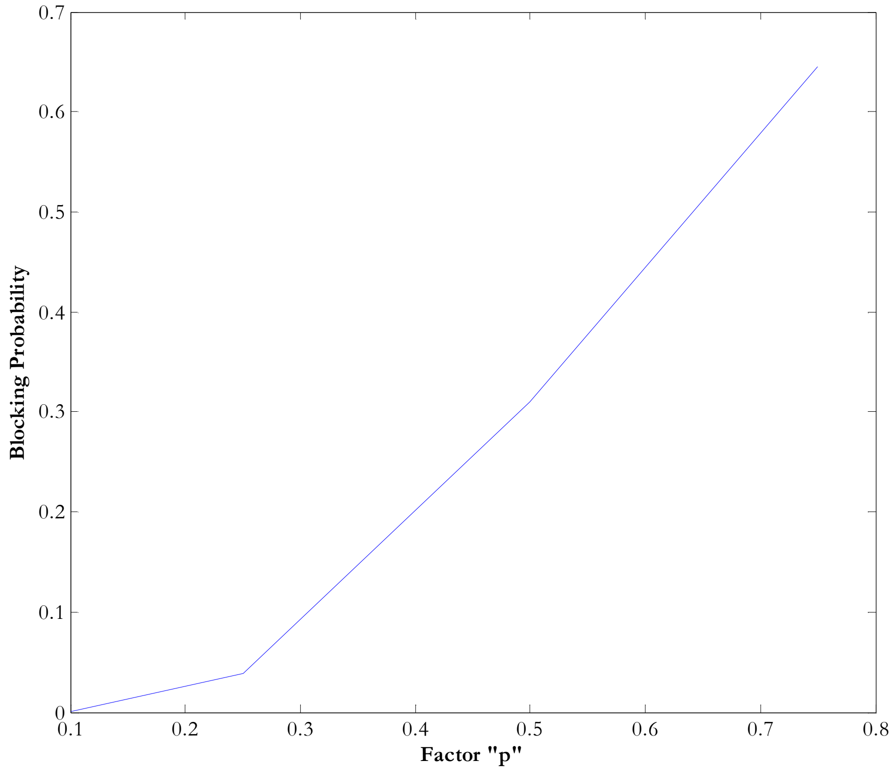

Figure 5-9. Analysis of Changing *p* in terms of Blocking Probability

5.6 Conclusion

The purpose of this chapter was to highlight the likelihood that operators can identify, admit and enable quality voice over a packet-switched wireless network. A Heuristic Analysis approach was introduced amalgamated with a newly introduced voice specific Admission Control (AC) scheme to ensure that voice packets at varying layers of the Open Systems Interconnection (OSI) stack can be analysed and the right Admission Control (AC) criteria exercised. In the research conducted it was proposed that recognition of voice packets be done via Deep Packet Inspection (DPI) at the eNodeB level. Deep Packet Inspection (DPI) was proposed to be positioned at the eNodeB or WiMAX Base Station (BS) level so as to enable the recognition of Session Initiation Protocol (SIP) voice or Over the Top (OTT) voice at the nearest point in the wireless network to the subscriber.

A newly proposed Heuristic Analysis recognition algorithm was modelled based on the recognition of the voice protocol set-up messaging, where each line of the voice protocol message represented an Object in the Heuristic algorithm and the variance in inter-arrival times between Objects as well as the variance between Objects (voice set-up protocol messages) in receiving spanning tree protocol as well as Point-to-Point Protocol (PPP) Link Control Protocol requests in-between Objects was represented by Axioms.

Both the Session Initiation Protocol (SIP) and Google Talk were used as experimental subjects for the modelling process. Options for the recognition of Session Initiation Protocol (SIP) were presented with the SIP non-numbered sequence approach incorporated with a fixed time duration and Softswitch IP yielding 100% recognition for all sample sets. The idea is that the Heuristic Analysis algorithm be implemented via filtering rules within the Deep Packet Inspection engine such that on recognition an output can be triggered in the form of an attribute value for input into the Admission Control (AC) structure and a Policy and Charging Rules Function (PCRF).

The Session Initiation Protocol (SIP) probability of blocking or non-recognition was combined with the blocking probability of the system based on the voice specific Admission Control (AC) scheme. This was simulated using MATLAB [56] and illustrated that the overall system probability of blocking increases or decreases dependent on the heuristic scheme used. Of greater importance was the Heuristic Analysis recognition and modelling of the Google talk voice pattern because this recognition algorithm combined with an LTE Admission control scheme and policy control enforcement engine has the capability of invoking a Quality of Service (QoS) Bearer for Over the Top (OTT) voice traffic, which shall be illustrated in later chapters.

Deep Packet Inspection (DPI) has a tendency to raise issues surrounding net neutrality and privacy. Net neutrality advocates government regulation of Internet Service Providers such that they are prevented from restricting consumers' access to the Internet in any way. Deep Packet Inspection (DPI) of layers 4 upwards has the possibility of restricting access on a per application basis in violation of net neutrality. A possible solution to this privacy issue can be to split signalling traffic from user plane traffic. Net neutrality should apply as per regulation to user plane traffic but not necessarily to signalling traffic. This will allow service providers to enable advanced services via the manipulation of signalling traffic in layers 4 and above yet not violate privacy issues associated with userplane traffic.

A newly proposed voice specific Admission Control (AC) scheme was presented providing operators with the option to vary capacity requirements between voice and other services based on a traffic analysis approach. A factor *p* was introduced and simulated indicating the ability to vary the number of voice channels based on varying the factor *p*. The significant result that was demonstrated was that a physical resource allocation balance can be ascertained between voice traffic and other traffic in the newly proposed voice specific Admission Control (AC) scheme.

In the following chapter the Bearer or Service Flow selection for voice traffic in packet switched wireless networks is discussed. The Quality of Service (QoS) framework as defined by the LTE and WiMAX standard bodies are set-up on a test bed and rigorously tested in order to identify and isolate the most appropriate Bearer or Service Flow for voice traffic in terms of latency, jitter and packet loss performance. This is done so that the appropriate Quality of Service (QoS) Bearer or Service flow will be implemented or invoked by the proposed Quality of Service (QoS) framework to carry the admitted voice call.

6 Bearer / Service Flow Selection and Testing for Voice

6.1 Introduction

The correct Bearer or Service Flow selection and control thereof is vital to ensure network quality metrics such as latency, jitter and packet loss are sufficiently adhered to for voice. In order for Quality of Service (QoS) to be applied to a voice traffic stream, the relevant Radio Link Control (RLC) and Media Access Control (MAC) configurations need to be invoked via selection of the appropriate QoS Class Identifier (QCI) or Service Flow. The QoS Class Identifier (QCI) or Service Flow makes reference to the Quality of Service (QoS) framework by the relevant standard bodies. The following chapter highlights the testing of the QoS Class Identifiers (QCI) and Service Flows as per Table 6-1 specified for both LTE and WiMAX networks, in order to determine the best applicable QoS Class Identifiers (QCI) or Service Flows to carry voice traffic.

Table 6-1. LTE QoS Class Identifier (QCI) and WiMAX Service Flow [12]

LTE QCI	WiMAX SF
1, 2	UGS
3, 4	ertPS
5, 6	rtPS
7	nrtPS
8	nrtPS
9	BE

The demarcated test bed consisted of WiMAX and LTE equipment installed and configured as a trial network located in Midrand, South Africa. The network set-up for each of the wireless standards as well as the testing procedures undertaken is described in the sections below.

The test results portray finite network quality metrics of the Bearers and Service Flows as well as comparisons with each other considering performance across multiple packet switched network domains. The test results indicate the superior performance of LTE QoS Class Identifier (QCI) 1, 2 and 3 as well as WiMAX Unsolicited Grant Service and enhanced real time Polling Service (ertPS) for the conveyance of voice traffic in terms of latency, jitter and packet loss.

Each of the standards bodies such as the International Telecommunications Union and 3rd Generation Partnership Project have not qualified and specified the mapping of the QoS Class Identifier (QCI) or Service Flow for voice from one network domain to another. Voice network parameter factors *f* are introduced so that integration between network domains for quality voice can be quantified in terms of *f* [42].

6.2 Testing of the LTE Quality of Service (QoS) Framework

The following chapters consist of the testing of each of the LTE QoS Class Identifier (QCI) Bearers. The purpose of this experiment was to determine the most appropriate QoS Class Identifier (QCI) that may be applicable to transport voice traffic. The experimental variables consisted of the various LTE QoS Class Identifiers (QCI) measured against voice network quality metrics; i.e. latency, jitter and packet loss over a mixed traffic LTE Radio Access Network (RAN).

The reason that these voice traffic network quality metrics were used is that they have a direct relationship to objective and subjective voice quality. It has been established that a greater than 0.25% packet loss, 5 millisecond jitter and /or 150 millisecond delay significantly contributed to speech delay, stutter and break up [14]. Although efficient Packet Loss Concealment algorithms have been created such that voice traffic can withstand a greater than 0.25% packet loss, for the purposes of this research Packet Loss Concealment algorithms are not considered, therefore reducing the comparison matrix to the base variables presented above.

6.2.1 Experimental Design - LTE Test System Model

Testing on an actual LTE system was conducted with the LTE system model consisting of the following components:

1. LTE User Equipment (UE).
2. An LTE eNodeB.
3. LTE network management system.
4. A Mobility Management Entity (MME).
5. A Serving Gateway (S-GW).
6. A Packet Data Network Gateway (P-GW).
7. Connectivity to an Ethernet backhaul network.
8. Laptop connected at the LTE User Equipment (UE).
9. Laptop connected to the Ethernet backhaul.

A diagrammatical representation of the network set-up is presented in Figure 6-1.

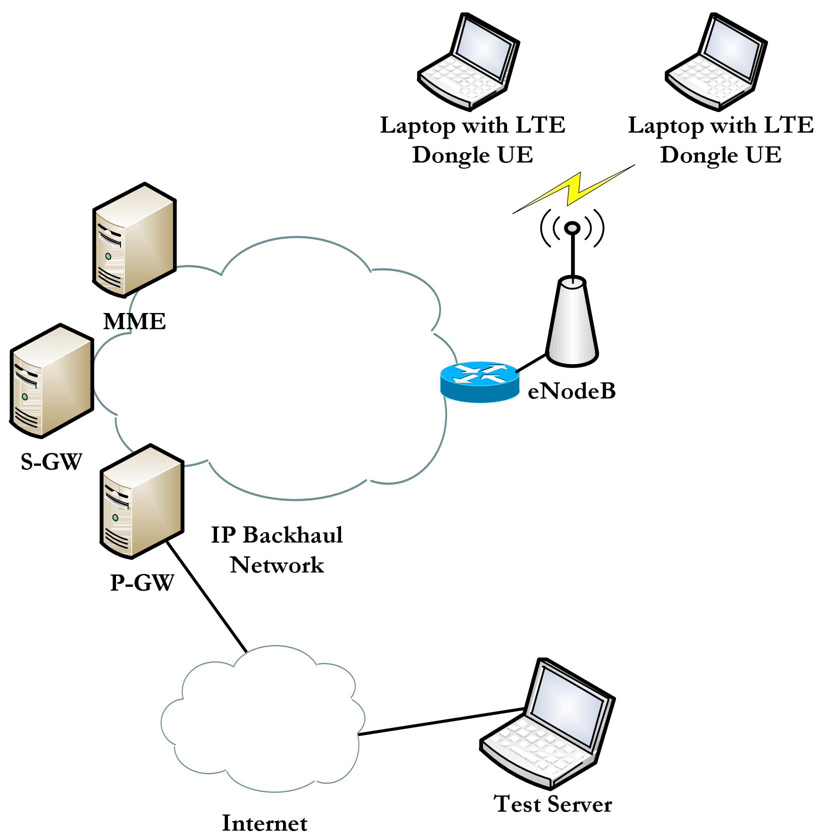

Figure 6-1. LTE Test Model

The system model was configured in the following manner; the eNodeB, Serving Gateway (S-GW), Mobility Management Entity (MME) and Packet Data Network Gateway (P-GW) were connected to each other via a singular Virtual Local Area Network (VLAN) subnet. An Ethernet backhaul network interlinked the Evolved Packet Core (EPC) components with the Access Network via a fibre connection originating from a switch positioned before the Evolved Packet Core (EPC) components and terminating at the eNodeB. Signalling and traffic paths between the Evolved Packet Core (EPC) and eNodeBs were separated, with the S1 and X2 interfaces occupying different Virtual Local Area Network (VLAN) subnets. The eNodeB on the Access Network was located about 2 Km line of sight from the Evolved Packet Core (EPC).

The first test laptop was connected to the eNodeB via an LTE dongle, with the LTE dongle obtaining an IP address and default Bearer connection from the PDN- Gateway (P-GW) via the User Equipment (UE) attach procedure. An external network connection was provided to the Packet Data Network Gateway (P-GW) thereby allowing a direct path from any of the User Equipment (UE) to this external network. The second test laptop was connected to the external connection, thereby creating a route between the LTE connected laptop and the second test laptop.

Due to its availability and recommendations from the electrical engineering fraternity Iperf and Ping test tools were selected to measure latency, jitter and packet loss. Iperf is a network testing tool that can create Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) streams and measure the throughput, jitter and packet loss of a network that these streams traverse. Iperf was developed by the Distributed Applications Support Team (DAST) at the National Laboratory for Applied Network Research (NLANR) [58] and allows the user to set various parameters that can be used for testing a network, or alternately for optimising or tuning a network. Iperf has the ability to create a constant bit rate User Datagram Protocol (UDP) stream that is very similar to voice communication [59].

Ping is a computer network administration utility used to test the reachability of a host on an Internet Protocol (IP) network and to measure the round-trip time for messages sent from the originating host to a destination computer [60]. Ping operates by dispatching Internet Control Message Protocol (ICMP) echo request packets to the target host and waiting for an ICMP response. In the process it measures the time from transmission to reception (round-trip time) and records any packet loss [60]. Further information on Iperf, Ping and portions of its source code is listed in Appendix A.

For the purposes of this experiment; QoS Class Identifier (QCI) profiles were applied to User Equipment (UE) for each of the 9 QoS Class Identifier (QCI) types. On enforcement of the relevant QoS Class Identifier (QCI) Bearer; latency, jitter and packet loss were measured using the Iperf network testing tool [58] and multiple Ping [60] tests. Multiple User Equipment (UE) on the same LTE sector were introduced, injecting spurious traffic on the sector between the test User Equipment (UE), the Evolved Packet Core (EPC) and other User Equipment (UE). The traffic injected was based on different User Equipment (UE) forwarding and receiving traffic whilst being configured with a different QoS Class Identifier (QCI) Bearer from the test User Equipment (UE). This was continuously changed such that the User Equipment (UE) from which the network metrics were being recorded was the only User Equipment (UE) on that QoS Class Identifier (QCI).

The experiments were conducted in accordance with the experimental design. The Iperf [58] and Ping [60] tests were conducted in groups of a minimum of 5 simultaneous tests with the average result of latency, jitter and packet loss recorded as a test-set. The test-set was repeated more than 8 times to obtain an acceptable average per Quality of service Class Identifier. Individual results that exceeded 10% of the average of the grouped test-set resulted in the entire test-set being discarded and the experimental test-set repeated.

The results from the Iperf network testing tool [58] and multiple Ping [60] tests are listed below.

6.2.2 Analysis of LTE Latency Results

As per Figure 6-2 below, the latency results for each of the QoS Class Identifier (QCI) profiles ranged between 27ms and 40ms. This was well below the 150ms one-way or 250ms round trip time thresholds for quality voice. Significant results indicated that QoS

Class Identifier (QCI) 1, 2 and 3 preformed favourably with QoS Class Identifier (QCI) 6 Bearer performing the worst.

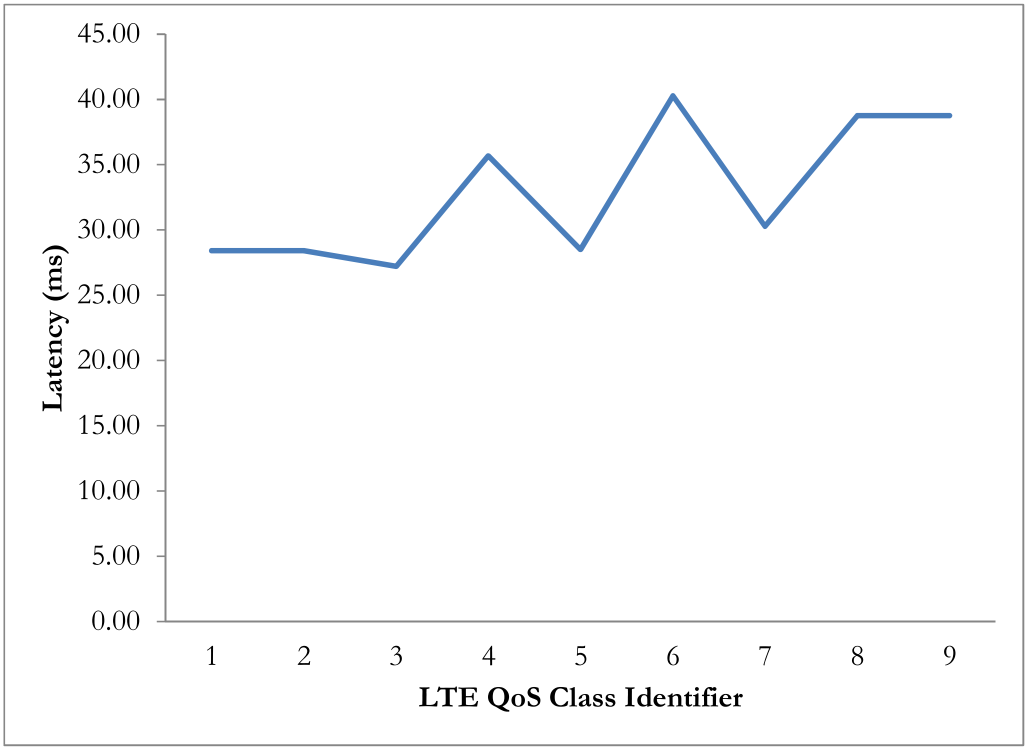

Figure 6-2. Latency on Differing LTE QoS Class Identifier (QCI) types

The results of QoS Class Identifier (QCI) 1, 2 and 3 immediately illustrated the application of appropriate Media Access Control (MAC) and Radio Link Control (RLC) techniques on these QoS Class Identifiers (QCI) to minimise latency and jitter. The results listed above validate the Quality of Service (QoS) framework by the 3rd Generation Partnership Project (3GPP) for LTE. If a trend line is plotted on Figure 6-2 it can be seen that as the QoS Class Identifier (QCI) increases in numerical classification so does the latency result. This confirms that traffic regulation is being enforced on the air interface in accordance with the Quality of Service (QoS) framework.

Although all of the Bearer types performed well below the required latency thresholds, it has to be noted, as is the nature of any wireless technology, such latency figures are considered as significantly high if compared to a fibre link along the same path. This may adversely affect intercontinental voice calls when combined with the latency associated with satellite or cable link transmission. In terms of voice traffic QoS Class Identifier (QCI) 1, 2 and 3 can be used if latency is the only consideration, however jitter and packet loss need to be amalgamated with this result.

6.2.3 Analysis of LTE Jitter Results

The Jitter results obtained for each of the QoS Class Identifier (QCI) types ranged between 4ms and 25ms with QoS Class Identifier (QCI) Bearers 1, 2 and 3 performed the best and the QoS Class Identifier (QCI) 9 and 10 performing the worst. It was also noted that the performance of QoS Class Identifier (QCI) Bearers 1, 2 and 3 were similarly grouped. Figure 6-3 illustrates the jitter results for the differing LTE QoS Class Identifier (QCI) types. Similarly to the LTE Latency results the Jitter results indicate that the appropriate Media Access Control (MAC) and Radio Link Control (RLC) techniques were being applied on the air interface in accordance with the LTE QoS Class Identifier (QCI) framework indicated.

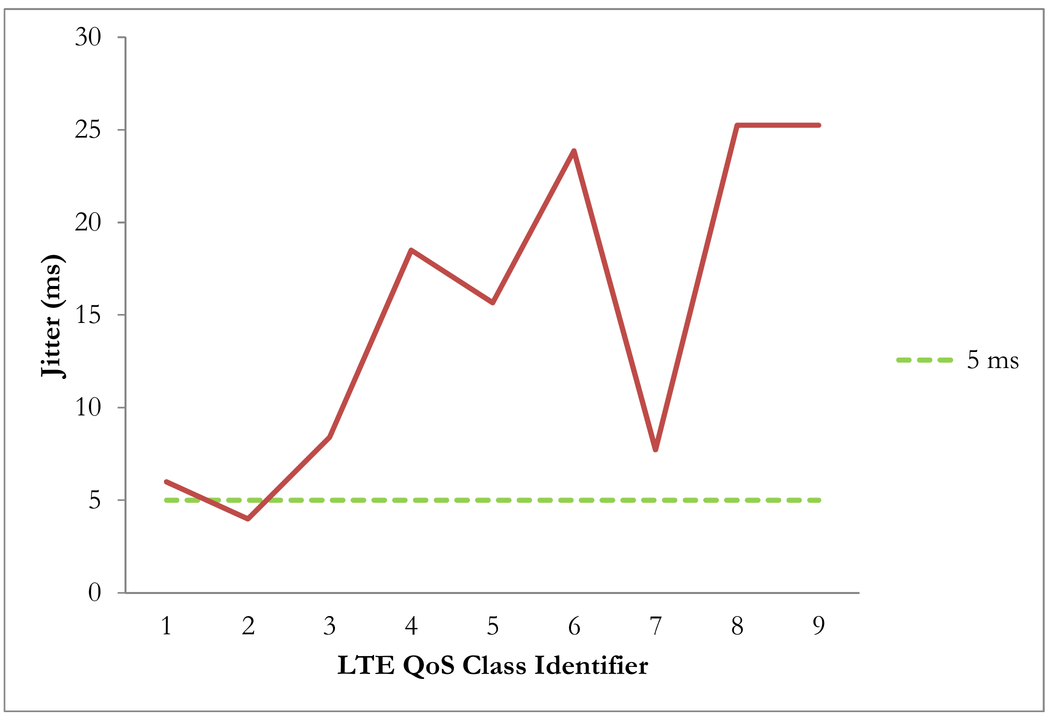

Figure 6-3. Jitter on Differing LTE QoS Class Identifier (QCI) Types

With the purpose of this test being the identification of the most appropriate Bearer to carry voice traffic based on network quality metrics, LTE QoS Class Identifier (QCI) 4 to LTE QoS Class Identifier (QCI) 9 were immediately ruled out as they were well above the 5ms jitter threshold for voice. LTE QoS Class Identifier (QCI) 1, 2 and 3 were deemed the most appropriate for voice. The results show that QoS Class Identifier (QCI) 1 and 3 were on average between 1ms and 4ms above the jitter threshold set at 5ms for quality voice therefore QoS Class Identifier (QCI) 2 became the preferred option due to it demonstrating superior jitter performance compared to the other QoS Class Identifier (QCI) types.

6.2.4 Analysis of LTE Packet Loss Results

The results for packet loss were not illustrated for the reason that across all the Service Flows packet loss was fundamentally zero with a once-off result of a less than 0.03% packet loss. With the packet loss graph plotted it indicated a straight line on the 0ms jitter mark ranging from LTE QoS Class Identifier (QCI) 1 to LTE QoS Class Identifier (QCI) 9.

6.3 Testing of the WiMAX Quality of Service (QoS) Framework

As WiMAX consists of multiple Quality of Service (QoS) Service Flows over the air interface, the best possible way to ensure acceptable voice quality over the WiMAX network is to utilise the best performing Quality of Service (QoS) Service Flow based on tested network quality metrics. It should also be noted that this Quality of Service (QoS) performance needs to be maintained all the way into the core of the WiMAX network [8] [12].

As an example, voice traffic can be carried from the Customer Premises Equipment device in a Virtual Local Area Network (VLAN) subnet to the WiMAX Subscriber Station (SS). This voice Virtual Local Area Network (VLAN) can then be forwarded across the air interface to the WiMAX Base Station (BS) with the appropriate WiMAX Quality of Service (QoS) Service Flow being applied to the Virtual Local Area Network

(VLAN) [11]. Differentiated Service [39] [47] can then be used between the WiMAX Base Station (BS) and the core packet switched network with the voice traffic being mapped onto the appropriate Differentiated Service Code Point (DSCP) at the egress point of the WiMAX Base Station (BS) [12].

The purpose of the following experimental set-up was to determine the most appropriate Service Flow over the WiMAX air interface to convey voice traffic. The experimental variables consisted of the various WiMAX Quality of Service (QoS) Service Flow measured against network quality metrics; i.e. latency, jitter and packet loss over a mixed traffic WiMAX Radio Access Network (RAN). Similarly to the LTE set-up these network quality metrics were used as that they have a direct relationship to objective and subjective voice quality.

6.3.1 Experimental Design – WiMAX Test System Model

As represented in Figure 6-4, the WiMAX test bed consisted of the following components:

1. A WiMAX Subscriber Station (SS).
2. A WiMAX Base Station (BS).
3. WiMAX Access Service Network Gateway (ASN-GW).
4. Access to the WiMAX network management element.
5. Laptop connected at the WiMAX Subscriber Station (SS) or Mobile Station (MS).
6. Laptop connected to the Ethernet backhaul.
7. Connectivity to an Ethernet backhaul network.

The system model was configured in a manner where the Access Service Network Gateway (ASN-GW) was linked to the WiMAX Base Station (BS) via an Ethernet backhaul network. The WiMAX Subscriber Station (SS), WiMAX Base Station (BS), Access Service Network Gateway (ASN-GW) and WiMAX management element were connected to each other via a singular private Virtual Local Area Network (VLAN) subnet. The WiMAX management element provided the ability to log into the WiMAX components and enabled the required configurations. Signalling and traffic paths between the Access Service Network Gateway (ASN-GW) and WiMAX Base Station

(BS) were separated with the user plane traffic occupying a different Virtual Local Area Network (VLAN) subnet. The WiMAX Base Station (BS) was located about 2-3 Km away from the WiMAX Subscriber Station (SS) [11].

The first test laptop was connected to the WiMAX Subscriber Station (SS), with the Subscriber Station (SS) obtaining an IP address from the Access Service Network Gateway (ASN-GW). The second test laptop was connected to the user plane traffic Virtual Local Area Network (VLAN) subnet, thereby creating a direct path between the laptop connected to the WiMAX Subscriber Station (SS) and the second test laptop.

The limitation of the WiMAX test bed set-up was that the Quality of Service (QoS) Service Flows was only manually configurable and not automatically triggered [11] [23]. Figure 6-4 is an illustration of the WiMAX test model configured for the testing procedures.

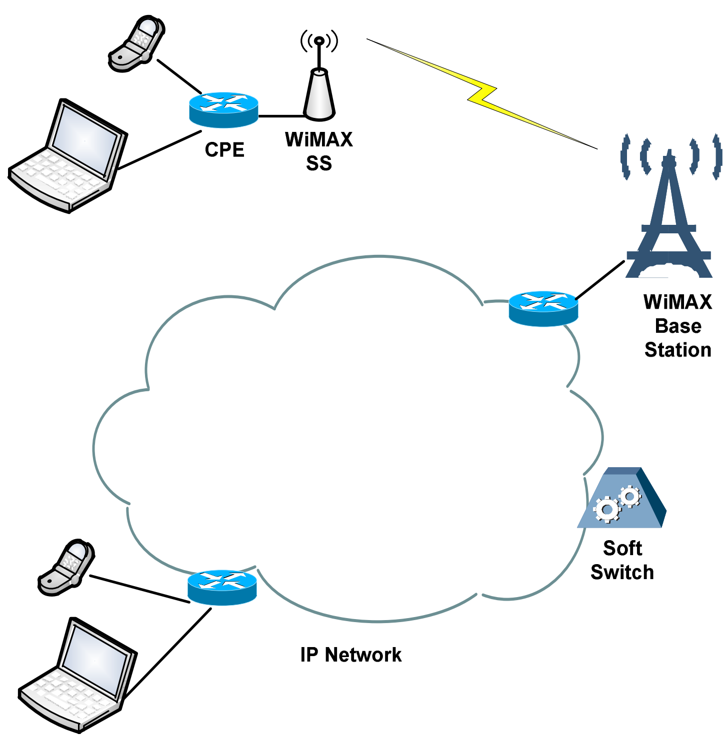

Figure 6-4. Voice over WiMAX Test Model [11]

The Iperf and Ping test tools were selected to measure latency, jitter and packet loss. For the purposes of this experiment; 2 Meg profiles on the uplink were created each with the following Service Flow types applied independently; Unsolicited Grant Service (UGS), enhanced real time Poling Service (ertPS), real time Poling Service (rtPS), non-real time Poling Service (nrtPS) or Best Effort (BE) [11]. On enforcement of the relevant Service Flow profile; latency, jitter and packet loss parameters were measured using the Iperf network testing tool [58] and multiple Ping [60] tests. Multiple User Equipment (UE) were active on the WiMAX sector being used thereby injecting actual user traffic between the test laptops, the Access Service Network Gateway (ASN-GW) and other User Equipment (UE).

The experiments were conducted in accordance with the experimental design. On enforcement of the various Service Flow types the Iperf [58] and Ping [60] tests were conducted in groups of a minimum of 5 simultaneous tests with the average result of latency, jitter and packet loss recorded. The test-sets were repeated more than 8 times to obtain an acceptable average. Individual test results that exceeded 10% of the average of the grouped test-set resulted in the test-set being discarded and the experiment repeated. The results of the Iperf network testing tool [58] and Ping [60] tests are listed below.

6.3.2 Analysis of WiMAX Latency Results

As illustrated in Figure 6-5 below, the latency results for each of the profiles ranged between 28ms and 37ms. This was well below the 150ms one way or 250ms round trip time thresholds for voice traffic. The Unsolicited Grant Service (UGS) Service Flow preformed the best with the Best Effort (BE) Service Flow performing the worst [11]. Although all of the Service Flow types performed well below the required latency thresholds, it has to be noted that the latency figures for WiMAX were in the same region as the latency figures for LTE.

The superior latency performance of the Unsolicited Grant Service (UGS) and the enhanced Real Time Polling Service (ertPS) Service Flows openly demonstrates the application of appropriate Scheduling techniques and Radio Link Control (RLC) on these Service Flows to minimise latency and jitter. The results listed above validate the Quality of Service (QoS) framework by the 3rd Generation Partnership Project 2 (3GPP2) for WiMAX. Similarly to LTE the latency results for WiMAX are satisfactory in comparison to the required latency for quality voice.

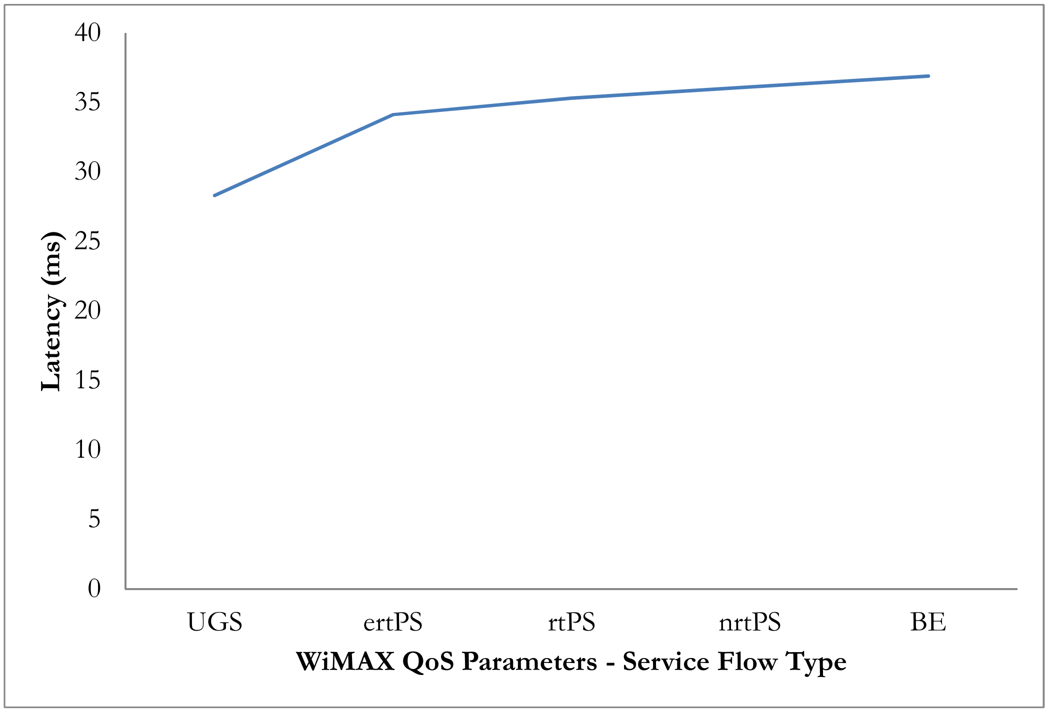

Figure 6-5. Latency on Differing WiMAX Service Flow Types [11]

6.3.3 Analysis of WiMAX Jitter results

As illustrated in Figure 6-6 below the Jitter results obtained for each of the profiles ranged between 3.5ms and 5.4ms with the enhanced real time Polling Service (ertPS) Service Flow preformed the best and the real time Polling Service (rtPS) Service Flow performing the worst. It was also noted that the performance of the real time Polling Service (rtPS), non-real time Polling Service (nrtPS) and Best Effort (BE) Service Flows were similarly grouped confirming that jitter was not prioritised for these Service Flows [11].

When considering the most appropriate Service Flow for voice, the real time Polling Service (rtPS), non-real time Polling Service (nrtPS) and Best Effort (BE) either fell above or were marginally close to the 5ms jitter threshold for voice, hence these Service Flows were ruled out. The Unsolicited Grant Service (UGS) and enhanced real time Polling Service (ertPS) Service Flows were deemed to be the most appropriate for voice, however of importance was that the enhanced real time Polling Service (ertPS) Service

Flow became the preferred option due to this Service Flow demonstrating a 20% better jitter performance that the Unsolicited Grant Service - Service Flow [11].

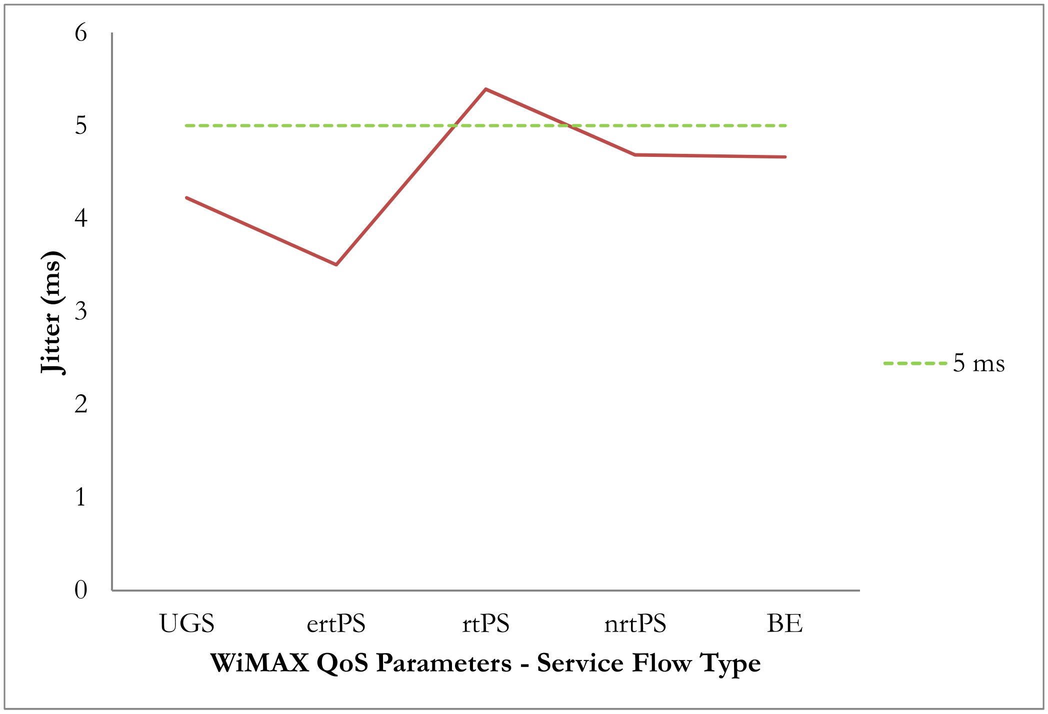

Figure 6-6. Jitter on Differing WiMAX Service Flow Types [11]

6.3.4 Analysis of WiMAX Packet Loss results

The results for packet loss were not illustrated because across all the WiMAX Service Flows packet loss was recorded as 0% [11].

6.3.5 General LTE and WiMAX Testing Comment

Of significance in all the tests listed above was to see the various frameworks on both LTE and WiMAX in full operation as well as to confirm the relationships between the Media Access Control (MAC), Scheduling and Radio Link Control (RLC) options with the numerous QoS Class Identifiers (QCI) and Service Flows as set by the standards bodies. Further analysis is also presented in the chapter on Critical Assessment of Results.

6.4 Voice Quality of Service (QoS) Classification across Domains

The following section considers the effects of the above listed network quality metrics in conjunction with voice traversal across multiple network domains. As stated previously, in order to carry voice of a satisfactory quality a network should consist of sufficient bandwidth to carry the sum of the coded voice and relevant application, transmission and network protocol overheads. The overall voice packet stream shall also consist of less than 0.25% packet loss, a maximum jitter of 5 millisecond and less than 150 millisecond packet delays [14]. This is exemplified in Equation 6-1 as follows;

$$\begin{gathered} p \leq 0.25\% \\ j \leq 5\ ms \\ d \leq 150 ms \end{gathered} \tag{6-1}$$

Where: p = maximum packet loss for an end to end call.

j = maximum packet jitter for an end to end call.

d = maximum packet delay for an end to end call.

If a voice call has to traverse across multiple network domains then the sum of each of the individual network parameters can be represented as follows;

$$p_{domain\ 1} + p_{domain\ 2} + p_{domain\ 3} \leq 0.25\% \tag{6-2}$$

$$j_{domain\ 1} + j_{domain\ 2} + j_{domain\ 3} \leq 5\ ms \tag{6-3}$$

$$d_{domain\ 1} + d_{domain\ 2} + d_{domain\ 3} \leq 150\ ms \tag{6-4}$$

Taking into consideration a 3 domain scenario as depicted in Equations 6-2, 6-3 and 6-4, it can be indicated that any one of the network domains can occupy the entire reserve network parameter. If this occurrence takes place then the summation of the network parameters of all the network domains will result in a degradation of voice quality in the overall voice system, hence an equipment vendor can indicate that their individual equipment is fully compliant to carry voice traffic, however when paired with other network domains the final result may be contradictory to this statement [42].

It is proposed that a voice network parameter factor *f* be introduced where each of the network quality metrics for a vendor designed system Bearer or Service Flows are classified with this factor *f*, where *f* represents the network quality metrics parameter for the vendor system Bearer or Service Flow in relation to the overall required voice system network quality metrics. This concept is listed in Equation 6-5, Equation 6-6 and Equation 6-7 below [42].

$$f_p = \sum_{i=1}^{n} \frac{p_n}{0.25\%} \qquad (6\text{-}5)$$

$$f_j = \sum_{i=1}^{n} \frac{j_n}{5ms} \qquad (6\text{-}6)$$

$$f_d = \sum_{i=1}^{n} \frac{d_n}{150ms} \qquad (6\text{-}7)$$

Where: f_p = sum of the maximum packet loss for a network domain in relation to the overall maximum packet loss.

f_j = sum of the maximum packet jitter for a network domain in relation to the overall maximum packet loss.

f_d = sum of the maximum packet delay for a network domain in relation to the overall maximum packet loss.

n = number of network domains.

The results obtained from the network quality metric tests that have been presented earlier for LTE and WiMAX are now represented as factor *f* as per Equation 6-5, Equation 6-6 and Equation 6-7 [42] in order to ascertain the vendor systems pairing capability with other packet switched networks.

6.4.1 LTE Network Quality Metric Results as a Factor *f*

For quality voice to be maintained across a network system, each of the voice network parameters $f \leq 1$, as per Equation 6-5, Equation 6-6 and Equation 6-7 where 1 represents the overall system network quality metric threshold. Considering LTE QoS Class Identifiers (QCI) 1, 2, 3, 4, 5, 6, 7, 8 and 9 from the results above, the following are the voice network parameter factors *f* for the associated Bearers [42]. As illustrated in Table 6-2, it can be seen that in respect of f_{d1} all QoS Class Identifiers (QCI) perform below the threshold factor of 1. QoS Class Identifier (QCI) 1, 2 and 3 however outperforms all other QoS Class Identifiers by a significant factor hence it is stated that QoS Class Identifier (QCI) 1, 2, and 3 are more robust in terms of being paired with other network domains in a voice system.

Table 6-2. LTE QoS Class Identifier (QCI) to Voice Network Parameter Factor *f* for LTE

QCI	Packet Delay Budget (ms)	Packet Jitter (ms)	f_{d1}	f_{j1}
1	28.400	6.000	0.189	1.200
2	28.400	4.000	0.189	0.800
3	27.200	8.400	0.181	1.680
4	35.667	18.500	0.238	3.700
5	28.500	15.667	0.190	3.133
6	40.286	23.857	0.269	4.771
7	30.286	7.714	0.202	1.543
8	38.750	25.250	0.258	5.050
9	38.750	25.250	0.258	5.050

In respect of f_{j1} it can be seen that QoS Class Identifier (QCI) 1 and 2 perform fairly well, although QoS Class Identifier (QCI) 1 is above the threshold factor of 1 it was noted that even though the average jitter value during testing was 1.2 times the threshold value many of the test sets were below the threshold factor of 1. Therefore it is stated that and both can be paired in line with other network domains however QoS Class Identifier (QCI) 2 is the preferred Bearer option.

6.4.2 WiMAX Network Quality Metric Results as a Factor *f*

As illustrated in Table 6-3, it can be seen that in respect of f_{d1} all Service Flows can be used for voice as they perform below the threshold factor of 1. However Unsolicited Grant Service (UGS) and enhanced real Time Polling Service (ertPS) outperform all other Service Flows, hence is more robust in terms of being paired with other network domains in a voice system.

Table 6-3. WiMAX Service Flow to Voice Network Parameter Factor *f* for the WiMAX Network Domain

SF	Packet Delay Budget (ms)	Packet Jitter (ms)	f_{d1}	f_{j1}
UGS	28.322	4.226	0.189	0.845
ertPS	34.123	3.504	0.227	0.701
rtPS	35.311	5.393	0.235	1.079
nrtPS	36.105	4.687	0.241	0.937
BE	36.896	4.665	0.246	0.933

In respect of f_{j1} it can be seen that Unsolicited Grant Service (UGS) and enhanced real Time Polling Service (ertPS) perform below the threshold factor of 1, and both have sufficient room to be paired in line with other network domains. Therefore it can be stated that both have sufficient room to be paired in line with other network domains, however enhanced real Time Polling Service (ertPS) is preferred as it has a superior jitter tolerance ratio.

6.5 Conclusion

Testing of the LTE QoS Class Identifiers (QCI) and WiMAX Service Flows were highlighted in the chapter above in order to determine the best applicable QoS Class Identifier (QCI) or Service Flow to carry voice traffic. The test platform set-up consisted of actual WiMAX and LTE equipment being installed and configured as a trial network. LTE Bearers and WiMAX Service Flows (SF) with differing Quality of Service (QoS) parameters were tested with each of the Quality of Service (QoS) parameters applied to the Radio Link Control (RLC) and Media Access Control (MAC) as per the Quality of Service (QoS) framework defined by the standards bodies. Latency, jitter and packet loss results were collected during various test sets using the Iperf network testing tool [58] and multiple Ping [60] tests.

The test results for the Quality of Service (QoS) framework illustrated superior network quality metrics performance in terms of latency and jitter when utilising the enhanced real time Poling Service (ertPS) and the Unsolicited Grant service (UGS) as compared to the other Service Flow types with the enhanced real time Poling Service (ertPS) selected as the preferred option. In respect of LTE the QoS Class Identifier (QCI) 1, 2 and 3 were deemed suitable for voice however QoS Class Identifier (QCI) 2 turn out to be the favoured option due to QoS Class Identifier (QCI) 2 demonstrating superior jitter performance that the other QoS Class Identifier (QCI) types.

In respect of voice traffic traversing multiple network domains a new voice network parameter factor *f* was introduced where each of the vendor designed system Bearers or Service Flows are categorised with this factor *f*. *f* is classified as the network quality metric parameter for an individual vendor designed system or network domain in relation to the overall voice system network quality metric threshold. Where: f_p represents the sum of the maximum packet loss for a network domain in relation to the overall maximum packet loss; f_j represents the sum of the maximum packet jitter for a network domain in relation to the overall maximum packet jitter and f_d represents the sum of the maximum packet delay for a network domain in relation to the overall maximum packet delay.

The concept of factor *f* confirmed that LTE Quality Class Identifier (QCI) 2 and WiMAX enhanced real time Polling Service (ertPS) were the best choices to be used for the pairing of voice streams with other packet switched systems. The proposed factor *f* if successfully reported by equipment vendors has the ability to provide operators and system integrators a fair indication of the preferred Quality of Service (QoS) Bearers or Service Flows that can be used for voice, as well as its robustness in terms of being paired with other network domains [42].

The following chapter provides the methodology to invoke Quality of Service (QoS) Bearers or Service Flows for voice traffic in a packet switched wireless network, with the LTE scheme specifically considered. These proposals are significant for the inclusion of Quality of Service (QoS) over packet switched wireless networks for Over the Top (OTT) voice applications. The overall scheme is voice protocol recognition, priority admission control, Quality of Service (QoS) Bearer selection and Quality of Service (QoS) Bearer initiation.

7 Invoking Voice Bearer Control across the Air Interface

7.1 Introduction

In order to implement Quality of Service (QoS) for voice over a packet-switched wireless network the appropriate Bearer or Service Flow set-up procedures need to be invoked such that the appropriate Quality of Service (QoS) metrics can be implemented. Considering the WiMAX architecture, the WiMAX Service Flow selection algorithm is static in nature utilising the principles of a fundamental IP network in order to apply the Service Flow to a packet stream. WiMAX Service Flows are inherently tied to a Virtual Local Area Network (VLAN) applied via the WiMAX network management system to the Subscriber Station (SS) and Base Station (BS); hence the Quality of Service (QoS) Bearer is associated to a specified Virtual Local Area Network (VLAN) at the Subscriber Station (SS). The invoking methodology was not investigated for WiMAX because the WiMAX system did not allow for the Service Flow invoking methodology that is investigated to be applied on the WiMAX system.

In respect of LTE, the LTE Bearer can be invoked by means of a Diameter Gx interface terminating into the Packet Data Network Gateway (P-GW) from a Policy and Charging Rules Function (PCRF) as illustrated in Figure 7-1 [61] or via a Bearer Modification procedure via an Rx interface terminating on the Policy and Charging Rules Function (PCRF). The Diameter protocol was derived from the Radius protocol and is the next generation Authentication, Authorization, and Accounting (AAA) protocol.

In terms of the voice architecture the Application Function represents a proxy server that interacts with the session layer or application layer protocol or service. Usually the application level signalling passes through the Application Function or is terminated in the Application Function. The Application Function extracts session information from the application signalling and provides this to the Policy and Charging Rules Function (PCRF) over the Rx interface [61].

The Policy and Charging Rules Function (PCRF) receives session information over the Rx interface as well as from the Serving Gateway (S-GW) via the Gxc / Gxa interfaces. The Policy and Charging Rules Function (PCRF) takes the available information and creates service-session level policy decisions that are provided to the Policy and Charging Enforcement Function (PCEF) and the Serving Gateway (S-GW).

The Policy and Charging Enforcement Function (PCEF) is part of the Packet Data Network Gateway (P-GW) and is the functional element that encompasses policy enforcement and flow based charging functionalities [61]. The Policy and Charging Enforcement Function (PCEF) enforces policy decisions such as maximum bit rate regulation that are received from the Policy and Charging Rules Function (PCRF) and also provides the Policy and Charging Rules Function (PCRF) with user and access specific information over the Gx interface. The Policy and Charging Enforcement Function (PCEF) also performs measurements of user traffic such as traffic volume and / or user session duration [61].

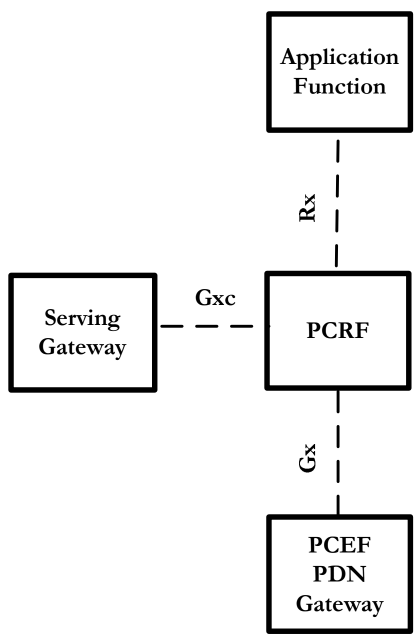

Figure 7-1. LTE Architecture and Diameter Interfaces Associated [61]

The following section proposes procedures for the ordering and initiation of a Quality of Service (QoS) Bearer for voice traffic.

The following standards defined procedures as well as original recommendations are discussed:

1. Static Voice Bearer Set-up where the Bearer is not invoked but rather originated during the User Equipment (UE) Attach procedure.
2. IP Multimedia Subsystem (IMS) initiated voice Bearer set-up.
3. User Equipment (UE) Application initiated voice Bearer set-up.
4. Deep Packet Inspection (DPI) initiated voice Bearer set-up using Heuristic Analysis

Each of the procedures embraces the Policy and Charging Rules Function (PCRF) as central to the initiation of a Quality of Service Bearer for voice traffic. The philosophy used was to manipulate the Rx interface on the Policy and Charging Rules Function (PCRF) to trigger a Guaranteed Bit Rate (GBR) Bearer for voice traffic only. This methodology can then be applied by the Heuristic Analysis algorithm or any other method to initiate a Guaranteed Bit Rate (GBR) Bearer for voice over an LTE network. The initiation of the Quality of Service bearer by the Policy and Charging Rules Function (PCRF) was tested, demonstrating the Bearer initiation mechanism as well as the effectiveness of the Guaranteed Bit Rate (GBR) Bearer on a Real Time Protocol (RTP) voice stream.

7.2 Static Voice Bearer Set-up

In order to statically set up a Quality of Service (QoS) Bearer for voice traffic the attach procedure of the User Equipment (UE) to the network is presented with a view to initiate the required Quality of Service (QoS) Bearer immediately after the attach procedure. The attach procedure is the first procedure that the User Equipment (UE) executes after being switched on and is performed so that the User Equipment (UE) receives services from the LTE network on start-up [61].

The attach procedure enables a default Bearer which is a Non-Guaranteed Bit Rate (Non-GBR) Bearer between the User Equipment (UE) and the Packet Data Network Gateway (P-GW) [17] [61]. In terms of static Bearer provisioning it is proposed that a Guaranteed Bit Rate (GBR) Bearer with QoS Class Identifier (QCI) 2 be established for voice in conjunction with the User Equipment (UE) attached procedure [42]. The relevant procedure combines the 3rd Generation Partnership (3GPP) attach procedure and dedicated Bearer creation procedure so as to form two Bearers during the User Equipment (UE) start-up process i.e. one default Bearer for general traffic and one Guaranteed Bit Rate (GBR) Bearer for voice traffic [33].

The proposed static voice Bearer set-up procedure is described below as well as illustrated in Figure 7-2 [61] [62]:

1. The User Equipment (UE) forwards an Attach Request message to the eNodeB which in turn scans the Mobility Management Entity (MME) identifier that has been transferred in the Radio Resource Control (RRC) layer. If the eNodeB is associated to the identified Mobility Management Entity (MME) it onward forwards the Attach Request to that Mobility Management Entity (MME), else the eNodeB picks a new Mobility Management Entity (MME) and forwards the Attach Request to the new Mobility Management Entity (MME). The User Equipment (UE) in idle State sends out Radio Resource Control (RRC) Connection Establishment to the eNodeB. If the eNodeB accepts the request it sends Radio Resource Control (RRC) Connection Accept to User Equipment (UE). Upon reception of this message the Radio Resource Control (RRC) Connection is established.
2. The User Equipment (UE) to Mobility Management Entity (MME) authentication and security procedures take place where the Mobility Management Entity (MME) identity is retrieved in conjunction with the Radio Resource Control (RRC) Connection Establishment.
3. If it has been established that the Mobility Management Entity (MME) has changed, the new Mobility Management Entity (MME) informs the HSS that the User Equipment (UE) has moved. The Home Subscriber Server (HSS) stores the new Mobility Management Entity (MME) address and instructs the old Mobility Management Entity (MME) to cancel the User Equipment (UE) context.

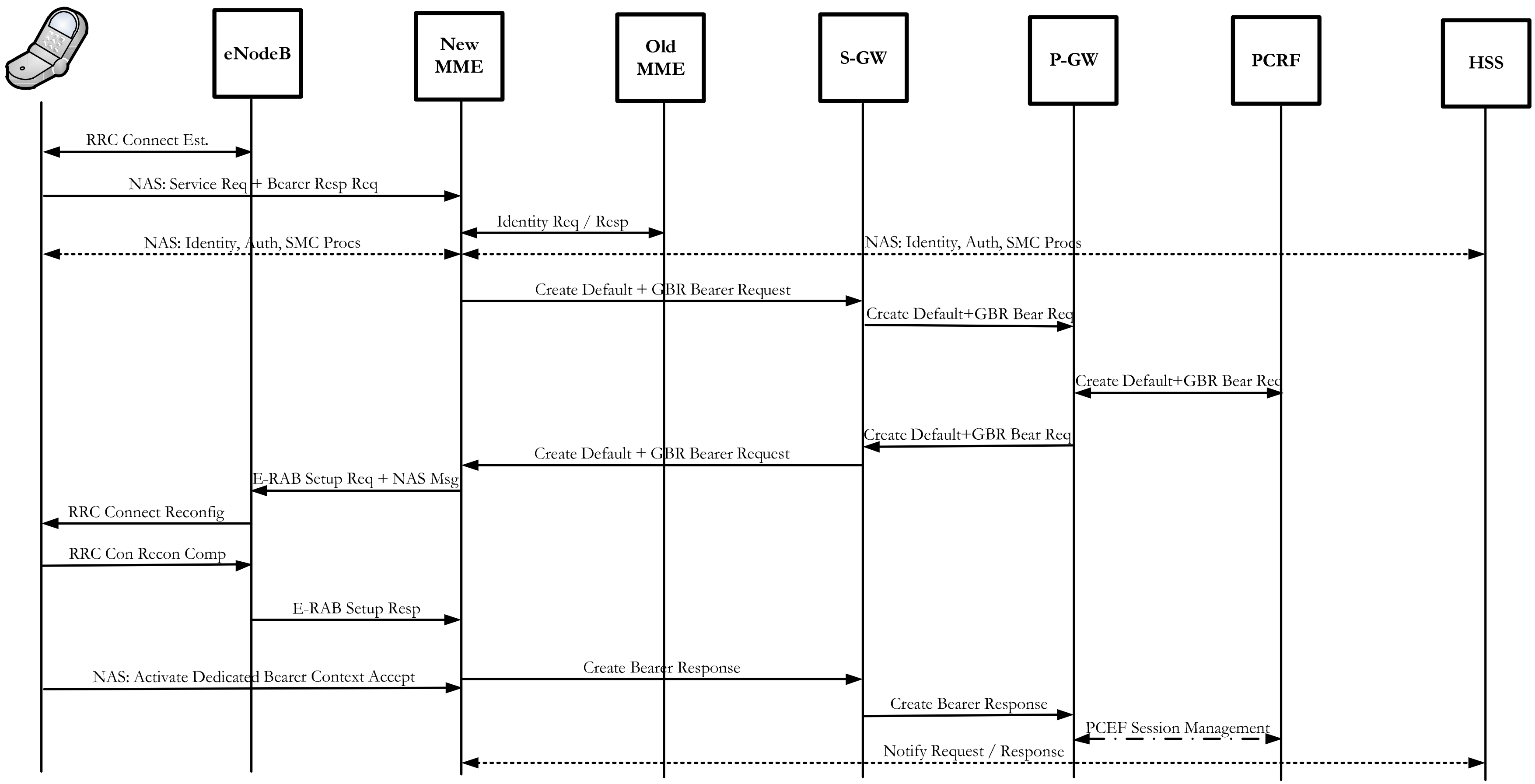

Figure 7-2. Static Voice Bearer Set-up Procedure [61]

4. On receipt of the Radio Resource Control (RRC) Connection Complete, along with Non-Access-Stratum (NAS) message the eNodeB extracts the Non-Access-Stratum (NAS) message, places it in S1AP message and passes it to Mobility Management Entity (MME). The Mobility Management Entity (MME) reads this Non-Access-Stratum (NAS) message and understands that User Equipment (UE) needs a default Bearer, Guaranteed Bit Rate (GBR) Bearer and the associated IP addresses. Mobility Management Entity (MME) creates a GPRS Tunnelling Protocol (GTP) message create session request, Bearer resource command and forwards this to the Serving Gateway (S-GW). Bearer resource command typically consists of Quality of Service (QoS) values.
5. The Serving Gateway (S-GW) contacts the Policy and Charging Rules Function (PCRF) and pulls out the Quality of Service (QoS) values for the Traffic Flow Templates (TFT). The Quality of Service (QoS) values can be the one that User Equipment (UE) has requested for or they can be different. The default Bearer and Guaranteed Bit Rate (GBR) Bearer is authorized by the Policy and Charging Rules Function (PCRF) and established between Serving Gateway (S-GW) and Packet Data Network Gateway (P-GW). The Policy and Charging Rules Function (PCRF) forward the IP 5 tuple including source IP as allocated by the Packet Data Network Gateway (P-GW) during the attach procedure as well as the network Quality of Service (QoS) required.
6. The Serving Gateway (S-GW) processes the Create Bearer Request and forwards to the Mobility Management Entity (MME). The Mobility Management Entity (MME) sends the E-UTRAN Radio Access Bearer (E-RAB) set-up request to the eNodeB for Bearer allocation between the eNodeB and the Serving Gateway (S-GW) via the NAS Activate Dedicated EPS Bearer Context Request to the User Equipment (UE).
7. The eNodeB allocates the resources for the Radio Bearers using a Radio Resource Control Connection Reconfig Request message to the User Equipment (UE), after which the User Equipment (UE) establishes the Radio Bearers and responds back with a Radio Resource Control Connection Reconfiguration Complete message to the eNodeB. The default Bearer and Guaranteed Bit Rate

(GBR) Bearer is established over the radio interface and the attach Accept is sent to the User Equipment (UE).

8. Mobility Management Entity (MME) informs the Serving Gateway (S-GW) of the eNodeB Tunnel Endpoint Identifiers (TEID) which completes the set-up of the default and Guaranteed Bit Rate (GBR) Bearers.
9. The Mobility Management Entity (MME) will send a notification of the new Packet Data Network Gateway (P-GW) identity to the Home Subscriber Server (HSS).

7.3 IP Multimedia Subsystem (IMS) Initiated Voice Bearer Set-up

A subscriber is required to perform IP Multimedia Subsystem (IMS) registration prior to IP Multimedia Subsystem (IMS) services being obtained. Registration can be initiated once the User Equipment (UE) has established the default Bearer or an Evolved Packet System (EPS) Bearer for the IP Multimedia Subsystem (IMS) signalling to take place and after Proxy Call Session Control Function (P-CSCF) discovery has taken place. The Proxy Call Session Control Function (P-CSCF) discovery is used by the User Equipment (UE) to obtain the Proxy Call Session Control Function (P-CSCF) address by using the Evolved Packet System (EPS) Bearer context activation procedure, Dynamic Host Configuration Protocol (DHCP) procedures or from preconfigured Proxy Call Session Control Function (P-CSCF) data on the Subscriber Identity Module (SIM) card [29] [38]. Registration of the User Equipment (UE) ensures that the User Equipment (UE) IP address can be connected to the user's public identity as known to other Session Initiation Protocol (SIP) subscribers [38].

The IP Multimedia Subsystem (IMS) voice call initiation is described as follow and illustrated in Figure 7-3 [61]:

1. Using Session Initiation Protocol (SIP) and Session Description Protocol (SDP) the subscriber initiates an IP Multimedia Subsystem (IMS) voice call and performs end-to-end application session signalling via the Proxy Call Session Control Function (P-CSCF).

2. Founded on service description information contained in the Session Description Protocol (SDP) application signalling, the Proxy Call Session Control Function (P-CSCF) provides the Policy and Charging Rules Function (PCRF) with the service-related information over the Rx interface. The session information is mapped from a Session Description Protocol (SDP) at the Proxy Call Session Control Function (P-CSCF) into information elements in the Rx messages onward to the Policy and Charging Rules Function (PCRF). The information elements include Quality of Service (QoS) information such as type of service, bit rate requirements as well as the IP 5-tuple that identify the IP Service Flow.
3. The Policy and Charging Rules Function (PCRF) utilises the information elements, session information, service policies and other data in order to build policy decisions that are formulated as Policy Charging and Control (PCC) rules.
4. The Policy Charging and Control (PCC) rules are forwarded by the Policy and Charging Rules Function (PCRF) to the Policy Control Enforcement Function (PCEF) contained at the Packet Data Network Gateway (P-GW). The Policy Control Enforcement Function (PCEF) will enforce the policy decision in accordance with the established Policy Charging and Control (PCC) rule.
5. The Policy Control Enforcement Function (PCEF) installs the Policy Charging and Control (PCC) rules and performs Bearer binding ensuring that the traffic for this service receives the appropriate Quality of Service (QoS) metrics. In some instances this may trigger the establishment of a new Bearer or in other instances the modification of an existing Bearer.
6. The Packet Data Network Gateway (P-GW) processes the Create Bearer Request and forwards to the Serving Gateway (S-GW). The Serving Gateway (S-GW) processes the Create Bearer Request and forwards to the Mobility Management Entity (MME). The Mobility Management Entity (MME) sends the EUTRAN Radio Access Bearer (E-RAB) set-up request to the eNodeB for Bearer allocation between the eNodeB and the Packet Data Network Gateway (P-GW) via the Non-Access Stratum (NAS) Activate Dedicated EPS Bearer Context Request to the User Equipment (UE).

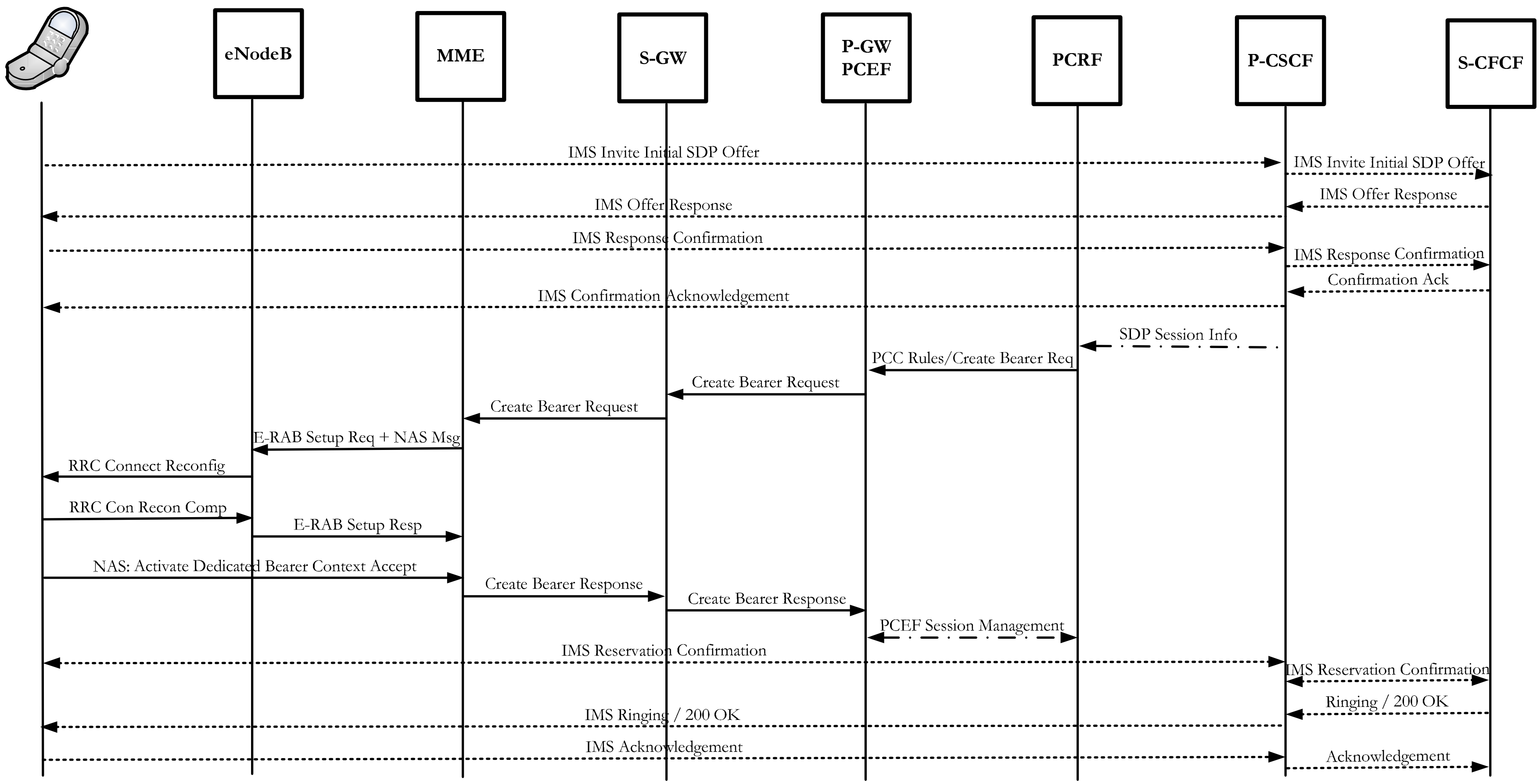

Figure 7-3. IP Multimedia Subsystem (IMS) Initiated Voice Bearer Set-up [38] [61]

7. The eNodeB allocates the resources for the Radio Bearers using a Radio Resource Control (RRC) Connection Reconfig Request message to the User Equipment (UE), after which the User Equipment (UE) establishes the Radio Bearers and responds back with a Radio Resource Control (RRC) Connection Reconfiguration Complete message to the eNodeB. The Guaranteed Bit Rate (GBR) Bearer is established over the radio interface and the attach Accept is sent to the User Equipment (UE).
8. After the reservation confirmation, IP Multimedia Subsystem (IMS) ringing, 200 OK and acknowledgement the media for the session is transported across the network. The Policy Control Enforcement Function (PCEF) performs service data flow detection to detect the IP flow for this service and this IP flow is transported over the allocated Bearer.

7.4 User Equipment (UE) Application Initiated Voice Bearer Set-up

With consideration to User Equipment (UE) Application initiated Quality of Service (QoS) Bearer set-up, the User Equipment (UE) triggers the commencement of the Guaranteed Bit Rate (GBR) Bearer for the voice application.

The User Equipment (UE) Application voice Bearer set-up is described as follows and illustrated in Figure 7-4 [61] [62]:

1. The subscriber launches the relevant application from the User Equipment (UE) which in turn executes a connection to the Application Server via the default Bearer.
2. The User Equipment (UE) application registers with the Application Server and forwards voice service description information contained in the application signalling. The Application Server provides the Policy and Charging Rules Function (PCRF) with the voice service information over the Rx interface. The Policy and Charging Rules Function (PCRF) waits until the User Equipment

(UE) prompts the Policy and Charging Rules Function (PCRF) for a download of rules into the Policy Control Enforcement Function (PCEF).

3. The Application in turn prompts the User Equipment (UE) to request the necessary Quality of Service (QoS) resources including Quality of Service (QoS) metrics such as the Quality of Service (QoS) class and packet filters. The User Equipment (UE) sends the Service Request to the Mobility Management Entity (MME) (UE-requested modify Bearer request) and requests a Guaranteed Bit Rate (GBR) dedicated Bearer resource. The Mobility Management Entity (MME) forward the modify Bearer request to the Serving Gateway (S-GW).
4. The Serving Gateway (S-GW) initiates interaction with the Policy and Charging Rules Function (PCRF) after receiving the request sent by the User Equipment (UE), instead of working via the Packet Data Network Gateway (P-GW). The Policy and Charging Rules Function (PCRF) builds policy decisions formulated as Policy Charging and Control (PCC) rules and sends theses Quality of Service (QoS) rules to the Serving Gateway (S-GW). The Policy and Charging Rules Function (PCRF) also sends the corresponding Policy Charging and Control (PCC) rules to the Policy Control Enforcement Function (PCEF).
5. The Serving Gateway (S-GW) installs the Quality of Service (QoS) rules and performs Bearer binding to ensure that the traffic for the voice service will receive appropriate Quality of Service (QoS) treatment. This will result in the establishment of a new Bearer. The Policy Control Enforcement Function (PCEF) also installs the Policy Charging and Control (PCC) rules for the Policy Control Enforcement Function (PCEF) to perform gating, bit rate enforcement and service level charging for the traffic flow.
6. The Serving Gateway (S-GW) processes the Create Bearer Request and forwards to the Mobility Management Entity (MME). The Mobility Management Entity (MME) sends the E-UTRAN Radio Access Bearer (E-RAB) set-up request to the eNodeB for Bearer allocation between the eNodeB and the Packet Data Network Gateway (P-GW) via the NAS Activate Dedicated EPS Bearer Context Request to the User Equipment (UE).

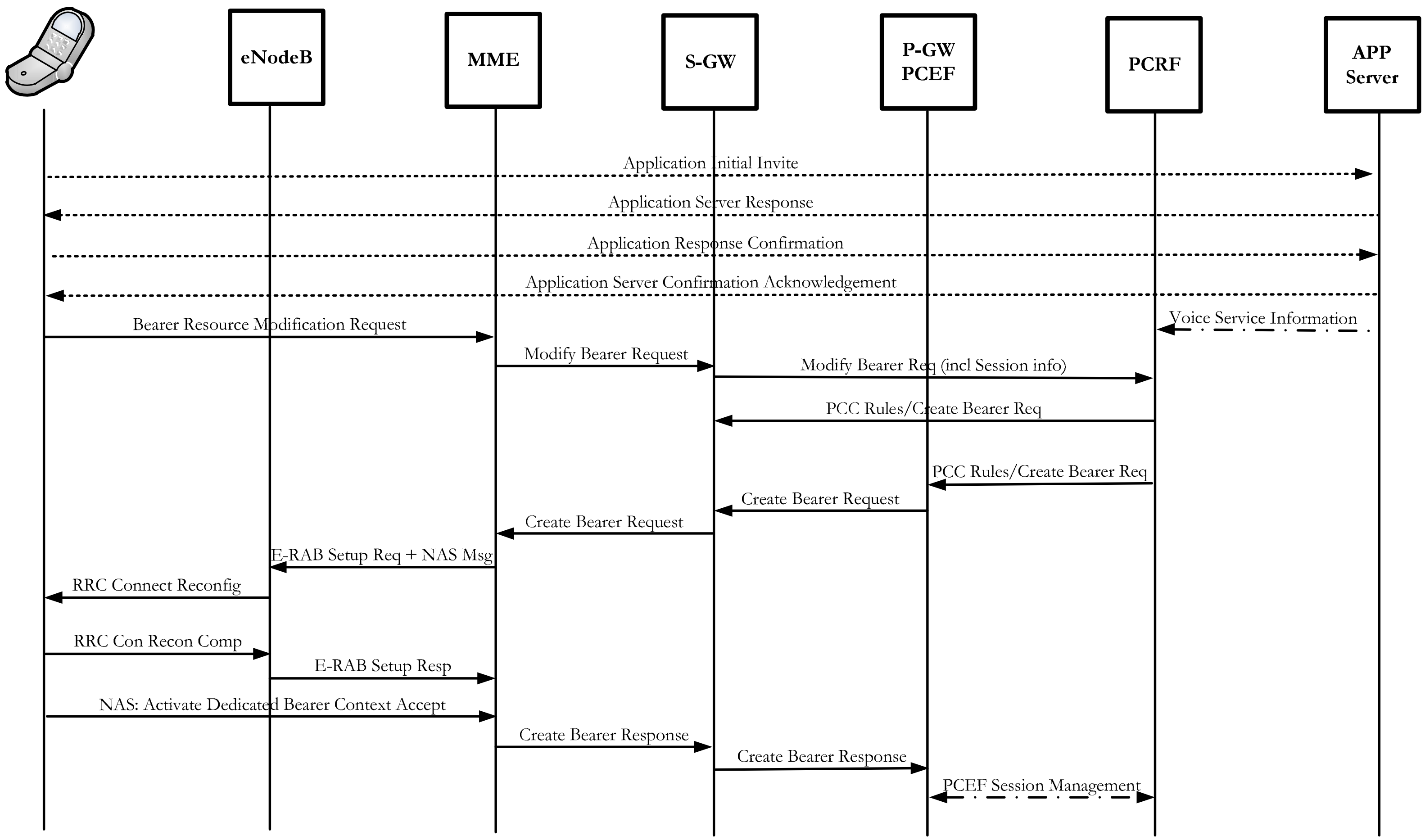

Figure 7-4. Voice Application Initiated Voice Bearer Set-up [61] [62]

7. The eNodeB allocates the resources for the Radio Bearers using a Radio Resource Control (RRC) Connection Reconfig Request message to the User Equipment (UE), after which the User Equipment (UE) establishes the Radio Bearers and responds back with a Radio Resource Control (RRC) Connection Reconfiguration Complete message to the eNodeB. The Guaranteed Bit Rate (GBR) Bearer is established over the radio interface and the attach Accept is sent to the User Equipment (UE).
8. The voice call is now being transported across the network. The Serving Gateway (S-GW) and Policy Control Enforcement Function (PCEF) perform service data flow detection to detect the IP flow for the voice service. The User Equipment (UE) utilises uplink packet filters to determine which Bearer shall carry uplink traffic whereas the Serving Gateway (S-GW) forwards downlink traffic over the designated Bearer.

7.5 Deep Packet Inspection (DPI) Initiated Voice Bearer Set-up using Heuristic Analysis

It is proposed that Heuristic Analysis be used to trigger a Guaranteed Bit Rate (GBR) Bearer to carry voice. In order to apply Heuristic Analysis in the recognition of voice over a packet switched wireless network, a Deep Packet Inspection (DPI) engine is required within the framework of the wireless architecture. As previously indicated it is proposed that a Deep Packet Inspection (DPI) engine be located at either the eNodeB or Serving Gateway (S-GW) nodes of the wireless architecture. These positions are advantageous because all user plane traffic flow through these points as opposed to the Mobility Management Entity (MME), the eNodeB is located at the edge of the network and can hence trigger the Guaranteed Bit Rate (GBR) Bearer earlier in the recognition process as opposed to the Serving Gateway (S-GW) which is more centrally located.

7.5.1 Deep Packet Inspection (DPI) Engine Located at Serving Gateway (S-GW)

The voice Bearer set-up using Heuristic Analysis with the Deep Packet Inspection (DPI) engine located at the Serving Gateway (S-GW) is described as follows and illustrated in Figure 7-5 [61] [62]:

1. All traffic flows through the default Bearer to the Serving Gateway (S-GW) as per the existing attach and Bearer set-up procedure.
2. The User Equipment (UE) Application (SIP / Google Talk / Skype) registers with the Application Server and forwards voice service description information contained in the application signalling. The Application Server provides the Policy and Charging Rules Function (PCRF) with the voice service information over the Rx interface. This step is done once or altogether skipped if the Policy Charging and Control (PCC) rules are statically set. The Policy and Charging Rules Function (PCRF) waits until the User Equipment (UE) prompts the Policy and Charging Rules Function (PCRF) for a download of rules into the Policy Control Enforcement Function (PCEF).
3. The Deep Packet Inspection (DPI) engine located at the Serving Gateway (S-GW) recognises the Application (SIP / Google Talk / Skype) and requests for the necessary Quality of Service (QoS) resources such as the Quality of Service (QoS) class and packet filters.
4. The Serving Gateway (S-GW) interacts with the Policy and Charging Rules Function (PCRF) after receiving the request, instead of working via the Packet Data Network Gateway (P-GW) pulls down the Policy Charging and Control (PCC) rules from the Policy and Charging Rules Function (PCRF). The Policy and Charging Rules Function (PCRF) builds policy decisions formulated as PCC rules and sends these Quality of Service (QoS) rules to the Serving Gateway (S-GW). The Policy and Charging Rules Function (PCRF) also sends the corresponding Policy Charging and Control (PCC) rules to the Policy Control Enforcement Function (PCEF).

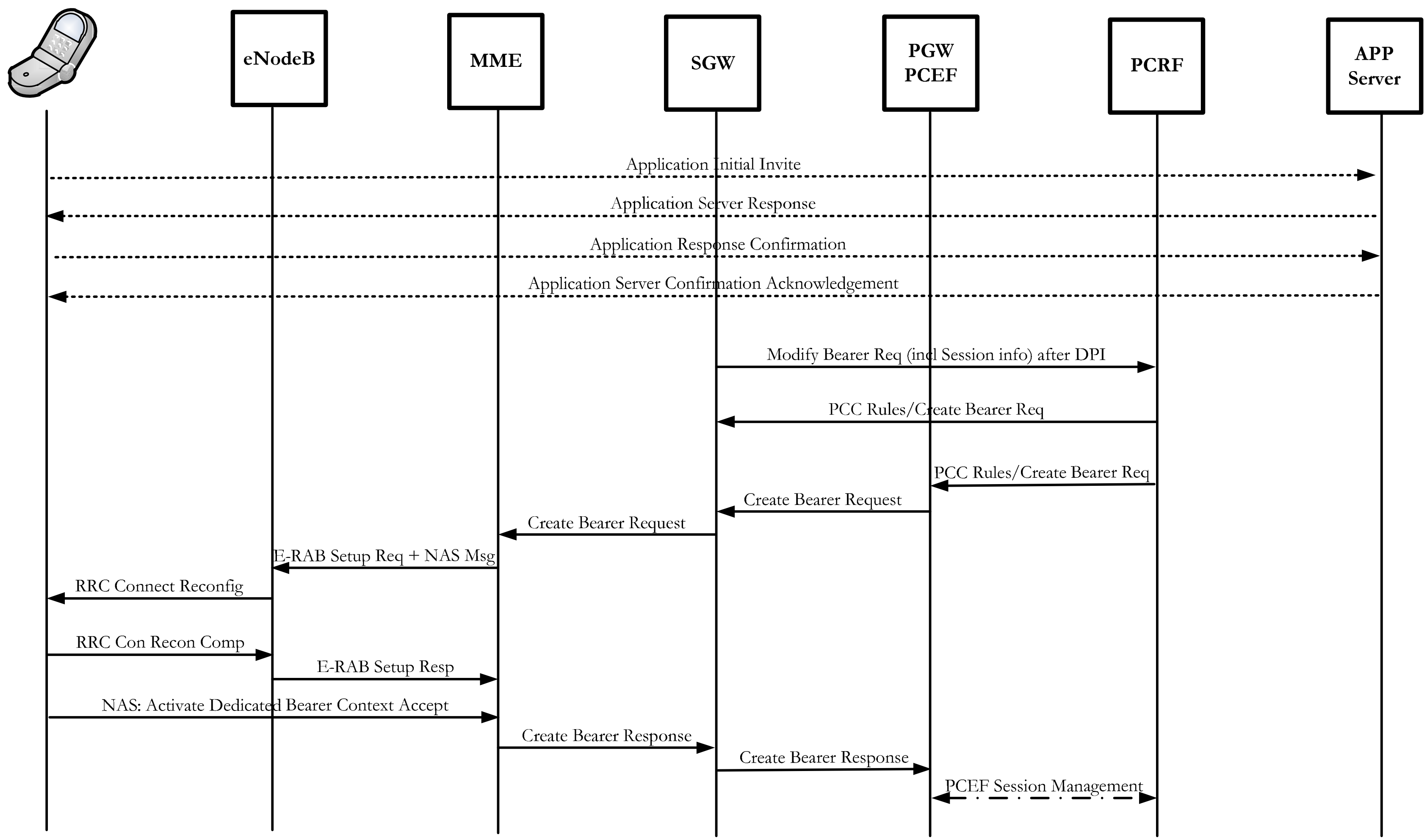

Figure 7-5. Serving Gateway (S-GW) DPI Initiated Voice Bearer Set-up

5. The Serving Gateway (S-GW) installs the Quality of Service (QoS) rules and performs Bearer binding to ensure that the traffic for the voice service will receive appropriate Quality of Service (QoS) treatment. This will result in the establishment of a new Bearer. The Policy Control Enforcement Function (PCEF) also installs the Policy Charging and Control (PCC) rules for the Policy Control Enforcement Function (PCEF) to perform gating, bit rate enforcement and service level charging for the traffic flow.
6. The Serving Gateway (S-GW) processes the Create Bearer Request and forwards to the Mobility Management Entity (MME). The Mobility Management Entity (MME) sends the E-UTRAN Radio Access Bearer (E-RAB) set-up request to the eNodeB for Bearer allocation between the eNodeB and the Packet Data Network Gateway (P-GW) via the NAS Activate Dedicated EPS Bearer Context Request to the User Equipment (UE).
7. The eNodeB allocates the resources for the Radio Bearers using a Radio Resource Control (RRC) Connection Reconfig Request message to the User Equipment (UE), after which the User Equipment (UE) establishes the Radio Bearers and responds back with a Radio Resource Control (RRC) Connection Reconfiguration Complete message to the eNodeB. The Guaranteed Bit Rate (GBR) Bearer is established over the radio interface and the attach Accept is sent to the User Equipment (UE).
8. The voice call is now being transported across the network. The Serving Gateway (S-GW) and Policy Control Enforcement Function (PCEF) perform service data flow detection to detect the IP flow for the voice service. The User Equipment (UE) utilises uplink packet filters to determine which Bearer shall carry uplink traffic whereas the Serving Gateway (S-GW) forwards downlink traffic over the designated Bearer.

7.5.2 Deep Packet Inspection (DPI) Engine Located at eNodeB

The proposed voice Bearer set-up using Heuristic Analysis with the Deep Packet Inspection (DPI) engine located at the eNodeB is described as follows [61] [62]:

1. All traffic on the Access Network, i.e. localised or edge traffic from the User Equipment (UE) flows through the default Bearer to the eNodeB as per the existing LTE User Equipment (UE) attach and Bearer set-up procedure.
2. The User Equipment (UE) Application (SIP / Google Talk / Skype) registers with the Application Server and forwards voice service description information contained in the application signalling. The Application Server provides the Policy and Charging Rules Function (PCRF) with the voice service information over the Rx interface. This step is done once or altogether skipped if the Policy Charging and Control (PCC) rules are statically set. The Policy and Charging Rules Function (PCRF) waits until the User Equipment (UE) prompts the Policy and Charging Rules Function (PCRF) for a download of rules into the Policy Control Enforcement Function (PCEF).
3. The Deep Packet Inspection (DPI) engine located at the eNodeB recognises the Application (SIP / Google Talk / Skype) via Heuristic Analysis and requests for the necessary Quality of Service (QoS) resources such as the Quality of Service (QoS) class and packet filters. The eNodeB sends a Modify Bearer Request to the Mobility Management Entity (MME) and requests a Guaranteed Bit Rate (GBR) dedicated Bearer resource. The Mobility Management Entity (MME) then forward the modify Bearer request to the Serving Gateway (S-GW).
4. The Serving Gateway (S-GW) initiates interaction with the Policy and Charging Rules Function (PCRF) after receiving the request, instead of working via the Packet Data Network Gateway (P-GW), pulls down the Policy Charging and Control (PCC) rules from the Policy and Charging Rules Function (PCRF). The Policy and Charging Rules Function (PCRF) builds policy decisions formulated as Policy Charging and Control (PCC) rules and sends these Quality of Service (QoS) rules to the Serving Gateway (S-GW). The Policy and Charging Rules Function (PCRF) also sends the corresponding Policy Charging and Control (PCC) rules to the Policy Control Enforcement Function (PCEF).

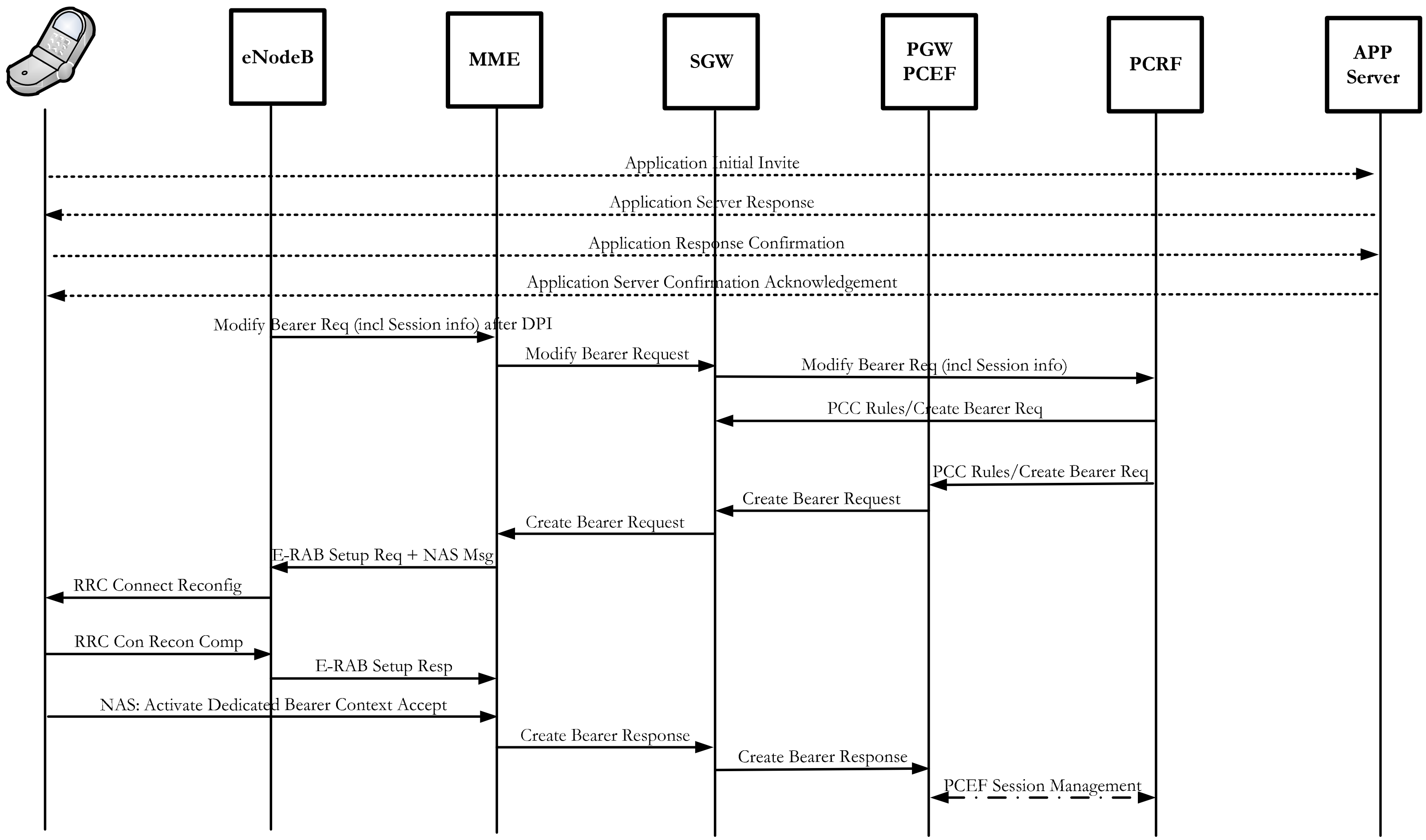

Figure 7-6. eNodeB DPI Initiated Voice Bearer Set-up

5. The Serving Gateway (S-GW) installs the Quality of Service (QoS) rules and performs Bearer binding to ensure that the traffic for the voice service will receive appropriate Quality of Service (QoS) treatment. This will result in the establishment of a new Bearer. The Policy Control Enforcement Function (PCEF) also installs the Policy Charging and Control (PCC) rules for the Policy Control Enforcement Function (PCEF) to perform gating, bit rate enforcement and service level charging for the traffic flow.
6. The Serving Gateway (S-GW) processes the Create Bearer Request and forwards it to the Mobility Management Entity (MME). The Mobility Management Entity (MME) sends the E-UTRAN Radio Access Bearer (E-RAB) set-up request to the eNodeB for Bearer allocation between the eNodeB and the Packet Data Network (P-GW) via the NAS Activate Dedicated EPS Bearer Context Request to the User Equipment (UE).
7. The eNodeB allocates the resources for the Radio Bearers using a Radio Resource Control (RRC) Connection Reconfig Request message to the User Equipment (UE), after which the User Equipment (UE) establishes the Radio Bearers and responds back with a Radio resource Control (RRC) Connection Reconfiguration Complete message to the eNodeB. The Guaranteed Bit Rate (GBR) Bearer is established over the radio interface and the attach Accept is sent to the User Equipment (UE).
8. The voice call is now being transported across the network. The Serving Gateway (S-GW) and Policy Control Enforcement Function (PCEF) perform service data flow detection to detect the IP flow for the voice service. The User Equipment (UE) utilises uplink packet filters to determine which Bearer shall carry uplink traffic whereas the Serving Gateway (S-GW) forwards downlink traffic over the designated Bearer.

7.6 Initiating a Quality Voice Bearer over a Packet Switched Wireless Network

The work presented thus far provides solutions to enable quality control for voice traffic on the admission control component, Bearer selection component and the bearer initiation or invoking component. In section 7.6.1, an experiment was set-up to demonstrate the ability to successfully invoke a Quality of Service (QoS) Bearer for voice on an LTE network. The philosophy used was to manipulate the Rx interface on the Policy and Charging Rules Function (PCRF) to trigger a Guaranteed Bit Rate (GBR) Bearer for voice traffic only. This methodology can then be applied by the Heuristic Analysis algorithm or any other manner to initiate a Guaranteed Bit Rate (GBR) Bearer for voice over an LTE network.

7.6.1 Testing of LTE QoS Bearer Initiation

The purpose of this experiment was to demonstrate that Quality of Service (QoS) Bearer initiation is possible. It can be integrated with the Heuristic Analysis voice Admission Control (AC) scheme and find a practical solution to demonstrate quality voice over an LTE network without the deployment of a compound solution such as the addition of IP Multimedia Services (IMS).

The experimental variables consist of the Policy and Charging Rules Function (PCRF) Rx interface that any User Equipment or Application Server can message with an attribute value, LTE Bearers with QoS Class Identifier (QCI) as well as Traffic Flow Templates (TFT) that are associated with the LTE Bearers.

The following defined LTE voice Bearer initiation procedures cannot be successfully implemented unless an LTE Guaranteed Bit Rate (GBR) Bearer can be successfully invoked.

1. Static Voice Bearer Set-up where the Bearer is not invoked but rather originated during the User Equipment (UE) Attach procedure.
2. IP Multimedia Subsystem (IMS) initiated voice Bearer set-up.
3. User Equipment (UE) Application initiated voice Bearer set-up.
4. Deep Packet Inspection (DPI) initiated voice Bearer set-up using Heuristic Analysis.

It can be said that these procedures can be classified as either network initiated or user initiated Guaranteed Bit Rate (GBR) Bearer procedures, however all procedures interact with the Policy and Charging Rules Function (PCRF) after receiving a request for a voice

Bearer. This request creates a pull down of Policy Charging and Control (PCC) rules from the Policy and Charging Rules Function (PCRF). The Policy and Charging Rules Function (PCRF) sends the Policy Charging and Control (PCC) rules to the Policy Control Enforcement Function (PCEF) and / or the Serving Gateway (S-GW) for appropriate enforcement.

As a recap when a Guaranteed Bit Rate (GBR) Bearer is initiated that specific Bearer is associated with a Traffic Flow Template (TFT) at the eNodeB and the PDN-Gateway (P-GW). This Traffic Flow Template (TFT) consists of a source IP address, destination IP address, source port, destination port and possible protocol filter applied at the eNodeB and the PDN-Gateway (P-GW). Inherently specified traffic is associated to a Traffic Flow Template (TFT), which is associated with a Quality of Service Class Identifier (QCI), which is then associated with the applicable Radio Link Control (RLC) and Media Access Control attributes that overall manage Quality of Service (QoS) over an LTE Bearer.

7.6.2 Experimental Design – LTE Quality of Service Bearer Initiation

Testing of the LTE Bearer initiation procedure was conducted on an actual LTE system with a system model consisting of the following components:

1. LTE User Equipment (UE).
2. An LTE eNodeB.
3. LTE network management system.
4. A Mobility Management Entity (MME).
5. A Serving Gateway (S-GW).
6. A Packet Data Network Gateway (P-GW).
7. Policy Control Enforcement Function (PCEF) which is part of the Packet Data Network Gateway (P-GW).
8. Policy and Charging Rules Function (PCRF).
9. Connectivity to an Ethernet backhaul network.
10. Laptop connected at the LTE User Equipment (UE).
11. Session Initiation Protocol (SIP) User Agent.
12. Softswitch.

An illustration of the network set-up is presented in Figure 7-7.

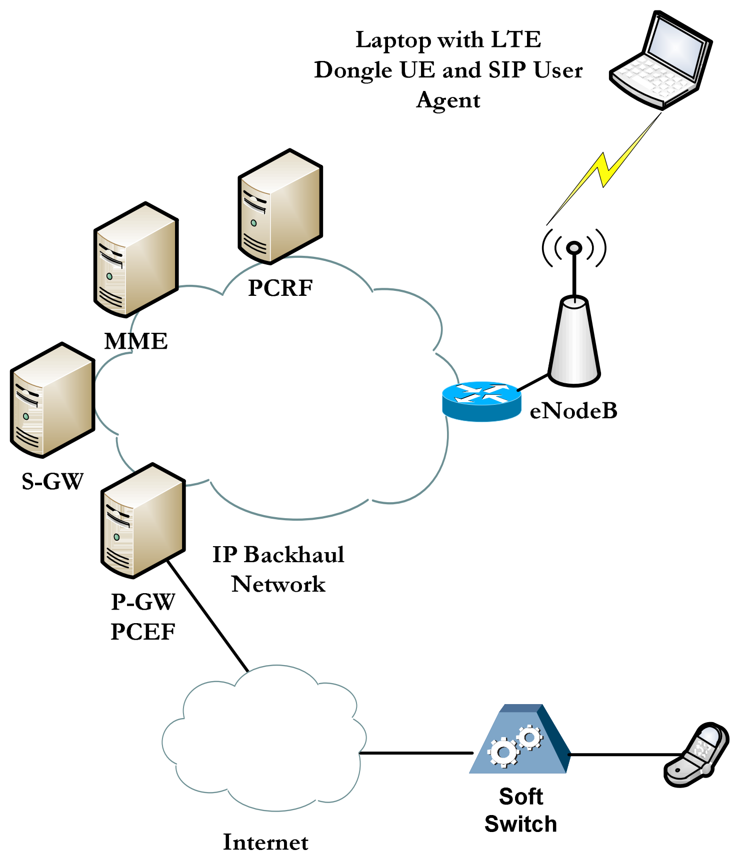

Figure 7-7. LTE Guaranteed Bit Rate (GBR) Bearer Initiation Test Model

The system model was configured in the following manner; the eNodeB, Serving Gateway (S-GW), Mobility Management Entity (MME), Packet Data Network Gateway (P-GW) and Policy and Charging Rules Function (PCRF) were connected to each other via a singular Virtual Local Area Network (VLAN) subnet. An Ethernet backhaul network interlinked the Evolved Packet Core (EPC) components with the Access Network via a fibre connection originating from a switch before the Evolved Packet Core (EPC) components and terminating at the eNodeB. Signalling and traffic paths between the Evolved Packet Core (EPC) and eNodeBs were separated, with the S1 and X2 interfaces occupying different Virtual Local Area Network (VLAN) subnets. The eNodeB on the Access Network was located about 2 Km line of sight from the Evolved Packet Core (EPC).

The test laptop which also consisted of the Session Initiation Protocol (SIP) User Agent was connected to the eNodeB via an LTE dongle. An external network connection was provided to the Packet Data Network Gateway (P-GW) thereby allowing a direct path from any of the User Equipment (UE) to this external network. The external connection consisted of traffic routing to a carrier grade Softswitch. The carrier grade Softswitch enabled voice calls to any Public Switched Telephone Network (PSTN) or SIP Agent reachable.

On the core of the LTE network, the Policy and Charging Rules Function (PCRF) was configured to consist of a profile that had a non-Guaranteed Bit Rate Bearer (non-GBR) with QoS Class Identifier (QCI) 9 as well as a profile that had a Guaranteed Bit Rate Bearer (GBR) with QoS Class Identifier (QCI) 1 for the sole purpose of voice traffic. The non-Guaranteed Bit Rate Bearer (non-GBR) was to be set during the User Equipment (UE) attach procedure and the Guaranteed Bit Rate Bearer (GBR) after the voice recognition trigger.

The following was also included as part of the testing:

1. The voice Bearer shall be triggered via an external interface, thereby confirming that the static voice procedure, User Equipment (UE) Application procedure and Deep Packet Inspection trigger procedure is operational.
2. Only voice traffic shall be allowed to flow through this Bearer and no other traffic.

Another part of the original contribution of this work included proposing the assignment of an attribute value by the voice recognition algorithm or any of the Bearer initiation procedures above to the Policy and Charging Rules Function (PCRF) via its Rx interface to initiate the Guaranteed Bit Rate Bearer (GBR) with QoS Class Identifier (QCI) 1 for voice traffic. For the purposes of emulating a trigger, the Access Point Name (APN) from the User Equipment (UE) attach procedure was used as an initiator of the voice Guaranteed Bit Rate Bearer (GBR) via the external interface of the LTE network. On recognition of the Access Point Name (APN) an attribute value would be passed through the Rx interface to initiate the Guaranteed Bit Rate Bearer (GBR) with QoS Class Identifier (QCI) 1 for voice traffic.

New theory that was also tested included the manipulation of the Traffic Flow Template (TFT) for the voice Guaranteed Bit Rate Bearer (GBR) such that the filtering mechanism used to classify traffic into the voice Guaranteed Bit Rate Bearer (GBR) was allocated only the Softswitch IP during the voice Guaranteed Bit Rate Bearer (GBR) attach process. This was done by assigning the Softswitch IP as part of the Policy Charging and Control (PCC) rules assigned by the Policy and Charging Rules Function (PCRF). When the Policy Charging and Control (PCC) rules were applied to the Traffic Flow Templates (TFT) at the eNodeB and PDN Gateway (P-GW) they possessed a fixed destination IP at the eNodeB and a fixed source IP address at the PDN Gateway (P-GW) that was the Softswitch IP address. This in turn meant that only traffic flowing to the Softswitch IP address would traverse the voice Guaranteed Bit Rate Bearer (GBR).

The LTE dongle was set-up with the Session Initiation Protocol (SIP) User Agent registered to the carrier grade Softswitch. Spurious traffic was simultaneously uploaded and downloaded by the test laptop with multiple voice calls being made on the same laptop. An alternative Softswitch IP address was then configured on the Session Initiation Protocol (SIP) User Agent to register again to Softswitch and multiple voice calls were made in the same test scenario. This was done so that a comparison can be made between traffic going to the Softswitch IP that was associated with the voice Guaranteed Bit Rate Bearer (GBR) and the other Softswitch IP which was not associated with the default Bearer.

7.6.3 Analysis of voice Guaranteed Bit Rate Bearer (GBR) Results

The experiments were conducted in accordance with the experimental design. A trace of the attach message from the Mobility Management Entity (MME) was used to confirm the initiation of the voice Guaranteed Bit Rate Bearer (GBR). The Mobility Management Entity (MME) trace is illustrated in Figure 7-8 below.

No.	Protocol Type	Direction	Event
1	EMM	RECEIVE	Attach request
2	EMM	SEND	Authentication request
3	S1AP	RECEIVE	INITIAL UE MESSAGE
4	S1AP	SEND	DOWNLINK NAS TRANSPORT
5	S1AP	RECEIVE	UPLINK NAS TRANSPORT
6	EMM	RECEIVE	Authentication response
7	EMM	SEND	Security mode command
8	S1AP	SEND	DOWNLINK NAS TRANSPORT
9	S1AP	RECEIVE	UPLINK NAS TRANSPORT
10	EMM	RECEIVE	Security mode complete
11	ESM	RECEIVE	PDN connectivity request
12	GTP	SEND	Delete Session Request
13	GTP	RECEIVE	Delete Session Response
14	ESM	SEND	ESM information request
15	S1AP	SEND	DOWNLINK NAS TRANSPORT
16	S1AP	RECEIVE	UPLINK NAS TRANSPORT
17	ESM	RECEIVE	ESM information response
18	GTP	SEND	Create Session Request
19	GTP	RECEIVE	Create Session Response
20	ESM	SEND	Activate default EPS bearer context request
21	ESM	SEND	Activate dedicated EPS bearer context request
22	EMM	SEND	Attach accept
23	S1AP	SEND	INITIAL CONTEXT SETUP REQUEST
24	S1AP	RECEIVE	UE CAPABILITY INFO INDICATION
25	S1AP	RECEIVE	INITIAL CONTEXT SETUP RESPONSE
26	S1AP	RECEIVE	UPLINK NAS TRANSPORT
27	EMM	RECEIVE	Attach complete
28	ESM	RECEIVE	Activate default EPS bearer context accept
29	S1AP	RECEIVE	UPLINK NAS TRANSPORT
30	ESM	RECEIVE	Activate dedicated EPS bearer context accept
31	GTP	SEND	Create Bearer Response
32	GTP	RECEIVE	Modify Bearer Response
33	DiameterAPP	SEND	Notify Request
34	DiameterAPP	RECEIVE	Notify Answer

Figure 7-8. Mobility Management Entity (MME) trace for voice specific Guaranteed Bit Rate Bearer (GBR) Creation

Based on the trace it can be seen that the Guaranteed Bit Rate Bearer (GBR) is successfully being created for voice traffic. Steps 1 to 19 represent the Attach procedure where the LTE User Equipment (UE) is effectively attached to the network. Steps 20 to 27 illustrate the activation of the default Bearer which is a Non-Guaranteed Bit Rate (Non-GBR) Bearer for basic data or internet connectivity. Steps 28-34 illustrate the request for the voice Guaranteed Bit Rate (GBR) Bearer, as well as answer activation from the Policy and Charging Rules Function (PCRF). To explain in more depth, on recognition of the Access Point Name (APN) from the external interface during the default Bearer set-up process the Policy and Charging Rules Function (PCRF) is assigned an attribute value via its Rx interface. On recognition of this attribute value, the Policy and Charging Rules Function (PCRF) sends through the Policy Charging and Control (PCC) rules which sets the modified Traffic Flow Template (TFT) such that the voice Guaranteed Bit Rate (GBR) Bearer is created and only traffic destined to the Softswitch traverses this Bearer.

Although based on the traces listed above and testing conducted in Chapter 6 it can be confidently said that Quality of Service (QoS) is successfully being applied to voice traffic over the LTE network. However in line with previous testing it was deemed necessary to conduct latency as well as jitter tests based on voice calls made to either Softswitch IP. The analysis of Real Time Protocol (RTP) streams for either Softswitch IP is illustrated in Figure 7-9 and 7-10.

Average latency and mean jitter results were recorded using the Wireshark Protocol Analyser on both the uplink and downlink directions. More information on the Wireshark Protocol Analyser can be found in Appendix A. Greater than 9 test-sets were recorded with each test-set consisting of hundreds of Real Time Protocol (RTP) packets. Individual test results that exceeded 10% of the average of the grouped test-set resulted in that test-set being discarded and the experiment repeated.

7.6.4 Latency and Jitter Results for Real Time Protocol (RTP) Packets across the Different LTE Bearers

As per Figure 7-9 and Figure 7-10, the Guaranteed Bit Rate (GBR) Bearer on average outperformed the Non-Guaranteed Bit Rate (Non-GBR) Bearer. The results for latency in the upload direction indicated a greater than 20% reduction in latency by the Guaranteed Bit Rate (GBR) Bearer as compared to the Non-Guaranteed Bit Rate (Non-GBR) Bearer. The results for the latency in the download direction indicated a marginal reduction of 5-10% for the Guaranteed Bit Rate (GBR) Bearer over the Non-Guaranteed Bit Rate (Non-GBR) Bearer.

The results for jitter in the upload direction were significant because this indicated a greater than 100% reduction in jitter for the Guaranteed Bit Rate (GBR) Bearer as compared to the Non-Guaranteed Bit Rate (Non-GBR) Bearer. The results for jitter in the download direction were similar. A possible reason for the similar results includes insufficient spurious traffic being passed through the system to significantly affect the latency and jitter on the downlink only. It has to be noted that the LTE system that the test-set was being conducted on, had an 82 Mb/s theoretical download limit based on 10 MHz spectrum in the 1800 MHz band.

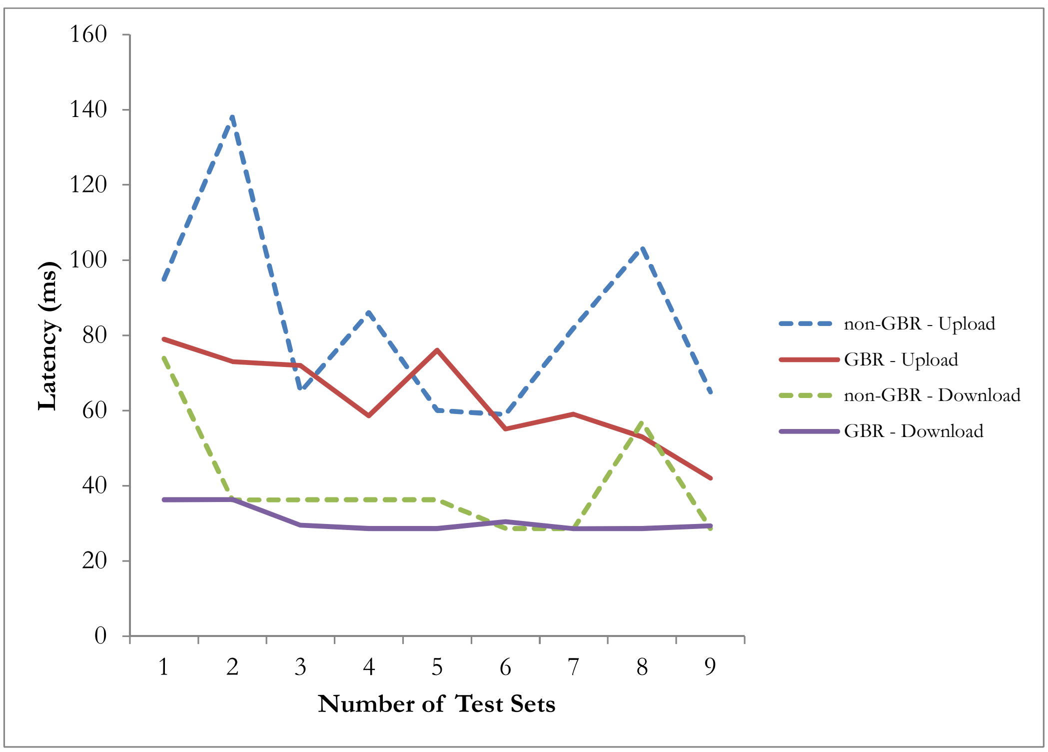

Figure 7-9. Average Latency for calls Destined to Non-Guaranteed (Non-GBR) Bit Rate Bearer versus calls to the voice Guaranteed Bit Rate (GBR) Bearer

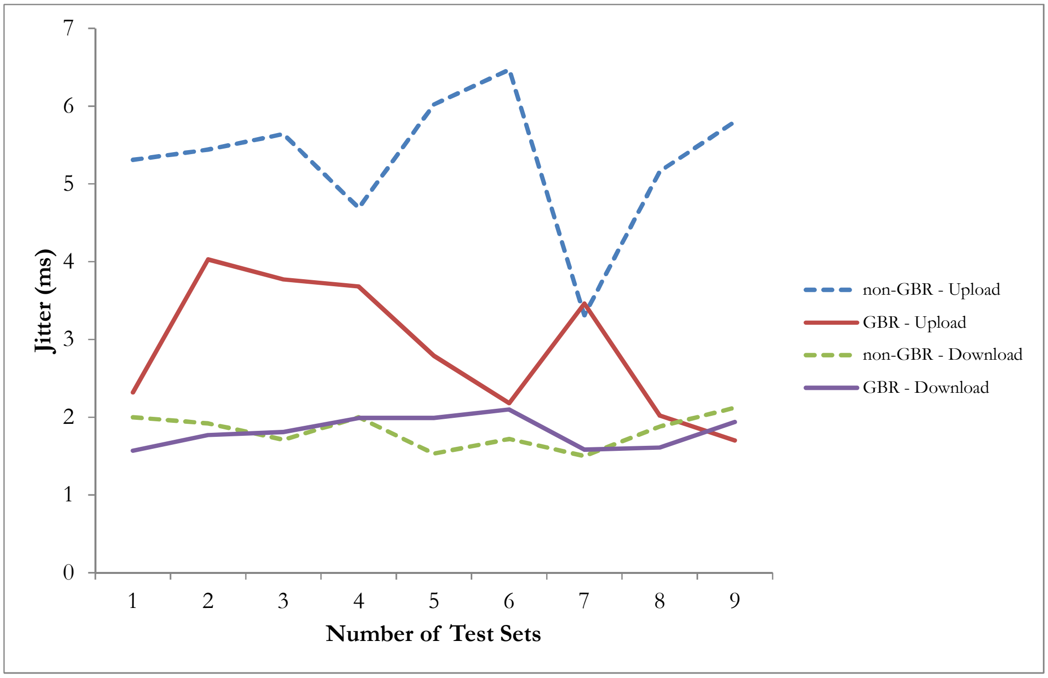

Figure 7-10. Average Jitter for calls Destined to Non-Guaranteed (Non-GBR) Bit Rate Bearer versus calls to the voice Guaranteed Bit Rate (GBR) Bearer

The above tests proved that the Guaranteed Bit Rate (GBR) Bearer outperformed the Non-Guaranteed Bit Rate (Non-GBR) Bearer in terms of Real Time Protocol (RTP) voice streams. From a subjective voice quality perspective, there was an audible difference between the voice calls listened to over the Guaranteed Bit Rate (GBR) Bearer, as compared to the Non-Guaranteed Bit Rate (Non-GBR) Bearer. The voice calls over the Guaranteed Bit Rate (GBR) Bearer were perceived clear and with minimal speech stutter. A minute volume of periodic echo was observed on the received (Non-LTE) side which was attributed to a fault on the telephony device. However this was not sufficient to disrupt the voice call or testing procedures.

7.7 Conclusion

The sections above have presented procedures for the initiation of a Quality of Service (QoS) Bearer for voice traffic over an LTE network. The WiMAX system was not investigated because it did not allow for Quality of Service (QoS) Bearer initiation.

However the LTE system did allow for Quality of Service (QoS) Bearer initiation via a Diameter Gx interface terminating into the Packet Data Network Gateway (P-GW) from a Policy and Charging Rules Function (PCRF).

Initially the procedure for a Static Voice Bearer set-up over LTE was presented where the Bearer is not invoked but rather set up during the existing User Equipment (UE) attach procedure, yet combining this procedure with the Modify Bearer Request procedure. IP Multimedia Subsystem (IMS) initiated voice Bearer set-up is also looked at as well as an Application initiated voice Bearer set-up. The differences between IP Multimedia Subsystem (IMS) and Application Bearer initiation is that IP Multimedia Subsystem (IMS) uses the Policy and Charging Rules Function (PCRF) to force the voice Bearer creation from the network side, i.e. from beyond the LTE core, however the Application Bearer initiation uses a pull down of Policy Control and Charging (PCC) rules that already exist in the Policy and Charging Rules Function (PCRF) to set the Guaranteed Bit Rate (GBR) Bearer for voice.

Deep Packet Inspection (DPI) initiated voice Bearer set-up using Heuristic Analysis via the Serving Gateway (S-GW), and Deep Packet Inspection (DPI) initiated voice Bearer set-up using Heuristic Analysis via the eNodeB are proposed for consideration. The notion is to recognise Over the Top (OTT) or other voice and use the processes presented to trigger a Quality of Service (QoS) Bearer for voice traffic. This required the installation of a Deep Packet Inspection (DPI) engine. Positioning the Deep Packet Inspection (DPI) engine at either point is acceptable; however the Deep Packet Inspection (DPI) engine at the eNodeB is advantageous because traffic is inspected at the access of the network which is more efficient in terms of traffic control and set-up time for the initiation of the Guaranteed Bit Rate (GBR) Bearer.

The validity of the procedures was demonstrated via the Bearer initiation experiment, where the original contribution of this work included proposing the assignment of an attribute value via an external interface by the voice recognition algorithm or any of the Bearer initiation procedures above to the Policy and Charging Rules Function (PCRF) via its Rx interface. New theory that was also tested included the manipulation of the Traffic Flow Template (TFT) for the voice Guaranteed Bit Rate Bearer (GBR), such that the filtering mechanism used to classify traffic into the voice Guaranteed Bit Rate Bearer

(GBR) was allocated only the Softswitch IP during the voice Guaranteed Bit Rate Bearer (GBR) attach process.

The testing proved that on recognition of the Access Point Name (APN) via an external interface, an attribute value would be assigned through the Rx interface to initiate the Guaranteed Bit Rate Bearer (GBR) with QoS Class Identifier (QCI) 1 for voice traffic. The manipulation of the Traffic Flow Template (TFT) profile in turn realised that only traffic flowing to and from the Softswitch IP address would traverse the voice Guaranteed Bit Rate Bearer (GBR).

Latency and jitter results confirmed that Quality of Service (QoS) was being applied to the Real Time Protocol (RTP) voice stream. Two Bearers were tested namely: the Guaranteed Bit Rate (GBR) Bearer and Non-Guaranteed Bit Rate (Non-GBR) Bearer, on average the Guaranteed Bit Rate (GBR) Bearer outperformed the Non-Guaranteed Bit Rate (Non-GBR) Bearer. The results for latency in the upload direction indicated a greater than 20% reduction in latency by the Guaranteed Bit Rate (GBR) Bearer as compared to the Non-Guaranteed Bit Rate (Non-GBR) Bearer. The results for the latency in the download direction indicated a marginal reduction of 5-10% for the Guaranteed Bit Rate (GBR) Bearer over the Non-Guaranteed Bit Rate (Non-GBR) Bearer. The jitter results in the upload direction provided an indication of a greater than 100% reduction in jitter for the Guaranteed Bit Rate (GBR) Bearer as compared to the Non-Guaranteed Bit Rate (Non-GBR) Bearer. The results for jitter in the download direction were similar.

The experiment demonstrated that Quality of Service (QoS) Bearer initiation is possible and available for integration with the Heuristic Analysis voice Admission Control (AC) scheme. A practical solution was presented that exhibited quality voice over an LTE network without the addition of IP Multimedia Subsystem (IMS). This concludes the research and results segment of the thesis with the following chapter presenting a critical analysis of the work completed in terms of Admission Control (AC), Bearer or Service Flow selection, initiation of the Bearer or Service Flow as well as mapping from this component to the transportation network.

8 Critical Assessment of Results

This current research "Voice Quality Control in Packet Switched Wireless Networks" offers a precise yet holistic methodology in the provisioning of Quality of Service (QoS) for voice over packet switched wireless networks. As part of the original contributions of this research, a unique quality control methodology was presented, focused on the development of an original voice specific Admission Control (AC) scheme, Quality of Service (QoS) Service Flow or Bearer selection framework for WiMAX and LTE 'all IP' networks, a mapping methodology for the routing of Quality of Service (QoS) Bearers or Service Flows to the transport network and an automated initiation process of Quality of Service (QoS) Bearers or Service Flows for voice streams.

In the proposed voice Quality Control methodology, mapping of the Bearer or Service Flow to the transport or backhaul network was recommended as more of a policy statement based on best practices. Best practices dictated that Differentiated Service Code Point (DSCP) Expedited Forwarding (EF) or Multi-Protocol Label Switching (MPLS) Experimental Bit 5 be utilised as the most appropriate classifier for voice traffic as these categorisations ensured that voice traffic always received the highest possible priority treatment in the relevant transportation or backhaul queuing framework.

The creation and simulation of a voice specific Admission Control (AC) model exemplified that a voice specific Admission Control (AC) model can be successfully implemented as long as voice can be effectively recognised and prioritised as it enters into the packet switched wireless network. The solution provided a view to the provisioning of a new Heuristic Analysis recognition algorithm at the Admission Control (AC) component. This proposed new scheme used Deep Packet Inspection (DPI) filters to recognise Over the Top (OTT) and other voice traffic. Thereafter facilitated a response into the Admission Control structure as well as trigger a Quality of Service Bearer or Service Flow for voice traffic. The new proposed Heuristic Analysis algorithm was a satisfactory option for the recognition of voice traffic because the advantages of Deep Packet Inspection (DPI) could now be leveraged for Over the Top (OTT) voice recognition. As a part of the original contribution of this research, the Objects and Axioms for the new Heuristic Analysis algorithm were derived via experimentation where Session Initiation Protocol (SIP) and the Google Talk application were analysed.

Options for the recognition of Session Initiation Protocol (SIP) were presented with the Session Initiation Protocol (SIP) non-numbered categorisation, incorporated with a fixed time duration and Softswitch IP yielding 100% recognition for the entire sample set. A Google talk voice Heuristic Analysis algorithm was derived to trigger (forward an attribute value) a Guaranteed Bit Rate (GBR) bearer for Over the Top (OTT) voice. It was noted that dependent on the method of recognition this affected the probability of successfully recognising, admitting a voice stream and allocating resources accordingly. This in turn had an effect on the probability of the voice calls not being offered a Quality of Service (QoS) Bearer or Service Flow by the system.

The original voice specific Admission Control (AC) model was specified to integrate with the Heuristic Analysis recognition algorithm. One of the key imperatives of the new voice Admission Control (AC) model was to consider voice traffic as crucial to the Admission Control (AC) model yet not catastrophically affecting resource allocation to other services participating in the overall system. In terms of resource allocation fairness which affects the quality of experience for other services, a new factor *p* was introduced which allowed an operator to control the radio resource sector capacity that voice traffic could cannibalise. As part of the original contribution of this research, *p* represented a reserve margin that created a balance for admittance between voice traffic and all other services. *p* provided operators with the ability to apply traffic engineering to the voice specific Admission Control (AC) model. The simulation results indicate that as *p* increased the blocking probability increased due to the operator allocated bandwidth for voice decreasing with an increased reserve margin. This proved the correct functionality of *p*.

The selection of the Bearer or Service Flow over the air interface to be used for voice traffic was of importance because this Bearer or Service Flow needed to support the network metrics for voice. Testing of the WiMAX Service Flow types yielded that the enhanced real time Polling Service (ertPS) out-performed all other Service Flows due to its combined superior latency and jitter performance as compared to the other Service Flow types. Although the Unsolicited Grant Service (UGS) performed better that the enhanced real time Polling Service (ertPS) in the latency tests the difference was not significant in relation to the 150ms threshold. The enhanced real time Polling Service (ertPS) was 23.3% of the total threshold as compared to 18.6% for the Unsolicited Grant

Service (UGS). Considering the jitter results, the enhanced real time Polling Service (ertPS) was 70% of the total threshold as compared to 86% for the Unsolicited Grant Service (UGS) a difference of 16% which was the deciding factor. The testing of the LTE Service Flow types yielded results indicating the QoS Class Identifier (QCI) 2 outperformed all other Bearers due to QoS Class Identifier (QCI) 2 illustrating superior latency and jitter performance as compared to the other Bearers. It was however highlighted that QoS Class Identifier (QCI) 1 and QoS Class Identifier (QCI) 3 exhibited good performance in the latency tests and were 18.9% of the 150ms threshold, similar to QoS Class Identifier (QCI) 2. The jitter results were where the difference was noticeable, that is, QoS Class Identifier (QCI) 2 was 80% of the threshold value of 5ms as compared to 120%, and 168% for QoS Class Identifier (QCI) 1 and 3. It has to be noted that QoS Class Identifier (QCI) 1 achieved fair jitter performance in individual test sets even though the average value is 120%.

Considering that a voice call may traverse multiple network domains, an original voice network parameter factor *f* was proposed for introduction by equipment vendors where each of the vendor designed system Quality of Service (QoS) Bearers or Service Flows are categorised with factor *f*. *f* is classified as the network quality metric parameter for an individual vendor designed system or network domain in relation to the overall voice system network quality metric threshold. f_p represented the sum of the maximum packet loss for a network domain in relation to the overall maximum packet loss, f_j represented the sum of the maximum packet jitter for a network domain in relation to the overall maximum packet loss and f_d represents the sum of the maximum packet delay for a network domain in relation to the overall maximum packet loss. The testing results from the LTE QoS Class Identifiers (QCI) Bearers and WiMAX Service Flows were applied to the methodology of factor *f* in respect of f_d and f_p.

With regard to LTE it was observed that in terms of f_{d1} all QoS Class Identifiers (QCI) performed below the threshold factor of 1. QoS Class Identifier (QCI) 1, 2 and 3 however performed better than all other QoS Class Identifiers by a margin of 80%. In respect of f_{j1} it was observed that QoS Class Identifier (QCI) 1 and 2 performed to an acceptable standard however QoS Class Identifier (QCI) 2 outperformed all other Bearers. It was observed that even though the average result for QoS Class Identifier (QCI) 1 was above the threshold factor of 1 many of the test-set results were below the threshold factor of 1. QoS Class Identifier (QCI) 2 was deemed the most robust in terms of being paired with other network domains in a voice system. With regard to WiMAX it was noted for the WiMAX Service Flows that in terms of f_{d1} all Service Flows performed below the threshold factor of 1. Unsolicited Grant Service (UGS) and enhanced real Time Polling Service (ertPS) recoded the lower factor calculation hence outperformed all other Service Flows. In respect of f_j it was observed that the Unsolicited Grant Service (UGS) and enhanced real Time Polling Service (ertPS) were the only factors that performed below the threshold factor of 1, hence had sufficient room to be paired in-line with other network domains.

In the case of LTE, the initiation of a Guaranteed Bit Rate (GBR) Bearer after identification by the Deep Packet Inspection (DPI) engine as well as admittance by the voice specific Admission Control (AC) model was further investigated. Original voice Bearer initiation procedures were presented, each utilising the Policy and Charging Rules Function (PCRF) of the LTE system. A part of the original contribution of this work included, the proposal of the assignment of an attribute value by the voice recognition algorithm or any of the other voice Bearer initiation procedures to the Policy and Charging Rules Function (PCRF) via its Rx interface to initiate the voice Guaranteed Bit Rate Bearer (GBR) with a QoS Class Identifier (QCI) 1. This request allowed for a pull of Policy Charging and Control (PCC) rules from the Policy and Charging Rules Function (PCRF) for the enablement of the voice Guaranteed Bit Rate (GBR) Bearer in LTE.

Other original contributions included the manipulation of the Traffic Flow Template (TFT) for the voice Guaranteed Bit Rate Bearer (GBR) such that the filtering mechanism used to classify traffic into the voice Guaranteed Bit Rate Bearer (GBR), was allocated only the Softswitch IP during the voice Guaranteed Bit Rate Bearer (GBR) attach process. This in turn meant that only traffic flowing to the Softswitch IP address would traverse the voice Guaranteed Bit Rate Bearer (GBR).

In the finally testing phase, the LTE voice Bearer initiation schemes were tested demonstrating that the Guaranteed Bit Rate (GBR) Bearer initiation procedures listed above was functional. An attribute value was assigned via an external interface to the Policy and Charging Rules Function (PCRF) Rx interface which allowed for the creation of a voice only Guaranteed Bit Rate (GBR) Bearer. Traces on the User Equipment (UE) attach procedure from the Mobility Management Entity (MME) confirmed that the voice only Guaranteed Bit Rate (GBR) Bearer was operational.

In order to test the voice only Guaranteed Bit Rate (GBR) Bearer including manipulation of the Traffic Flow Template (TFT) profile, an alternative Softswitch IP was configured in the Softswitch. Testing to both Softswitch IP addresses were conducted, confirming that only traffic flowing to and from the Softswitch IP address that was configured in the Traffic Flow Template (TFT) of the voice Guaranteed Bit Rate Bearer (GBR) did receive superior network performance. Latency and jitter results confirmed that Quality of Service (QoS) was being applied to the Real Time Protocol (RTP) voice stream. The voice Guaranteed Bit Rate (GBR) Bearer outperformed the Non-Guaranteed Bit Rate (Non-GBR) Bearer. The results for latency in the upload direction indicated a greater than 20% reduction in latency by the voice Guaranteed Bit Rate (GBR) Bearer as compared to the Non-Guaranteed Bit Rate (Non-GBR) Bearer. The results for the latency in the download direction indicated a marginal reduction of 5-10% for the voice Guaranteed Bit Rate (GBR) Bearer over the Non-Guaranteed Bit Rate (Non-GBR) Bearer.

In respect of the overall testing of the system, Heuristic Analysis was tested individually from the Bearer or Service Flow initiation framework because it was to be proven that the best possible configuration for voice across each of the components was successfully discovered and could be implementable. It can be noted with confidence that the pairing of each of the individual components will be successful because each component dealt with distinct issues. The Admission Control (AC) component dealt with probability of blocking voice circuits yet the Bearer / Service Flow or transport network dealt with network metrics, nonetheless it all had an influence on the voice quality control in a packet switched wireless network.

The experiments presented demonstrated that Quality of Service (QoS) Bearer initiation for voice traffic is possible and available for integration with the Heuristic Analysis voice Admission Control (AC) scheme, a statically configured Bearer initiation scheme, Application initiated Bearer initiation scheme or Deep Packet Inspection (DPI) initiated Bearer initiation scheme. A practical solution was presented that exhibited quality voice over an LTE network without the addition of IP Multimedia Subsystem (IMS). The solution provided could be construed as a competitor to IP Multimedia Subsystem (IMS). In terms of voice traffic this assumption is correct, however it has to be noted that IP Multimedia Subsystem (IMS) is a larger framework for multimedia services and not just only voice. Of greater importance is that such a solution may have an impact for providers that do not have IP Multimedia Services (IMS) but have a Softswitch in their network. Such providers do not need to focus on Circuit Switch Fallback (CSFB), Voice over LTE Generic Access (VoLGA) or similar options because they can now use their native LTE network for voice traffic. The major drawback in this approach is the current lack of handsets in the LTE space. Handsets are required that are able to house a Session Initiation Protocol (SIP) agent, such handsets do exist but LTE handsets are currently expensive, power hungry and lack variety in terms of the number of available devices. This will change in the future. Finally the summary conclusions and further work are presented in the last chapter below.

9 Conclusion and Further Work

As has been highlighted, in order to carry good quality voice, physical resources assigned to voice traffic over the transportation network need to be made available above all other traffic classes [13]. The research embarked on presented a unique quality control methodology that enabled quality voice over packet switched wireless networks such as LTE and WiMAX. This was done by developing or utilising the above mentioned Quality of Service (QoS) features so as to ensure that an acceptable level of voice quality is maintained in comparison to 2nd Generation or 3rd Generation wireless networks.

The background work to this research presented an investigation into the Radio Link Control (RLC), Media Access Control (MAC), Scheduling schemes and the Admission Control (AC) structures defined in the WiMAX and LTE standards. The background work highlighted that the Radio Link Control (RLC) and Media Access Control configuration contributed significantly to Quality of Service (QoS) over the air interface. Service Flows or QoS Class Identifiers (QCI) were presented that represent specific Radio Link Control (RLC) and Media Access Control configurations. An investigation into Bearers or Service Flows were conducted in order to determine the most appropriate fit for voice traffic with UGS, rtPS, ertPS, QCI 1, QCI 2 or QCI 3 specifically identified with the most potential to carry voice traffic.

An assessment of research initiatives and the interim options that operators may have to carry quality voice over packet switched wireless networks were investigated highlighting Circuit Switch Fallback (CSFB), Voice over LTE Generic Access (VoLGA), Over the Top (OTT) voice and IP Multimedia Subsystem (IMS) as well as advantages and disadvantages of these options. The value of these options was presented such that operators are equipped with the ability to evolve their existing networks yet maximise on investment already made into 2nd Generation or 3rd Generation networks. The interim options are also significant in the enablement of voice over an LTE network without the complexity of deploying a new IP Multimedia Subsystem (IMS).

IP Multimedia Subsystem (IMS) is the standards bodies preferred option for network evolution however it is a network revolutionary shift from traditional circuit switched platforms or to a lesser extent a Softswitch platform. For an operator to implement IP

Multimedia Subsystem (IMS) their entire switching core needs to be replaced, operators find it more acceptable for their Access Network to be replaced than for their core voice network to be replaced. The reasons are that operators do not want to disrupt existing wholesale and enterprise customers unless they actually need to. IP Multimedia Subsystem (IMS) has the advantage of enabling multimedia services to all customers. Operators need to find a point of evolution when multimedia services become important to their wholesale and enterprise customers it will then become relevant to migrate to the IP Multimedia Subsystem (IMS).

The proposed methodology in providing Quality of Service (QoS) over the Radio Access and Core portions of a packet switched wireless network was presented highlighting a unique resource management framework for voice. Voice pointers representing a framework at the ingress of the Admission Control (AC), Admission Control (AC) to Bearer or Service Flow mapping and Bearer or Service Flow to Transportation mapping were illustrated describing an alignment of voice traffic conditioning across each of the wireless system nodal elements with the theme of low latency, jitter and packet loss. At this junction the importance of the Admission Control (AC) and Bearer or Service Flow selection became prevalent voice.

An original Heuristic Analysis approach was introduced, incorporated with the Admission Control (AC) component to ensure that voice packets at varying layers of the Open Systems Interconnection (OSI) stack could be analysed and the correct Admission Control (AC) criteria exercised. In the research conducted it was proposed that recognition of voice packets be done via Deep Packet Inspection (DPI) at the eNodeB level. Deep Packet Inspection (DPI) was proposed to be positioned at the eNodeB or Access Service Network Gateway (ASN-GW) level in order to enable the recognition of Session Initiation Protocol (SIP) voice or Over the Top (OTT) voice at the nearest point in the wireless network to the subscriber. An original contribution of this work included the proposal of a variable Admission Control (AC) scheme providing operators the option to vary capacity requirements based on a traffic engineering approach via an element p.

After the Admission Control (AC) node, in line with selection of voice applicable QoS Class Identifiers and Service Flows, both LTE and WiMAX frameworks were tested with the test set-up consisted of live WiMAX and LTE test beds installed and configured as a trial network. Latency, jitter and packet loss results were gleaned during various applied test-sets. The test results illustrated a greater performance in terms of latency and jitter when utilising the enhanced real time Poling Service (ertPS) and the Unsolicited Grant service (UGS) as compared to all other Service Flow types with the enhanced real time Poling Service (ertPS) identified as the preferred option. In respect of LTE, the QoS Class Identifier (QCI) 1, 2 and 3 were deemed suitable for voice, however QoS Class Identifier (QCI) 2 became the favoured option due to it demonstrating superior jitter performance that the other QoS Class Identifier (QCI) types. The control aspect of this research was highlighted by the fact that the network operator has a choice or options to shape the Admission Control (AC), Bearer or Service Flow selection to determine the level of Quality of Service (QoS) required for voice traffic.

In respect of voice traffic traversing multiple network domains it was proposed that a new voice network parameter factor *f* be introduced by vendors, where each of the Quality of Service (QoS) Bearer or Service Flow can be categorised with this factor *f*. *f* is classified as the network quality metric parameter for an individual network domain in relation to the overall required network quality metric parameter. It is anticipated that the proposed factor will provide operators and system integrators a fair indication of the robustness of their wireless network in terms of being paired with other network domains [42].

In the case of LTE, procedures for the initiation of a Quality of Service (QoS) Bearer for voice traffic were presented. Initially the procedure for a Static Voice Bearer set-up was presented. IP Multimedia Subsystem (IMS) initiated voice Bearer set-up procedure was viewed as well as an Application initiated voice Bearer set-up procedure. An original Deep Packet inspection procedure was also presented with the initiation of a voice Guaranteed Bit Rate (GBR) Bearer after voice traffic identification by a Deep Packet Inspection (DPI) engine. A part of the original contribution of this work included the proposal of the assignment of an attribute value by the Heuristic Analysis voice recognition algorithm or any of the other voice Bearer initiation procedures to the Policy and Charging Rules Function (PCRF). This was done via the Policy and Charging Rules Function (PCRF) Rx interface to initiate the Guaranteed Bit Rate (GBR) Bearer with QoS Class Identifier (QCI) 1 for voice traffic as well as apply Traffic Flow Template

(TFT) manipulation to only allow voice traffic over the Guaranteed Bit Rate (GBR) Bearer.

Further work on this research may include; exploration into alternative methods of invoking Quality of Service (QoS) Bearers for LTE networks, further analysis and testing of recognition methods for other Over the Top (OTT) applications, testing the implementation of Deep Packet Inspection (DPI) at the eNodeB level by vendors and modification the Radio Link Control (RLC) and Media Access Control (MAC) for alternative Bearers or Service Flows for voice traffic.

Finally, it can be stated that this research is invaluable to operators with a desire to deploy voice over a next generation packet switched wireless network. It highlights interim options for operators who require control over their wireless network evolutionary path from 3rd Generation to 4th Generation wireless networks yet still ensure that Quality of Service (QoS) for voice is maintained. This research provides a holistic original approach in the provisioning of Quality of Service (QoS) for voice over a packet switched wireless network yet enunciates precise technical criteria for each of the substantive components in the wireless architecture. The uniqueness in the methodology can be found in the Heuristic Analysis approach, the introduction of Deep Packet Inspection (DPI) into the Access layer of the wireless network, the introduction of the voice specific Admission Control scheme with element p, the introduction of factor f for pairing / integration and proposal of the methods to invoke Quality of Service (QoS) Service Flows for Over the Top (OTT) voice. Lastly, voice quality control in packet switch wireless networks is viable and successful as long at the Quality of Service (QoS) framework proposed is adhered to and efficiently implemented.

10 References

[1] Goldstuck A. (Accessed June 2010) World Wide Worx. [Online]. http://www.worldwideworx.com/

[2] Nageshar N, Mneney S H, Sewsunker R, "Universal Telecommunications Access using Satellite Platforms," in *South African Telecommunication Networks and Applications Conference (SATNAC)*, Wild Coast Sun, South Africa, 2001, pp. 1 - 5.

[3] Nageshar N, Mneney S H, "The Southern African Development Community Telecommunications Satellite," in *South African Telecommunication Networks and Applications Conference (SATNAC)*, Cape Town, 2000.

[4] Nageshar N, Mneney S H, Sewsunker R, "Economics Involved in providing Universal Telecommunications Access for the SADC Region," in *South African Telecommunication Networks and Applications Conference (SATNAC)*, Champagne Sports Resort, Kwazulu-Natal, South Africa., 2002, pp. 1 - 5.

[5] Nageshar N, Mneney S H, Sewsunker R, "Economic Analysis of a Satellite Terrestrial Telephony System for Rural SADC," in *IEEE Africon*, George, South Africa, 2002, pp. 401 – 405.

[6] Fitzek F, Reisslein M, Rein S, "Voice Quality Evaluation in Wireless Packet Communication Systems: A Tutorial and Performance Results for ROHC," *IEEE Wireless Communications*, pp. 60 - 67, February 2005.

[7] Alcatel Lucent, "The LTE Network Architecture," ALU, Strategic White Paper 2009.

[8] IEEE Computer Society and IEEE Microwave Theory and Techniques Society, "Part 16: Air Interface for Broadband Wireless Access Systems," IEEE, Standards 802.16 – 2009, 2009.

[9] Lee K, Bahk S, Choi Y, "All-IP 4G Network architecture for efficient mobility and resource management," *IEEE Wireless Communications Journal*, pp. 42 - 46, May 2007.

[10] 3rd Generation Partnership Project, "Evolved Universal Terrestrial Radio Access (E-UTRA); LTE physical layer; General description (Release 9)," 3GPP, Technical Specification 3GPP TS 36.201 V9.1.0, 2010.

[11] Nageshar N, Van Olst R, "Maintenance of Voice Quality Control in the Evolution to Packet Switched Wireless Networks," in *South African Telecommunication Networks and Applications Conference (SATNAC)*, East London, South Africa, 2011.

[12] Neekzad B, Hui J, Vannithamby R, Alasti M, "Quality of Service in WiMAX and LTE Networks," *IEEE Communications Magazine*, pp. 104 - 111, May 2010.

[13] Wang H, Malkamaki E, Tuomaala E, Jiang D, "Principle and Performance of Semi-persistent Scheduling for VoIP in LTE System," in *WICOM proceedings*, 2007, pp. 2861 – 2864.

[14] Helberg A, Schutte J, "A Study of the Effect of MPLS on Quality of Service in Wireless LANS," *South African Institute of Electrical Engineers Journal*, vol. 99, pp. 70 - 76, September 2008.

[15] 3rd Generation Partnership Project, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Stage 2," 3GPP, Technical Specification 3GPP TS 36.300 V9.3.0, 2010.

[16] Nageshar N, Van Olst R, "A Heuristic Analysis Approach to Admission Control for Voice in Packet Switched Wireless Networks," in *IEEE Africon*, Livingstone, Zambia, 2011, pp. 1 - 6.

[17] Toufik I, Baker M, Sesia S, *LTE – The UMTS Long Term Evolution From Theory to Practice*, 1st ed. West Sussex, United Kingdom: John Wiley & Sons Ltd, 2009.

[18] 3rd Generation Partnership Project, "Quality of Service (QoS) concept and architecture (Release 9)," 3GPP, Technical Specification 3GPP TS 23.107 V9.0.0, 2009.

[19] Dornal S. (Accessed September 2011) Wired n Wireless - LTE QoS. [Online]. http://wired-n-wireless.blogspot.com/2009/05/lte-qos.html

[20] IEEE Computer Society and IEEE Microwave Theory and Techniques Society, "IEEE Standard for Local and Metropolitan Area Networks, Air Interface for Fixed Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1," IEEE, Standards 802.16e, 2006.

[21] IEEE Computer Society and IEEE Microwave Theory and Techniques Society, "IEEE Standard for Local and Metropolitan Area Networks, Air Interface for Fixed Broadband Wireless Access Systems," IEEE, Standards 802.16-2004, 2004.

[22] Motorola, "WiMAX: E vs. D, The Advantages of 802.16e over 802.16d," Motorola, White Paper 2007.

[23] Unstrung Insider, "Voice over LTE: Many Questions, No Easy Answers," *Light Reading*, vol. 8, no. 10, October 2009.

[24] 3rd Generation Partnership Project, "Circuit Switched (CS) Fallback in Evolved Packet System (EPS) Stage 2," 3GPP, Technical Specification 3GPP TS 23.272 V9.3.0, 2010.

[25] Arnott R, Kubota M, Kwan R, "On Radio Admission Control for LTE Systems," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, Ottawa, 2010, pp. 1 - 5.

[26] Choi B, Chung M, Lee J, Kwon S, Jae Bae S, "Delay-aware call admission control algorithm in 3GPP LTE system," in *TENCON 2009 - 2009 IEEE Region 10 Conference*, Singapore, 2009, pp. 1 - 6.

[27] Yu M, Zhao A, Chang Y, Yang D, Lei H, "Adaptive Connection Admission Control Algorithm for LTE Systems," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, Singapore, 2008, pp. 2336 - 2340.

[28] Rosa C, Calabrese F, Pedersen K, Mogensen P, Anas M, "Combined Admission Control and Scheduling for QoS Differentiation in LTE Uplink," in *IEEE 68th Vehicular Technology Conference*, 2008, pp. 1 - 5.

[29] 3rd Generation Partnership Project, "IP Multimedia Subsystem (IMS) - Stage 2 (release 11)," 3GPP, Technical Report 3GPP TS 23.228 V11.4.0, 2012.

[30] Mosmondor M, Dobrijevic O, Matijasevic M, Skorin-Kapov L, "Application-Level QoS Negotiation and Signalling for Advanced Multimedia Services in the IMS," *IEEE Communications Magazine*, pp. 108 - 116, July 2007.

[31] Weik P, Vingarzan D, "End-to-end performance of the IP multimedia subsystem over various wireless networks," in *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, Las Vegas, NV, 2006, pp. 183 - 188.

[32] 3rd Generation Partnership Project, "IMS Multimedia Telephony Communication Service and Supplementary Services - Stage 3 (Release 9)," 3GPP, Technical Specification 3GPP TS 24.173 V9.1.0, 2010.

[33] ETSI 3rd Generation Partnership Project (3GPP), "Universal Mobile Telecommunications System (UMTS) LTE - Internet Protocol (IP) multimedia call control protocol based on Session Initiation Protocol(SIP) and Session Description Protocol (SDP)," ETSI & 3GPP, Technical Specification ETSI TS 134 229-1 V9.7.0, Jan 2011.

[34] Elramly S, Ibrahim M, Aziz W, "VoIP Quality Optimization in IP-Multimedia Subsystem (IMS)," in *Computational Intelligence, Modelling and Simulation (CIMSiM), 2010 Second International Conference*, Bali, 2010, pp. 546 - 552.

[35] Jammeh, E, Sun L, Khan A, Ifeachor E, Mkwawa I, "Open IMS Core with VoIP Quality Adaptation," in *Autonomic and Autonomous Systems, 2009. ICAS '09. Fifth International Conference*, Valencia, 2009, pp. 295 - 300.

[36] Ventura N, Good R, "End to End Session Based Bearer Control for IP Multimedia Subsystems," in *IFIP/IEEE International Symposium on Integrated Network Management*, 2009, pp. 497 - 504.

[37] Anand S, Kumar S, Patel K, "A Novel Scalable Architecture for Efficient QoS to Cater IMS Services for Handheld Devices Based on Android," in *Next Generation Mobile Applications, Services and Technologies (NGMAST), 2010 Fourth International Conference*, Amman, 2010, pp. 106 - 111.

[38] Gerlach O, Gessner C, "Voice and SMS in LTE," Rhode & Schwarz, München, White Paper 2011.

[39] Blake S, Baker F, Black D, Nichols K, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," IETF, RFC 2474, 1998.

[40] Rosen E, Viswanathan A, Callon R, "Multiprotocol Label Switching Architecture," IETF - Network Working Group, RFC 3031, 2001.

[41] AROMA, "Final Report on AROMA Algorithms and Simulation results," Sixth Framework Programme, Technical Report AROMA IST-4-027567, 2007.

[42] Nageshar N, Van Olst R, "Regulation of Bearer / Service Flow Selection between Network Domains for Voice over Packet Switched Wireless Networks," in *ITU Kaleidoscope*, Cape Town, South Africa, 2011, pp. 175 - 180.

[43] Person F. (Accessed November 2012) Voice over IP Realized for the 3GPP Long Term Evolution. Ericsson Publications. [Online]. http://www1.ericsson.com/res/thecompany/docs/journal_conference_papers/wireless_access/VTC07F_persson.pdf

[44] Huang Y, Shi J, Yuan Y, Tian L, Dutkiewicz E, Qian M, "A Novel Radio Admission Control Scheme for Multiclass Services in LTE Systems," in *IEEE Global Telecommunications Conference*, 2009, pp. 1 - 6.

[45] Ferrus R, Sallent O, Perez-Romero J, Casadevall F, Olmos J, "A Functional End-to-End QoS Architecture Enabling Radio and IP Transport Coordination," in *WCNC proceedings*, Kowloon, 2007, pp. 4348 – 4353.

[46] Braden R, Clark D, Shenker S, "Integrated Services in the Internet Architecture: an Overview," IETF - Network Working Group, RFC 1633, 1994.

[47] Black D, Carlson M, Davies E, Wang Z, Weiss W, Blake S, "An Architecture for Differentiated Services," IETF - Network Working Group, RFC 2475, 1998.

[48] IEEE Computer Society and IEEE Microwave Theory and Techniques Society, "Traffic Class Expediting and Dynamic Multicast Filtering," IEEE, Standards 802.1p – 1998, 1998.

[49] Allot Communications, "Digging Deeper Into Deep Packet Inspection (DPI)," White Paper 04.2007 Allot Communications, 2007.

[50] Morin S, Allan K, Dolgonow A, "Application-Level Processing for Default LTE Bearer in S-GW," United States Patent Application Publication, Patent Pub. No. US 2010/0067400 A1, 2010.

[51] Kokash N. (Accessed June 2012) An Introduction to Heuristic Algorithms. [Online]. www.citeseerx.ist.psu.edu

[52] Sickafus E, "Heuristics for Solving Technical Problems - Theory, Derivation, Application," Ntelleck, Grosse Ile, Publication 2004.

[53] Combs G. (Accessed April 2011) Wireshark. [Online]. http://en.wikipedia.org/wiki/Wireshark

[54] Wireshark Foundation. (Accessed January 2011) Wireshark. [Online]. http://www.wireshark.org/

[55] Wireshark Wiki. (Accessed January 2011) Wireshark. [Online]. http://en.wikipedia.org/wiki/Wireshark

[56] The MathWorks Inc. (Accessed February 2011) MATLAB The Language of Technical Computing. [Online]. http://www.mathworks.com/products/matlab/

[57] The MathWorks Inc. (Accessed June 2012) MATLAB The Language of Technical Computing. [Online]. http://www.mathworks.com/tagteam/70533_91199v01_MATLABDataSheet_v9.pdf?s_cid=ML2012_bb_datasheet

[58] The IPERF Team. (Accessed April 2011) Wikipedia. [Online]. http://en.wikipedia.org/wiki/Iperf

[59] The National Laboratory for Advanced Network Research (NLANR). (Accessed June 2012) Iperf version 2.0.0. [Online]. http://john.de-graaff.net/docs/iperf_user_docs_v2.0.0/

[60] Muuss M. (Accessed February 2012) Ping Test - Wikipedia. [Online]. http://en.wikipedia.org/wiki/Ping

[61] Sultana S, Rommer S, Frid L, Mulligan C, Olsson M, *SAE and the Evolved Packet Core*, 1st ed. Oxford, UK: Elsevier Ltd., 2009.

[62] Srinivasa Rao V. (Accessed March 2010) Signaling Procedures in LTE. Published in webbuyersguide.com. [Online]. http://www.slideshare.net/allabout4g/lte-signaling

[63] Casner S, Frederick R, Jacobson V, Schulzrinne H, "RTP: A Transport Protocol for Real-Time Applications," IETF - Network Working Group, RFC 1889, 1996.

[64] Muuss M. (Accessed June 2012) PING.C. [Online]. http://www.ping127001.com/pingpage/ping.text

[65] Rohani B, Homayounfar K, "Capacity Enhancement of VoIP over LTE by Stochastic Adaptive Modulation and Coding," IEICE, Technical Report 391, 2009.

11 Appendix A – Testing, Simulation Software and Probes Used

11.1 Iperf

Iperf is a commonly used network testing tool that can create TCP and UDP data streams and measure the throughput of a network that is carrying these streams. Iperf is a tool for network performance measurement written in C++ and was developed by the Distributed Applications Support Team (DAST) at the National Laboratory for Applied Network Research (NLANR) [58].

Iperf allows the user to set various parameters that can be used for testing a network, or alternately for optimizing or tuning a network. Iperf has a client and server functionality, and can measure the throughput between the two ends, either unidirectional or bi-directionally. It is open source software and runs on various platforms including Linux, Unix and Windows [58].

The primary goal of Iperf is to help in tuning TCP connections over a particular path. The most fundamental tuning issue for TCP is the TCP window size, which controls how much data can be in the network at any one point. If the TCP window size is too small, the senders will at times and get poor performance [59].

Iperf creates a constant bit rate UDP stream. This is a very artificial stream, similar to voice communication but not much else. The server detects UDP datagram loss by ID numbers in the datagrams. Usually a UDP datagram becomes several IP packets. Losing a single IP packet will lose the entire datagram. To measure packet loss instead of datagram loss Iperf makes the datagrams small enough to fit into a single packet, using the -l option. The default size of 1470 bytes works for Ethernet. Out-of-order packets are also detected. (Out-of-order packets cause some ambiguity in the lost packet count; Iperf assumes they are not duplicate packets, so they are excluded from the lost packet count.) Since TCP does not report loss to the user, it can be noted that UDP tests are helpful to see packet loss along a path [59].

Jitter calculations are continuously computed by the server function, as specified by RTP in RFC 1889 [63]. The client function records a 64 bit second/microsecond timestamp in the packet. The server computes the relative transit time as (server's receive time - client's send time). The client's and server's clocks do not need to be synchronised; any difference is subtracted out in the jitter calculation. Jitter is the smoothed mean of differences between consecutive transit times [59].

The Iperf source code is listed below [58].

11.1.1 Iperf Server Source Code

```
/*---------------------------------------------------------------
 * Copyright (c) 1999,2000,2001,2002,2003
 * The Board of Trustees of the University of Illinois
 * All Rights Reserved.
 *---------------------------------------------------------------
 * Permission is hereby granted, free of charge, to any person
 * obtaining a copy of this software (Iperf) and associated
 * documentation files (the "Software"), to deal in the Software
 * without restriction, including without limitation the
 * rights to use, copy, modify, merge, publish, distribute,
 * sublicense, and/or sell copies of the Software, and to permit
 * persons to whom the Software is furnished to do
 * so, subject to the following conditions:
 *
 *
 * Redistributions of source code must retain the above
 * copyright notice, this list of conditions and
 * the following disclaimers.
 *
 *
 * Redistributions in binary form must reproduce the above
 * copyright notice, this list of conditions and the following
 * disclaimers in the documentation and/or other materials
 * provided with the distribution.
 *
 *
 * Neither the names of the University of Illinois, NCSA,
 * nor the names of its contributors may be used to endorse
 * or promote products derived from this Software without
 * specific prior written permission.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
 * OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
 * NONINFRINGEMENT. IN NO EVENT SHALL THE CONTIBUTORS OR COPYRIGHT
 * HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
 * WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
 * ARISING FROM, OUT OF OR IN CONNECTION WITH THE
 * SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
 * ________________________________________________________________
 * National Laboratory for Applied Network Research
 * National Center for Supercomputing Applications
 * University of Illinois at Urbana-Champaign
 * http://www.ncsa.uiuc.edu
 * ________________________________________________________________
 *
```

```
 * Server.cpp
 * by Mark Gates <mgates@nlanr.net>
 *     Ajay Tirumala (tirumala@ncsa.uiuc.edu>.
 * -------------------------------------------------------------------
 * A server thread is initiated for each connection accept() returns.
 * Handles sending and receiving data, and then closes socket.
 * Changes to this version : The server can be run as a daemon
 * ------------------------------------------------------------------- */

#define HEADERS()

#include "headers.h"
#include "Server.hpp"
#include "List.h"
#include "Extractor.h"
#include "Reporter.h"
#include "Locale.h"

/* -------------------------------------------------------------------
 * Stores connected socket and socket info.
 * ------------------------------------------------------------------- */

Server::Server( thread_Settings *inSettings ) {
    mSettings = inSettings;
    mBuf = NULL;

    // initialize buffer
    mBuf = new char[ mSettings->mBufLen ];
    FAIL_errno( mBuf == NULL, "No memory for buffer\n", mSettings );
}

/* -------------------------------------------------------------------
 * Destructor close socket.
 * ------------------------------------------------------------------- */

Server::~Server() {
    if ( mSettings->mSock != INVALID_SOCKET ) {
        int rc = close( mSettings->mSock );
        WARN_errno( rc == SOCKET_ERROR, "close" );
        mSettings->mSock = INVALID_SOCKET;
    }
    DELETE_ARRAY( mBuf );
}

void Server::Sig_Int( int inSigno ) {
}

/* -------------------------------------------------------------------
 * Receive data from the (connected) socket.
 * Sends termination flag several times at the end.
 * Does not close the socket.
 * ------------------------------------------------------------------- */
void Server::Run( void ) {
    long currLen;
    max_size_t totLen = 0;
    struct UDP_datagram* mBuf_UDP  = (struct UDP_datagram*) mBuf;

    ReportStruct *reportstruct = NULL;

    reportstruct = new ReportStruct;
    if ( reportstruct != NULL ) {
        reportstruct->packetID = 0;
        mSettings->reporthdr = InitReport( mSettings );
        do {
            // perform read
            currLen = recv( mSettings->mSock, mBuf, mSettings->mBufLen, 0 );

            if ( isUDP( mSettings ) ) {
```

```
                // read the datagram ID and sentTime out of the buffer
                reportstruct->packetID = ntohl( mBuf_UDP->id );
                reportstruct->sentTime.tv_sec = ntohl( mBuf_UDP->tv_sec  );
                reportstruct->sentTime.tv_usec = ntohl( mBuf_UDP->tv_usec );
              reportstruct->packetLen = currLen;
              gettimeofday( &(reportstruct->packetTime), NULL );
            } else {
              totLen += currLen;
           }

            // terminate when datagram begins with negative index
            // the datagram ID should be correct, just negated
            if ( reportstruct->packetID < 0 ) {
                reportstruct->packetID = -reportstruct->packetID;
                currLen = -1;
            }

           if ( isUDP (mSettings)) {
              ReportPacket( mSettings->reporthdr, reportstruct );
            } else if ( !isUDP (mSettings) && mSettings->mInterval > 0) {
                reportstruct->packetLen = currLen;
                gettimeofday( &(reportstruct->packetTime), NULL );
                ReportPacket( mSettings->reporthdr, reportstruct );
            }



        } while ( currLen > 0 );


        // stop timing
        gettimeofday( &(reportstruct->packetTime), NULL );

       if ( !isUDP (mSettings)) {
              if(0.0 == mSettings->mInterval) {
                        reportstruct->packetLen = totLen;
                }
              ReportPacket( mSettings->reporthdr, reportstruct );
       }
        CloseReport( mSettings->reporthdr, reportstruct );

        // send a acknowledgement back only if we're NOT receiving multicast
        if ( isUDP( mSettings ) && !isMulticast( mSettings ) ) {
            // send back an acknowledgement of the terminating datagram
            write_UDP_AckFIN( );
        }
    } else {
        FAIL(1, "Out of memory! Closing server thread\n", mSettings);
    }

    Mutex_Lock( &clients_mutex );
    Iperf_delete( &(mSettings->peer), &clients );
    Mutex_Unlock( &clients_mutex );

    DELETE_PTR( reportstruct );
    EndReport( mSettings->reporthdr );
}
// end Recv

/* -------------------------------------------------------------------
 * Send an AckFIN (a datagram acknowledging a FIN) on the socket,
 * then select on the socket for some time. If additional datagrams
 * come in, probably our AckFIN was lost and they are re-transmitted
 * termination datagrams, so re-transmit our AckFIN.
 * ------------------------------------------------------------------- */

void Server::write_UDP_AckFIN( ) {
```

```
    int rc;

    fd_set readSet;
    FD_ZERO( &readSet );

    struct timeval timeout;

    int count = 0;
    while ( count < 10 ) {
        count++;

        UDP_datagram *UDP_Hdr;
        server_hdr *hdr;

        UDP_Hdr = (UDP_datagram*) mBuf;

        if ( mSettings->mBufLen > (int) ( sizeof( UDP_datagram )
                                          + sizeof( server_hdr ) ) ) {
            Transfer_Info *stats = GetReport( mSettings->reporthdr );
            hdr = (server_hdr*) (UDP_Hdr+1);

            hdr->flags        = htonl( HEADER_VERSION1 );
            hdr->total_len1   = htonl( (long) (stats->TotalLen >> 32) );
            hdr->total_len2   = htonl( (long) (stats->TotalLen & 0xFFFFFFFF)
);
            hdr->stop_sec     = htonl( (long) stats->endTime );
            hdr->stop_usec    = htonl( (long)((stats->endTime - (long)stats-
>endTime)
                                              * rMillion));
            hdr->error_cnt    = htonl( stats->cntError );
            hdr->outorder_cnt = htonl( stats->cntOutofOrder );
            hdr->datagrams    = htonl( stats->cntDatagrams );
            hdr->jitter1      = htonl( (long) stats->jitter );
            hdr->jitter2      = htonl( (long) ((stats->jitter - (long)stats-
>jitter)
                                               * rMillion) );

        }

        // write data
        write( mSettings->mSock, mBuf, mSettings->mBufLen );

        // wait until the socket is readable, or our timeout expires
        FD_SET( mSettings->mSock, &readSet );
        timeout.tv_sec  = 1;
        timeout.tv_usec = 0;

        rc = select( mSettings->mSock+1, &readSet, NULL, NULL, &timeout );
        FAIL_errno( rc == SOCKET_ERROR, "select", mSettings );

        if ( rc == 0 ) {
            // select timed out
            return;
        } else {
            // socket ready to read
            rc = read( mSettings->mSock, mBuf, mSettings->mBufLen );
            WARN_errno( rc < 0, "read" );
            if ( rc <= 0 ) {
                // Connection closed or errored
                // Stop using it.
                return;
            }
        }
    }

    fprintf( stderr, warn_ack_failed, mSettings->mSock, count );
}
// end write_UDP_AckFIN
```

11.1.2 Iperf Client Source Code

```
/*---------------------------------------------------------------
 * Copyright (c) 1999,2000,2001,2002,2003
 * The Board of Trustees of the University of Illinois
 * All Rights Reserved.
 *---------------------------------------------------------------
 * Permission is hereby granted, free of charge, to any person
 * obtaining a copy of this software (Iperf) and associated
 * documentation files (the "Software"), to deal in the Software
 * without restriction, including without limitation the
 * rights to use, copy, modify, merge, publish, distribute,
 * sublicense, and/or sell copies of the Software, and to permit
 * persons to whom the Software is furnished to do
 * so, subject to the following conditions:
 *
 *
 * Redistributions of source code must retain the above
 * copyright notice, this list of conditions and
 * the following disclaimers.
 *
 *
 * Redistributions in binary form must reproduce the above
 * copyright notice, this list of conditions and the following
 * disclaimers in the documentation and/or other materials
 * provided with the distribution.
 *
 *
 * Neither the names of the University of Illinois, NCSA,
 * nor the names of its contributors may be used to endorse
 * or promote products derived from this Software without
 * specific prior written permission.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
 * OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
 * NONINFRINGEMENT. IN NO EVENT SHALL THE CONTIBUTORS OR COPYRIGHT
 * HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
 * WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
 * ARISING FROM, OUT OF OR IN CONNECTION WITH THE
 * SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
 * ________________________________________________________________
 * National Laboratory for Applied Network Research
 * National Center for Supercomputing Applications
 * University of Illinois at Urbana-Champaign
 * http://www.ncsa.uiuc.edu
 * ________________________________________________________________
 *
 * Client.cpp
 * by Mark Gates <mgates@nlanr.net>
 * -------------------------------------------------------------------
 * A client thread initiates a connect to the server and handles
 * sending and receiving data, then closes the socket.
 * ------------------------------------------------------------------- */

#include "headers.h"
#include "Client.hpp"
#include "Thread.h"
#include "SocketAddr.h"
#include "PerfSocket.hpp"
#include "Extractor.h"
#include "delay.hpp"
#include "util.h"
#include "Locale.h"

/* -------------------------------------------------------------------
```

```
 * Store server hostname, optionally local hostname, and socket info.
 * ------------------------------------------------------------------- */

Client::Client( thread_Settings *inSettings ) {
    mSettings = inSettings;
    mBuf = NULL;

    // initialize buffer
    mBuf = new char[ mSettings->mBufLen ];
    pattern( mBuf, mSettings->mBufLen );
    if ( isFileInput( mSettings ) ) {
        if ( !isSTDIN( mSettings ) )
            Extractor_Initialize( mSettings->mFileName, mSettings->mBufLen,
mSettings );
        else
            Extractor_InitializeFile( stdin, mSettings->mBufLen, mSettings
);

        if ( !Extractor_canRead( mSettings ) ) {
            unsetFileInput( mSettings );
        }
    }

    // connect
    Connect( );

    if ( isReport( inSettings ) ) {
        ReportSettings( inSettings );
        if ( mSettings->multihdr && isMultipleReport( inSettings ) ) {
            mSettings->multihdr->report->connection.peer = mSettings->peer;
            mSettings->multihdr->report->connection.size_peer = mSettings-
>size_peer;
            mSettings->multihdr->report->connection.local = mSettings-
>local;
            SockAddr_setPortAny( &mSettings->multihdr->report-
>connection.local );
            mSettings->multihdr->report->connection.size_local = mSettings-
>size_local;
        }
    }

} // end Client

/* -------------------------------------------------------------------
 * Delete memory (hostname strings).
 * ------------------------------------------------------------------- */

Client::~Client() {
    if ( mSettings->mSock != INVALID_SOCKET ) {
        int rc = close( mSettings->mSock );
        WARN_errno( rc == SOCKET_ERROR, "close" );
        mSettings->mSock = INVALID_SOCKET;
    }
    DELETE_ARRAY( mBuf );
} // end ~Client

const double kSecs_to_usecs = 1e6;
const int    kBytes_to_Bits = 8;

void Client::RunTCP( void ) {
    unsigned long currLen = 0;
    struct itimerval it;
    max_size_t totLen = 0;

    int err;

    char* readAt = mBuf;
```

```
    // Indicates if the stream is readable
    bool canRead = true, mMode_Time = isModeTime( mSettings );

    ReportStruct *reportstruct = NULL;

    // InitReport handles Barrier for multiple Streams
    mSettings->reporthdr = InitReport( mSettings );
    reportstruct = new ReportStruct;
    reportstruct->packetID = 0;

    lastPacketTime.setnow();
    if ( mMode_Time ) {
       memset (&it, 0, sizeof (it));
       it.it_value.tv_sec = (int) (mSettings->mAmount / 100.0);
       it.it_value.tv_usec = (int) 10000 * (mSettings->mAmount -
           it.it_value.tv_sec * 100.0);
       err = setitimer( ITIMER_REAL, &it, NULL );
       if ( err != 0 ) {
           perror("setitimer");
           exit(1);
       }
    }
    do {
        // Read the next data block from
        // the file if it's file input
        if ( isFileInput( mSettings ) ) {
            Extractor_getNextDataBlock( readAt, mSettings );
            canRead = Extractor_canRead( mSettings ) != 0;
        } else
            canRead = true;

        // perform write
        currLen = write( mSettings->mSock, mBuf, mSettings->mBufLen );
        if ( currLen < 0 ) {
            WARN_errno( currLen < 0, "write2" );
            break;
        }
       totLen += currLen;

       if(mSettings->mInterval > 0) {
           gettimeofday( &(reportstruct->packetTime), NULL );
            reportstruct->packetLen = currLen;
            ReportPacket( mSettings->reporthdr, reportstruct );
        }

        if ( !mMode_Time ) {
            /* mAmount may be unsigned, so don't let it underflow! */
            if( mSettings->mAmount >= currLen ) {
                mSettings->mAmount -= currLen;
            } else {
                mSettings->mAmount = 0;
            }
        }

    } while ( ! (sInterupted  ||
                   (!mMode_Time  &&  0 >= mSettings->mAmount)) && canRead );

    // stop timing
    gettimeofday( &(reportstruct->packetTime), NULL );

    // if we're not doing interval reporting, report the entire transfer as
one big packet
    if(0.0 == mSettings->mInterval) {
        reportstruct->packetLen = totLen;
        ReportPacket( mSettings->reporthdr, reportstruct );
    }
    CloseReport( mSettings->reporthdr, reportstruct );
```

```
    DELETE_PTR( reportstruct );
    EndReport( mSettings->reporthdr );
}

/* -------------------------------------------------------------------
 * Send data using the connected UDP/TCP socket,
 * until a termination flag is reached.
 * Does not close the socket.
 * ------------------------------------------------------------------- */

void Client::Run( void ) {
    struct UDP_datagram* mBuf_UDP = (struct UDP_datagram*) mBuf;
    unsigned long currLen = 0;

    int delay_target = 0;
    int delay = 0;
    int adjust = 0;

    char* readAt = mBuf;

#if HAVE_THREAD
    if ( !isUDP( mSettings ) ) {
       RunTCP();
       return;
    }
#endif

    // Indicates if the stream is readable
    bool canRead = true, mMode_Time = isModeTime( mSettings );

    // setup termination variables
    if ( mMode_Time ) {
        mEndTime.setnow();
        mEndTime.add( mSettings->mAmount / 100.0 );
    }

    if ( isUDP( mSettings ) ) {
        // Due to the UDP timestamps etc, included
        // reduce the read size by an amount
        // equal to the header size

        // compute delay for bandwidth restriction, constrained to [0,1]
seconds
        delay_target = (int) ( mSettings->mBufLen * ((kSecs_to_usecs *
kBytes_to_Bits)
                                                     / mSettings->mUDPRate)
);
        if ( delay_target < 0  ||
             delay_target > (int) 1 * kSecs_to_usecs ) {
            fprintf( stderr, warn_delay_large, delay_target / kSecs_to_usecs
);
            delay_target = (int) kSecs_to_usecs * 1;
        }
        if ( isFileInput( mSettings ) ) {
            if ( isCompat( mSettings ) ) {
                Extractor_reduceReadSize( sizeof(struct UDP_datagram),
mSettings );
                readAt += sizeof(struct UDP_datagram);
            } else {
                Extractor_reduceReadSize( sizeof(struct UDP_datagram) +
                                          sizeof(struct client_hdr),
mSettings );
                readAt += sizeof(struct UDP_datagram) +
                          sizeof(struct client_hdr);
            }
        }
    }
```

```
    ReportStruct *reportstruct = NULL;

    // InitReport handles Barrier for multiple Streams
    mSettings->reporthdr = InitReport( mSettings );
    reportstruct = new ReportStruct;
    reportstruct->packetID = 0;

    lastPacketTime.setnow();

    do {

        // Test case: drop 17 packets and send 2 out-of-order:
        // sequence 51, 52, 70, 53, 54, 71, 72
        //switch( datagramID ) {
        //  case 53: datagramID = 70; break;
        //  case 71: datagramID = 53; break;
        //  case 55: datagramID = 71; break;
        //  default: break;
        //}
        gettimeofday( &(reportstruct->packetTime), NULL );

        if ( isUDP( mSettings ) ) {
            // store datagram ID into buffer
            mBuf_UDP->id      = htonl( (reportstruct->packetID)++ );
            mBuf_UDP->tv_sec  = htonl( reportstruct->packetTime.tv_sec );
            mBuf_UDP->tv_usec = htonl( reportstruct->packetTime.tv_usec );

            // delay between writes
            // make an adjustment for how long the last loop iteration took
            // TODO this doesn't work well in certain cases, like 2 parallel
streams
            adjust = delay_target + lastPacketTime.subUsec( reportstruct-
>packetTime );
            lastPacketTime.set( reportstruct->packetTime.tv_sec,
                                reportstruct->packetTime.tv_usec );

            if ( adjust > 0  ||  delay > 0 ) {
                delay += adjust;
            }
        }

        // Read the next data block from
        // the file if it's file input
        if ( isFileInput( mSettings ) ) {
            Extractor_getNextDataBlock( readAt, mSettings );
            canRead = Extractor_canRead( mSettings ) != 0;
        } else
            canRead = true;

        // perform write
        currLen = write( mSettings->mSock, mBuf, mSettings->mBufLen );
        if ( currLen < 0 && errno != ENOBUFS ) {
            WARN_errno( currLen < 0, "write2" );
            break;
        }

        // report packets
        reportstruct->packetLen = currLen;
        ReportPacket( mSettings->reporthdr, reportstruct );

        if ( delay > 0 ) {
            delay_loop( delay );
        }
        if ( !mMode_Time ) {
            /* mAmount may be unsigned, so don't let it underflow! */
            if( mSettings->mAmount >= currLen ) {
                mSettings->mAmount -= currLen;
            } else {
```

```
                mSettings->mAmount = 0;
            }
        }

    } while ( ! (sInterupted  ||
                 (mMode_Time   &&  mEndTime.before( reportstruct->packetTime
))  ||
                 (!mMode_Time  &&  0 >= mSettings->mAmount)) && canRead );

    // stop timing
    gettimeofday( &(reportstruct->packetTime), NULL );
    CloseReport( mSettings->reporthdr, reportstruct );

    if ( isUDP( mSettings ) ) {
        // send a final terminating datagram
        // Don't count in the mTotalLen. The server counts this one,
        // but didn't count our first datagram, so we're even now.
        // The negative datagram ID signifies termination to the server.

        // store datagram ID into buffer
        mBuf_UDP->id      = htonl( -(reportstruct->packetID)  );
        mBuf_UDP->tv_sec  = htonl( reportstruct->packetTime.tv_sec );
        mBuf_UDP->tv_usec = htonl( reportstruct->packetTime.tv_usec );

        if ( isMulticast( mSettings ) ) {
            write( mSettings->mSock, mBuf, mSettings->mBufLen );
        } else {
            write_UDP_FIN( );
        }
    }
    DELETE_PTR( reportstruct );
    EndReport( mSettings->reporthdr );
}
// end Run

void Client::InitiateServer() {
    if ( !isCompat( mSettings ) ) {
        int currLen;
        client_hdr* temp_hdr;
        if ( isUDP( mSettings ) ) {
            UDP_datagram *UDPhdr = (UDP_datagram *)mBuf;
            temp_hdr = (client_hdr*)(UDPhdr + 1);
        } else {
            temp_hdr = (client_hdr*)mBuf;
        }
        Settings_GenerateClientHdr( mSettings, temp_hdr );
        if ( !isUDP( mSettings ) ) {
            currLen = send( mSettings->mSock, mBuf, sizeof(client_hdr), 0 );
            if ( currLen < 0 ) {
                WARN_errno( currLen < 0, "write1" );
            }
        }
    }
}

/* -------------------------------------------------------------------
 * Setup a socket connected to a server.
 * If inLocalhost is not null, bind to that address, specifying
 * which outgoing interface to use.
 * ------------------------------------------------------------------- */

void Client::Connect( ) {
    int rc;
    SockAddr_remoteAddr( mSettings );

    assert( mSettings->inHostname != NULL );

    // create an internet socket
```

```
    int type = ( isUDP( mSettings )  ?  SOCK_DGRAM : SOCK_STREAM);

    int domain = (SockAddr_isIPv6( &mSettings->peer ) ?
#ifdef HAVE_IPV6
                  AF_INET6
#else
                  AF_INET
#endif
                  : AF_INET);

    mSettings->mSock = socket( domain, type, 0 );
    WARN_errno( mSettings->mSock == INVALID_SOCKET, "socket" );

    SetSocketOptions( mSettings );


    SockAddr_localAddr( mSettings );
    if ( mSettings->mLocalhost != NULL ) {
        // bind socket to local address
        rc = bind( mSettings->mSock, (sockaddr*) &mSettings->local,
                   SockAddr_get_sizeof_sockaddr( &mSettings->local ) );
        WARN_errno( rc == SOCKET_ERROR, "bind" );
    }

    // connect socket
    rc = connect( mSettings->mSock, (sockaddr*) &mSettings->peer,
                  SockAddr_get_sizeof_sockaddr( &mSettings->peer ));
    FAIL_errno( rc == SOCKET_ERROR, "connect", mSettings );

    getsockname( mSettings->mSock, (sockaddr*) &mSettings->local,
                 &mSettings->size_local );
    getpeername( mSettings->mSock, (sockaddr*) &mSettings->peer,
                 &mSettings->size_peer );
} // end Connect

/* -------------------------------------------------------------------
 * Send a datagram on the socket. The datagram's contents should signify
 * a FIN to the application. Keep re-transmitting until an
 * acknowledgement datagram is received.
 * ------------------------------------------------------------------- */

void Client::write_UDP_FIN( ) {
    int rc;
    fd_set readSet;
    struct timeval timeout;

    int count = 0;
    while ( count < 10 ) {
        count++;

        // write data
        write( mSettings->mSock, mBuf, mSettings->mBufLen );

        // wait until the socket is readable, or our timeout expires
        FD_ZERO( &readSet );
        FD_SET( mSettings->mSock, &readSet );
        timeout.tv_sec  = 0;
        timeout.tv_usec = 250000; // quarter second, 250 ms

        rc = select( mSettings->mSock+1, &readSet, NULL, NULL, &timeout );
        FAIL_errno( rc == SOCKET_ERROR, "select", mSettings );

        if ( rc == 0 ) {
            // select timed out
            continue;
        } else {
            // socket ready to read
            rc = read( mSettings->mSock, mBuf, mSettings->mBufLen );
```

```
            WARN_errno( rc < 0, "read" );
           if ( rc < 0 ) {
                break;
            } else if ( rc >= (int) (sizeof(UDP_datagram) +
sizeof(server_hdr)) ) {
                ReportServerUDP( mSettings, (server_hdr*)
((UDP_datagram*)mBuf + 1) );
            }

            return;
        }
    }

    fprintf( stderr, warn_no_ack, mSettings->mSock, count );
}
// end write_UDP_FIN
```

11.1.3 Iperf Listener Source Code

```
/*---------------------------------------------------------------
 * Copyright (c) 1999,2000,2001,2002,2003
 * The Board of Trustees of the University of Illinois
 * All Rights Reserved.
 *---------------------------------------------------------------
 * Permission is hereby granted, free of charge, to any person
 * obtaining a copy of this software (Iperf) and associated
 * documentation files (the "Software"), to deal in the Software
 * without restriction, including without limitation the
 * rights to use, copy, modify, merge, publish, distribute,
 * sublicense, and/or sell copies of the Software, and to permit
 * persons to whom the Software is furnished to do
 * so, subject to the following conditions:
 *
 *
 * Redistributions of source code must retain the above
 * copyright notice, this list of conditions and
 * the following disclaimers.
 *
 *
 * Redistributions in binary form must reproduce the above
 * copyright notice, this list of conditions and the following
 * disclaimers in the documentation and/or other materials
 * provided with the distribution.
 *
 *
 * Neither the names of the University of Illinois, NCSA,
 * nor the names of its contributors may be used to endorse
 * or promote products derived from this Software without
 * specific prior written permission.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
 * OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
 * NONINFRINGEMENT. IN NO EVENT SHALL THE CONTIBUTORS OR COPYRIGHT
 * HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
 * WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
 * ARISING FROM, OUT OF OR IN CONNECTION WITH THE
 * SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
 * ________________________________________________________________
 * National Laboratory for Applied Network Research
 * National Center for Supercomputing Applications
 * University of Illinois at Urbana-Champaign
 * http://www.ncsa.uiuc.edu
 * ________________________________________________________________
 *
```

```
 * Listener.cpp
 * by Mark Gates <mgates@nlanr.net>
 * &  Ajay Tirumala <tirumala@ncsa.uiuc.edu>
 * -------------------------------------------------------------------
 * Listener sets up a socket listening on the server host. For each
 * connected socket that accept() returns, this creates a Server
 * socket and spawns a thread for it.
 *
 * Changes to the latest version. Listener will run as a daemon
 * Multicast Server is now Multi-threaded
 * -------------------------------------------------------------------
 * headers
 * uses
 *   <stdlib.h>
 *   <stdio.h>
 *   <string.h>
 *   <errno.h>
 *
 *   <sys/types.h>
 *   <unistd.h>
 *
 *   <netdb.h>
 *   <netinet/in.h>
 *   <sys/socket.h>
 * ------------------------------------------------------------------- */


#define HEADERS()

#include "headers.h"
#include "Listener.hpp"
#include "SocketAddr.h"
#include "PerfSocket.hpp"
#include "List.h"
#include "util.h"

/* -------------------------------------------------------------------
 * Stores local hostname and socket info.
 * ------------------------------------------------------------------- */

Listener::Listener( thread_Settings *inSettings ) {

    mClients = inSettings->mThreads;
    mBuf = NULL;
    mSettings = inSettings;

    // initialize buffer
    mBuf = new char[ mSettings->mBufLen ];

    // open listening socket
    Listen( );
    ReportSettings( inSettings );

} // end Listener

/* -------------------------------------------------------------------
 * Delete memory (buffer).
 * ------------------------------------------------------------------- */
Listener::~Listener() {
    if ( mSettings->mSock != INVALID_SOCKET ) {
        int rc = close( mSettings->mSock );
        WARN_errno( rc == SOCKET_ERROR, "close" );
        mSettings->mSock = INVALID_SOCKET;
    }
    DELETE_ARRAY( mBuf );
} // end ~Listener

/* -------------------------------------------------------------------
```

```
 * Listens for connections and starts Servers to handle data.
 * For TCP, each accepted connection spawns a Server thread.
 * For UDP, handle all data in this thread for Win32 Only, otherwise
 *          spawn a new Server thread.
 * ------------------------------------------------------------------- */
void Listener::Run( void ) {
#ifdef WIN32
    if ( isUDP( mSettings ) && !isSingleUDP( mSettings ) ) {
        UDPSingleServer();
    } else
#else
#ifdef sun
    if ( ( isUDP( mSettings ) &&
           isMulticast( mSettings ) &&
           !isSingleUDP( mSettings ) ) ||
         isSingleUDP( mSettings ) ) {
        UDPSingleServer();
    } else
#else
    if ( isSingleUDP( mSettings ) ) {
        UDPSingleServer();
    } else
#endif
#endif
    {
        bool client = false, UDP = isUDP( mSettings ), mCount = (mSettings-
>mThreads != 0);
        thread_Settings *tempSettings = NULL;
        Iperf_ListEntry *exist, *listtemp;
        client_hdr* hdr = ( UDP ? (client_hdr*) (((UDP_datagram*)mBuf) + 1)
:
                                  (client_hdr*) mBuf);

        if ( mSettings->mHost != NULL ) {
            client = true;
            SockAddr_remoteAddr( mSettings );
        }
        Settings_Copy( mSettings, &server );
        server->mThreadMode = kMode_Server;


        // Accept each packet,
        // If there is no existing client, then start
        // a new thread to service the new client
        // The listener runs in a single thread
        // Thread per client model is followed
        do {
            // Get a new socket
            Accept( server );
            if ( server->mSock == INVALID_SOCKET ) {
                break;
            }
            if ( sInterupted != 0 ) {
                close( server->mSock );
                break;
            }
            // Reset Single Client Stuff
            if ( isSingleClient( mSettings ) && clients == NULL ) {
                mSettings->peer = server->peer;
                mClients--;
                client = true;
                // Once all the server threads exit then quit
                // Must keep going in case this client sends
                // more streams
                if ( mClients == 0 ) {
                    thread_release_nonterm( 0 );
                    mClients = 1;
                }
```

```
            }
            // Verify that it is allowed
            if ( client ) {
                if ( !SockAddr_Hostare_Equal( (sockaddr*) &mSettings->peer,
                                              (sockaddr*) &server->peer ) )
{
                    // Not allowed try again
                    close( server->mSock );
                    if ( isUDP( mSettings ) ) {
                        mSettings->mSock = -1;
                        Listen();
                    }
                    continue;
                }
            }

            // Create an entry for the connection list
            listtemp = new Iperf_ListEntry;
            memcpy(listtemp, &server->peer, sizeof(iperf_sockaddr));
            listtemp->next = NULL;

            // See if we need to do summing
            Mutex_Lock( &clients_mutex );
            exist = Iperf_hostpresent( &server->peer, clients);

            if ( exist != NULL ) {
                // Copy group ID
                listtemp->holder = exist->holder;
                server->multihdr = exist->holder;
            } else {
                server->mThreads = 0;
                Mutex_Lock( &groupCond );
                groupID--;
                listtemp->holder = InitMulti( server, groupID );
                server->multihdr = listtemp->holder;
                Mutex_Unlock( &groupCond );
            }

            // Store entry in connection list
            Iperf_pushback( listtemp, &clients );
            Mutex_Unlock( &clients_mutex );

            tempSettings = NULL;
            if ( !isCompat( mSettings ) && !isMulticast( mSettings ) ) {
                if ( !UDP ) {
                    // TCP does not have the info yet
                    if ( recv( server->mSock, (char*)hdr,
sizeof(client_hdr), 0) > 0 ) {
                        Settings_GenerateClientSettings( server,
&tempSettings,
                                                          hdr );
                    }
                } else {
                    Settings_GenerateClientSettings( server, &tempSettings,
                                                     hdr );
                }
            }


            if ( tempSettings != NULL ) {
                client_init( tempSettings );
                if ( tempSettings->mMode == kTest_DualTest ) {
#ifdef HAVE_THREAD
                    server->runNow =  tempSettings;
#else
                    server->runNext = tempSettings;
#endif
                } else {
```

```
                    server->runNext =  tempSettings;
                }
            }

            // Start the server
#if defined(WIN32) && defined(HAVE_THREAD)
            if ( UDP ) {
                // WIN32 does bad UDP handling so run single threaded
                if ( server->runNow != NULL ) {
                    thread_start( server->runNow );
                }
                server_spawn( server );
                if ( server->runNext != NULL ) {
                    thread_start( server->runNext );
                }
            } else
#endif
            thread_start( server );

            // create a new socket
            if ( UDP ) {
                mSettings->mSock = -1;
                Listen( );
            }

            // Prep for next connection
            if ( !isSingleClient( mSettings ) ) {
                mClients--;
            }
            Settings_Copy( mSettings, &server );
            server->mThreadMode = kMode_Server;
        } while ( !sInterupted && (!mCount || ( mCount && mClients > 0 )) );

        Settings_Destroy( server );
    }
} // end Run

/* -------------------------------------------------------------------
 * Setup a socket listening on a port.
 * For TCP, this calls bind() and listen().
 * For UDP, this just calls bind().
 * If inLocalhost is not null, bind to that address rather than the
 * wildcard server address, specifying what incoming interface to
 * accept connections on.
 * ------------------------------------------------------------------- */
void Listener::Listen( ) {
    int rc;

    SockAddr_localAddr( mSettings );

    // create an internet TCP socket
    int type = (isUDP( mSettings )  ?  SOCK_DGRAM  :  SOCK_STREAM);
    int domain = (SockAddr_isIPv6( &mSettings->local ) ?
#ifdef HAVE_IPV6
                  AF_INET6
#else
                  AF_INET
#endif
                  : AF_INET);

#ifdef WIN32
    if ( SockAddr_isMulticast( &mSettings->local ) ) {
        // Multicast on Win32 requires special handling
        mSettings->mSock = WSASocket( domain, type, 0, 0, 0,
WSA_FLAG_MULTIPOINT_C_LEAF | WSA_FLAG_MULTIPOINT_D_LEAF );
        WARN_errno( mSettings->mSock == INVALID_SOCKET, "socket" );

    } else
```

```
#endif
    {
        mSettings->mSock = socket( domain, type, 0 );
        WARN_errno( mSettings->mSock == INVALID_SOCKET, "socket" );
    }

    SetSocketOptions( mSettings );

    // reuse the address, so we can run if a former server was killed off
    int boolean = 1;
    Socklen_t len = sizeof(boolean);
    setsockopt( mSettings->mSock, SOL_SOCKET, SO_REUSEADDR, (char*)
&boolean, len );

    // bind socket to server address
#ifdef WIN32
    if ( SockAddr_isMulticast( &mSettings->local ) ) {
        // Multicast on Win32 requires special handling
        rc = WSAJoinLeaf( mSettings->mSock, (sockaddr*) &mSettings->local,
mSettings->size_local,0,0,0,0,JL_BOTH);
        WARN_errno( rc == SOCKET_ERROR, "WSAJoinLeaf (aka bind)" );
    } else
#endif
    {
        rc = bind( mSettings->mSock, (sockaddr*) &mSettings->local,
mSettings->size_local );
        WARN_errno( rc == SOCKET_ERROR, "bind" );
    }
    // listen for connections (TCP only).
    // default backlog traditionally 5
    if ( !isUDP( mSettings ) ) {
        rc = listen( mSettings->mSock, 5 );
        WARN_errno( rc == SOCKET_ERROR, "listen" );
    }

#ifndef WIN32
    // if multicast, join the group
    if ( SockAddr_isMulticast( &mSettings->local ) ) {
        McastJoin( );
    }
#endif
} // end Listen

/* -------------------------------------------------------------------
 * Joins the multicast group, with the default interface.
 * ------------------------------------------------------------------- */

void Listener::McastJoin( ) {
#ifdef HAVE_MULTICAST
    if ( !SockAddr_isIPv6( &mSettings->local ) ) {
        struct ip_mreq mreq;

        memcpy( &mreq.imr_multiaddr, SockAddr_get_in_addr( &mSettings->local
),
                sizeof(mreq.imr_multiaddr));

        mreq.imr_interface.s_addr = htonl( INADDR_ANY );

        int rc = setsockopt( mSettings->mSock, IPPROTO_IP,
IP_ADD_MEMBERSHIP,
                             (char*) &mreq, sizeof(mreq));
        WARN_errno( rc == SOCKET_ERROR, "multicast join" );
    }
#ifdef HAVE_IPV6_MULTICAST
      else {
        struct ipv6_mreq mreq;
```

```
        memcpy( &mreq.ipv6mr_multiaddr, SockAddr_get_in6_addr( &mSettings-
>local ),
                sizeof(mreq.ipv6mr_multiaddr));

        mreq.ipv6mr_interface = 0;

        int rc = setsockopt( mSettings->mSock, IPPROTO_IPV6,
IPV6_ADD_MEMBERSHIP,
                             (char*) &mreq, sizeof(mreq));
        WARN_errno( rc == SOCKET_ERROR, "multicast join" );
    }
#endif
#endif
}
// end McastJoin

/* -------------------------------------------------------------------
 * Sets the Multicast TTL for outgoing packets.
 * ------------------------------------------------------------------- */

void Listener::McastSetTTL( int val ) {
#ifdef HAVE_MULTICAST
    if ( !SockAddr_isIPv6( &mSettings->local ) ) {
        int rc = setsockopt( mSettings->mSock, IPPROTO_IP, IP_MULTICAST_TTL,
                             (char*) &val, sizeof(val));
        WARN_errno( rc == SOCKET_ERROR, "multicast ttl" );
    }
#ifdef HAVE_IPV6_MULTICAST
      else {
        int rc = setsockopt( mSettings->mSock, IPPROTO_IPV6,
IPV6_MULTICAST_HOPS,
                             (char*) &val, sizeof(val));
        WARN_errno( rc == SOCKET_ERROR, "multicast ttl" );
    }
#endif
#endif
}
// end McastSetTTL

/* -------------------------------------------------------------------
 * After Listen() has setup mSock, this will block
 * until a new connection arrives.
 * ------------------------------------------------------------------- */

void Listener::Accept( thread_Settings *server ) {

    server->size_peer = sizeof(iperf_sockaddr);
    if ( isUDP( server ) ) {
        /* -------------------------------------------------------------------
--
         * Do the equivalent of an accept() call for UDP sockets. This waits
         * on a listening UDP socket until we get a datagram.
         * -------------------------------------------------------------------
-- */
        int rc;
        Iperf_ListEntry *exist;
        int32_t datagramID;
        server->mSock = INVALID_SOCKET;
        while ( server->mSock == INVALID_SOCKET ) {
            rc = recvfrom( mSettings->mSock, mBuf, mSettings->mBufLen, 0,
                           (struct sockaddr*) &server->peer, &server-
>size_peer );
            FAIL_errno( rc == SOCKET_ERROR, "recvfrom", mSettings );

            Mutex_Lock( &clients_mutex );

            // Handle connection for UDP sockets.
            exist = Iperf_present( &server->peer, clients);
```

```
            datagramID = ntohl( ((UDP_datagram*) mBuf)->id );
            if ( exist == NULL && datagramID >= 0 ) {
                server->mSock = mSettings->mSock;
                int rc = connect( server->mSock, (struct sockaddr*) &server-
>peer,
                                  server->size_peer );
                FAIL_errno( rc == SOCKET_ERROR, "connect UDP", mSettings );
            } else {
                server->mSock = INVALID_SOCKET;
            }
            Mutex_Unlock( &clients_mutex );
        }
    } else {
        // Handles interupted accepts. Returns the newly connected socket.
        server->mSock = INVALID_SOCKET;

        while ( server->mSock == INVALID_SOCKET ) {
            // accept a connection
            server->mSock = accept( mSettings->mSock,
                                    (sockaddr*) &server->peer, &server-
>size_peer );
            if ( server->mSock == INVALID_SOCKET &&  errno == EINTR ) {
                continue;
            }
        }
    }
    server->size_local = sizeof(iperf_sockaddr);
    getsockname( server->mSock, (sockaddr*) &server->local,
                 &server->size_local );
} // end Accept

void Listener::UDPSingleServer( ) {

    bool client = false, UDP = isUDP( mSettings ), mCount = (mSettings-
>mThreads != 0);
    thread_Settings *tempSettings = NULL;
    Iperf_ListEntry *exist, *listtemp;
    int rc;
    int32_t datagramID;
    client_hdr* hdr = ( UDP ? (client_hdr*) (((UDP_datagram*)mBuf) + 1) :
                              (client_hdr*) mBuf);
    ReportStruct *reportstruct = new ReportStruct;

    if ( mSettings->mHost != NULL ) {
        client = true;
        SockAddr_remoteAddr( mSettings );
    }
    Settings_Copy( mSettings, &server );
    server->mThreadMode = kMode_Server;


    // Accept each packet,
    // If there is no existing client, then start
    // a new report to service the new client
    // The listener runs in a single thread
    Mutex_Lock( &clients_mutex );
    do {
        // Get next packet
        while ( sInterupted == 0) {
            server->size_peer = sizeof( iperf_sockaddr );
            rc = recvfrom( mSettings->mSock, mBuf, mSettings->mBufLen, 0,
                           (struct sockaddr*) &server->peer, &server-
>size_peer );
            WARN_errno( rc == SOCKET_ERROR, "recvfrom" );
            if ( rc == SOCKET_ERROR ) {
                return;
            }
```

```
            // Handle connection for UDP sockets.
            exist = Iperf_present( &server->peer, clients);
            datagramID = ntohl( ((UDP_datagram*) mBuf)->id );
            if ( datagramID >= 0 ) {
                if ( exist != NULL ) {
                    // read the datagram ID and sentTime out of the buffer
                    reportstruct->packetID = datagramID;
                    reportstruct->sentTime.tv_sec = ntohl( ((UDP_datagram*)
mBuf)->tv_sec  );
                    reportstruct->sentTime.tv_usec = ntohl( ((UDP_datagram*)
mBuf)->tv_usec );

                    reportstruct->packetLen = rc;
                    gettimeofday( &(reportstruct->packetTime), NULL );

                    ReportPacket( exist->server->reporthdr, reportstruct );
                } else {
                    Mutex_Lock( &groupCond );
                    groupID--;
                    server->mSock = -groupID;
                    Mutex_Unlock( &groupCond );
                    server->size_local = sizeof(iperf_sockaddr);
                    getsockname( mSettings->mSock, (sockaddr*) &server-
>local,
                                 &server->size_local );
                    break;
                }
            } else {
                if ( exist != NULL ) {
                    // read the datagram ID and sentTime out of the buffer
                    reportstruct->packetID = -datagramID;
                    reportstruct->sentTime.tv_sec = ntohl( ((UDP_datagram*)
mBuf)->tv_sec  );
                    reportstruct->sentTime.tv_usec = ntohl( ((UDP_datagram*)
mBuf)->tv_usec );

                    reportstruct->packetLen = rc;
                    gettimeofday( &(reportstruct->packetTime), NULL );

                    ReportPacket( exist->server->reporthdr, reportstruct );
                    // stop timing
                    gettimeofday( &(reportstruct->packetTime), NULL );
                    CloseReport( exist->server->reporthdr, reportstruct );

                    if ( rc > (int) ( sizeof( UDP_datagram )
                                      + sizeof( server_hdr )
) ) {
                        UDP_datagram *UDP_Hdr;
                        server_hdr *hdr;

                        UDP_Hdr = (UDP_datagram*) mBuf;
                        Transfer_Info *stats = GetReport( exist->server-
>reporthdr );
                        hdr = (server_hdr*) (UDP_Hdr+1);

                        hdr->flags        = htonl( HEADER_VERSION1 );
                        hdr->total_len1   = htonl( (long) (stats->TotalLen
>> 32) );
                        hdr->total_len2   = htonl( (long) (stats->TotalLen &
0xFFFFFFFF) );
                        hdr->stop_sec     = htonl( (long) stats->endTime );
                        hdr->stop_usec    = htonl( (long)((stats->endTime -
(long)stats->endTime)
                                                  * rMillion));
                        hdr->error_cnt    = htonl( stats->cntError );
                        hdr->outorder_cnt = htonl( stats->cntOutofOrder );
                        hdr->datagrams    = htonl( stats->cntDatagrams );
```

```
                        hdr->jitter1      = htonl( (long) stats->jitter );
                        hdr->jitter2      = htonl( (long) ((stats->jitter -
(long)stats->jitter)
                                                            * rMillion) );

                    }
                    EndReport( exist->server->reporthdr );
                    exist->server->reporthdr = NULL;
                    Iperf_delete( &(exist->server->peer), &clients );
                } else if ( rc > (int) ( sizeof( UDP_datagram )
                                                  + sizeof( server_hdr ) ) )
{
                    UDP_datagram *UDP_Hdr;
                    server_hdr *hdr;

                    UDP_Hdr = (UDP_datagram*) mBuf;
                    hdr = (server_hdr*) (UDP_Hdr+1);
                    hdr->flags = htonl( 0 );
                }
                sendto( mSettings->mSock, mBuf, mSettings->mBufLen, 0,
                        (struct sockaddr*) &server->peer, server-
>size_peer);
            }
        }
        if ( server->mSock == INVALID_SOCKET ) {
            break;
        }
        if ( sInterupted != 0 ) {
            close( server->mSock );
            break;
        }
        // Reset Single Client Stuff
        if ( isSingleClient( mSettings ) && clients == NULL ) {
            mSettings->peer = server->peer;
            mClients--;
            client = true;
            // Once all the server threads exit then quit
            // Must keep going in case this client sends
            // more streams
            if ( mClients == 0 ) {
                thread_release_nonterm( 0 );
                mClients = 1;
            }
        }
        // Verify that it is allowed
        if ( client ) {
            if ( !SockAddr_Hostare_Equal( (sockaddr*) &mSettings->peer,
                                          (sockaddr*) &server->peer ) ) {
                // Not allowed try again
                connect( mSettings->mSock,
                         (sockaddr*) &server->peer,
                         server->size_peer );
                close( mSettings->mSock );
                mSettings->mSock = -1;
                Listen( );
                continue;
            }
        }

        // Create an entry for the connection list
        listtemp = new Iperf_ListEntry;
        memcpy(listtemp, &server->peer, sizeof(iperf_sockaddr));
        listtemp->server = server;
        listtemp->next = NULL;

        // See if we need to do summing
        exist = Iperf_hostpresent( &server->peer, clients);
```

```
        if ( exist != NULL ) {
            // Copy group ID
            listtemp->holder = exist->holder;
            server->multihdr = exist->holder;
        } else {
            server->mThreads = 0;
            Mutex_Lock( &groupCond );
            groupID--;
            listtemp->holder = InitMulti( server, groupID );
            server->multihdr = listtemp->holder;
            Mutex_Unlock( &groupCond );
        }

        // Store entry in connection list
        Iperf_pushback( listtemp, &clients );

        tempSettings = NULL;
        if ( !isCompat( mSettings ) && !isMulticast( mSettings ) ) {
            Settings_GenerateClientSettings( server, &tempSettings,
                                              hdr );
        }


        if ( tempSettings != NULL ) {
            client_init( tempSettings );
            if ( tempSettings->mMode == kTest_DualTest ) {
#ifdef HAVE_THREAD
                thread_start( tempSettings );
#else
                server->runNext = tempSettings;
#endif
            } else {
                server->runNext =  tempSettings;
            }
        }
        server->reporthdr = InitReport( server );

        // Prep for next connection
        if ( !isSingleClient( mSettings ) ) {
            mClients--;
        }
        Settings_Copy( mSettings, &server );
        server->mThreadMode = kMode_Server;
    } while ( !sInterupted && (!mCount || ( mCount && mClients > 0 )) );
    Mutex_Unlock( &clients_mutex );

    Settings_Destroy( server );
}

/* -------------------------------------------------------------------
 * Run the server as a daemon
 * --------------------------------------------------------------------*/

void Listener::runAsDaemon(const char *pname, int facility) {
#ifndef WIN32
    pid_t pid;

    /* Create a child process & if successful, exit from the parent process
*/
    if ( (pid = fork()) == -1 ) {
        fprintf( stderr, "error in first child create\n");
        exit(0);
    } else if ( pid != 0 ) {
        exit(0);
    }

    /* Try becoming the session leader, once the parent exits */
    if ( setsid() == -1 ) {           /* Become the session leader */
```

```
        fprintf( stderr, "Cannot change the session group leader\n");
    } else {
    }
    signal(SIGHUP,SIG_IGN);


    /* Now fork() and get released from the terminal */
    if ( (pid = fork()) == -1 ) {
        fprintf( stderr, "error\n");
        exit(0);
    } else if ( pid != 0 ) {
        exit(0);
    }

    chdir(".");
    fprintf( stderr, "Running Iperf Server as a daemon\n");
    fprintf( stderr, "The Iperf daemon process ID : %d\n",((int)getpid()));
    fflush(stderr);

    fclose(stdin);
#else
    fprintf( stderr, "Use the precompiled windows version for service
(daemon) option\n");
#endif

}
```

11.2 Ping

Ping is a computer network administration utility used to test the reachability of a host on an Internet Protocol (IP) network and to measure the round-trip time for messages sent from the originating host to a destination host. The name comes from active sonar terminology which sends a pulse of sound and listens for the echo [60].

Ping operates by sending Internet Control Message Protocol (ICMP) echo request packets to the target host and waiting for an ICMP response. In the process it measures the time from transmission to reception (round-trip time) and records any packet loss. The results of the test are printed in the form of a statistical summary of the response packets received, including the minimum, maximum, and the mean round-trip times, and sometimes the standard deviation of the mean [60].

Depending on the implementation, the ping command can be run with various command line switches to enable special operational modes. Example options include: specifying the packet size used as the probe, automatic repeated operation for sending a specified count of probes, and time stamping [58]. The Ping source code is provided below [64].

11.2.1 Ping Source Code

```
/*
 *                        P I N G . C
 *
 * Using the InterNet Control Message Protocol (ICMP) "ECHO" facility,
 * measure round-trip-delays and packet loss across network paths.
 *
 * Author -
 *      Mike Muuss
 *      U. S. Army Ballistic Research Laboratory
 *      December, 1983
 * Modified at Uc Berkeley
 *
 * Changed argument to inet_ntoa() to be struct in_addr instead of u_long
 * DFM BRL 1992
 *
 * Status -
 *      Public Domain.  Distribution Unlimited.
 *
 * Bugs -
 *      More statistics could always be gathered.
 *      This program has to run SUID to ROOT to access the ICMP socket.
 */

#include <stdio.h>
#include <errno.h>
#include <sys/time.h>

#include <sys/param.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/file.h>

#include <netinet/in_systm.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#include <netdb.h>

#define  MAXWAIT          10      /* max time to wait for response, sec. */
#define  MAXPACKET        4096    /* max packet size */
#define VERBOSE           1       /* verbose flag */
#define QUIET             2       /* quiet flag */
#define FLOOD             4       /* floodping flag */
#ifndef MAXHOSTNAMELEN
#define MAXHOSTNAMELEN    64
#endif

u_char   packet[MAXPACKET];
int      i, pingflags, options;
extern   int errno;

int s;                    /* Socket file descriptor */
struct hostent *hp;       /* Pointer to host info */
struct timezone tz;       /* leftover */

struct sockaddr whereto;/* Who to ping */
int datalen;              /* How much data */

char usage[] =
"Usage:  ping [-dfqrv] host [packetsize [count [preload]]]n";

char *hostname;
char hnamebuf[MAXHOSTNAMELEN];
```

```
int npackets;
int preload = 0;          /* number of packets to "preload" */
int ntransmitted = 0;             /* sequence # for outbound packets = #sent
*/
int ident;

int nreceived = 0;                /* # of packets we got back */
int timing = 0;
int tmin = 999999999;
int tmax = 0;
int tsum = 0;                     /* sum of all times, for doing average */
int finish(), catcher();
char *inet_ntoa();

/*
 *                        M A I N
 */
main(argc, argv)
char *argv[];
{
        struct sockaddr_in from;
        char **av = argv;
        struct sockaddr_in *to = (struct sockaddr_in *) &whereto;
        int on = 1;
        struct protoent *proto;

        argc--, av++;
        while (argc > 0 && *av[0] == '-') {
                while (*++av[0]) switch (*av[0]) {
                        case 'd':
                                options |= SO_DEBUG;
                                break;
                        case 'r':
                                options |= SO_DONTROUTE;
                                break;
                        case 'v':
                                pingflags |= VERBOSE;
                                break;
                        case 'q':
                                pingflags |= QUIET;
                                break;
                        case 'f':
                                pingflags |= FLOOD;
                                break;
                }
                argc--, av++;
        }
        if(argc < 1 || argc > 4)  {
                printf(usage);
                exit(1);
        }

        bzero((char *)&whereto, sizeof(struct sockaddr) );
        to->sin_family = AF_INET;
        to->sin_addr.s_addr = inet_addr(av[0]);
        if(to->sin_addr.s_addr != (unsigned)-1) {
                strcpy(hnamebuf, av[0]);
                hostname = hnamebuf;
        } else {
                hp = gethostbyname(av[0]);
                if (hp) {
                        to->sin_family = hp->h_addrtype;
                        bcopy(hp->h_addr, (caddr_t)&to->sin_addr, hp-
>h_length);
                        hostname = hp->h_name;
                } else {
                        printf("%s: unknown host %sn", argv[0], av[0]);
                        exit(1);
```

```
        }
}

if( argc >= 2 )
        datalen = atoi( av[1] );
else
        datalen = 64-8;
if (datalen > MAXPACKET) {
        fprintf(stderr, "ping: packet size too largen");
        exit(1);
}
if (datalen >= sizeof(struct timeval))    /* can we time 'em? */
        timing = 1;

if (argc >= 3)
        npackets = atoi(av[2]);

if (argc == 4)
        preload = atoi(av[3]);

ident = getpid() & 0xFFFF;

if ((proto = getprotobyname("icmp")) == NULL) {
        fprintf(stderr, "icmp: unknown protocoln");
        exit(10);
}

if ((s = socket(AF_INET, SOCK_RAW, proto->p_proto)) < 0) {
        perror("ping: socket");
        exit(5);
}
if (options & SO_DEBUG) {
        if(pingflags & VERBOSE)
                printf("...debug on.n");
        setsockopt(s, SOL_SOCKET, SO_DEBUG, &on, sizeof(on));
}
if (options & SO_DONTROUTE) {
        if(pingflags & VERBOSE)
                printf("...no routing.n");
        setsockopt(s, SOL_SOCKET, SO_DONTROUTE, &on, sizeof(on));
}

if(to->sin_family == AF_INET) {
        printf("PING %s (%s): %d data bytesn", hostname,
          inet_ntoa(to->sin_addr), datalen);      /* DFM */
} else {
        printf("PING %s: %d data bytesn", hostname, datalen );
}
setlinebuf( stdout );

signal( SIGINT, finish );
signal(SIGALRM, catcher);

/* fire off them quickies */
for(i=0; i < preload; i++)
        pinger();

if(!(pingflags & FLOOD))
        catcher();      /* start things going */

for (;;) {
        int len = sizeof (packet);
        int fromlen = sizeof (from);
        int cc;
        struct timeval timeout;
        int fdmask = 1 << s;

        timeout.tv_sec = 0;
```

```
		timeout.tv_usec = 10000;

		if(pingflags & FLOOD) {
			pinger();
			if( select(32, &fdmask, 0, 0, &timeout) == 0)
				continue;
		}
		if ( (cc=recvfrom(s, packet, len, 0, &from, &fromlen)) < 0)
{
			if( errno == EINTR )
				continue;
			perror("ping: recvfrom");
			continue;
		}
		pr_pack( packet, cc, &from );
		if (npackets && nreceived >= npackets)
			finish();
	}
	/*NOTREACHED*/
}

/*
 *			C A T C H E R
 *
 * This routine causes another PING to be transmitted, and then
 * schedules another SIGALRM for 1 second from now.
 *
 * Bug -
 *	Our sense of time will slowly skew (ie, packets will not be launched
 *	exactly at 1-second intervals).  This does not affect the quality
 *	of the delay and loss statistics.
 */
catcher()
{
	int waittime;

	pinger();
	if (npackets == 0 || ntransmitted < npackets)
		alarm(1);
	else {
		if (nreceived) {
			waittime = 2 * tmax / 1000;
			if (waittime == 0)
				waittime = 1;
		} else
			waittime = MAXWAIT;
		signal(SIGALRM, finish);
		alarm(waittime);
	}
}

/*
 *			P I N G E R
 *
 * Compose and transmit an ICMP ECHO REQUEST packet.  The IP packet
 * will be added on by the kernel.  The ID field is our UNIX process ID,
 * and the sequence number is an ascending integer.  The first 8 bytes
 * of the data portion are used to hold a UNIX "timeval" struct in VAX
 * byte-order, to compute the round-trip time.
 */
pinger()
{
	static u_char outpack[MAXPACKET];
	register struct icmp *icp = (struct icmp *) outpack;
	int i, cc;
	register struct timeval *tp = (struct timeval *) &outpack[8];
	register u_char *datap = &outpack[8+sizeof(struct timeval)];
```

```
        icp->icmp_type = ICMP_ECHO;
        icp->icmp_code = 0;
        icp->icmp_cksum = 0;
        icp->icmp_seq = ntransmitted++;
        icp->icmp_id = ident;            /* ID */

        cc = datalen+8;                  /* skips ICMP portion */

        if (timing)
                gettimeofday( tp, &tz );

        for( i=8; i<datalen; i++) /* skip 8 for time */
                *datap++ = i;

        /* Compute ICMP checksum here */
        icp->icmp_cksum = in_cksum( icp, cc );

        /* cc = sendto(s, msg, len, flags, to, tolen) */
        i = sendto( s, outpack, cc, 0, &whereto, sizeof(struct sockaddr) );

        if( i < 0 || i != cc )  {
                if( i<0 )  perror("sendto");
                printf("ping: wrote %s %d chars, ret=%dn",
                        hostname, cc, i );
                fflush(stdout);
        }
        if(pingflags == FLOOD) {
                putchar('.');
                fflush(stdout);
        }
}

/*
 *                      P R _ T Y P E
 *
 * Convert an ICMP "type" field to a printable string.
 */
char *
pr_type( t )
register int t;
{
        static char *ttab[] = {
                "Echo Reply",
                "ICMP 1",
                "ICMP 2",
                "Dest Unreachable",
                "Source Quench",
                "Redirect",
                "ICMP 6",
                "ICMP 7",
                "Echo",
                "ICMP 9",
                "ICMP 10",
                "Time Exceeded",
                "Parameter Problem",
                "Timestamp",
                "Timestamp Reply",
                "Info Request",
                "Info Reply"
        };

        if( t < 0 || t > 16 )
                return("OUT-OF-RANGE");

        return(ttab[t]);
}

/*
```

```
 *                       P R _ P A C K
 *
 * Print out the packet, if it came from us.  This logic is necessary
 * because ALL readers of the ICMP socket get a copy of ALL ICMP packets
 * which arrive ('tis only fair).  This permits multiple copies of this
 * program to be run without having intermingled output (or statistics!).
 */
pr_pack( buf, cc, from )
char *buf;
int cc;
struct sockaddr_in *from;
{
        struct ip *ip;
        register struct icmp *icp;
        register long *lp = (long *) packet;
        register int i;
        struct timeval tv;
        struct timeval *tp;
        int hlen, triptime;

        from->sin_addr.s_addr = ntohl( from->sin_addr.s_addr );
        gettimeofday( &tv, &tz );

        ip = (struct ip *) buf;
        hlen = ip->ip_hl << 2;
        if (cc < hlen + ICMP_MINLEN) {
                if (pingflags & VERBOSE)
                        printf("packet too short (%d bytes) from %sn", cc,
                                inet_ntoa(ntohl(from->sin_addr))); /* DFM
*/
                return;
        }
        cc -= hlen;
        icp = (struct icmp *)(buf + hlen);
        if( (!(pingflags & QUIET)) && icp->icmp_type != ICMP_ECHOREPLY )  {
                printf("%d bytes from %s: icmp_type=%d (%s) icmp_code=%dn",
                  cc, inet_ntoa(ntohl(from->sin_addr)),
                  icp->icmp_type, pr_type(icp->icmp_type), icp-
>icmp_code);/*DFM*/
                if (pingflags & VERBOSE) {
                        for( i=0; i<12; i++)
                                printf("x%2.2x: x%8.8xn", i*sizeof(long),
                                  *lp++);
                }
                return;
        }
        if( icp->icmp_id != ident )
                return;                 /* 'Twas not our ECHO */

        if (timing) {
                tp = (struct timeval *)&icp->icmp_data[0];
                tvsub( &tv, tp );
                triptime = tv.tv_sec*1000+(tv.tv_usec/1000);
                tsum += triptime;
                if( triptime < tmin )
                        tmin = triptime;
                if( triptime > tmax )
                        tmax = triptime;
        }

        if(!(pingflags & QUIET)) {
                if(pingflags != FLOOD) {
                        printf("%d bytes from %s: icmp_seq=%d", cc,
                          inet_ntoa(from->sin_addr),
                          icp->icmp_seq );       /* DFM */
                        if (timing)
                                printf(" time=%d msn", triptime );
                        else
```

```
                                putchar('n');
                } else {
                        putchar('b');
                        fflush(stdout);
                }
        }
        nreceived++;
}


/*
 *                      I N _ C K S U M
 *
 * Checksum routine for Internet Protocol family headers (C Version)
 *
 */
in_cksum(addr, len)
u_short *addr;
int len;
{
        register int nleft = len;
        register u_short *w = addr;
        register u_short answer;
        register int sum = 0;

        /*
         *  Our algorithm is simple, using a 32 bit accumulator (sum),
         *  we add sequential 16 bit words to it, and at the end, fold
         *  back all the carry bits from the top 16 bits into the lower
         *  16 bits.
         */
        while( nleft > 1 )  {
                sum += *w++;
                nleft -= 2;
        }

        /* mop up an odd byte, if necessary */
        if( nleft == 1 ) {
                u_short  u = 0;

                *(u_char *)(&u) = *(u_char *)w ;
                sum += u;
        }

        /*
         * add back carry outs from top 16 bits to low 16 bits
         */
        sum = (sum >> 16) + (sum & 0xffff);       /* add hi 16 to low 16 */
        sum += (sum >> 16);                       /* add carry */
        answer = ~sum;                            /* truncate to 16 bits */
        return (answer);
}

/*
 *                      T V S U B
 *
 * Subtract 2 timeval structs:  out = out - in.
 *
 * Out is assumed to be >= in.
 */
tvsub( out, in )
register struct timeval *out, *in;
{
        if( (out->tv_usec -= in->tv_usec) < 0 )   {
                out->tv_sec--;
                out->tv_usec += 1000000;
        }
        out->tv_sec -= in->tv_sec;
```

```
}

/*
 *                        F I N I S H
 *
 * Print out statistics, and give up.
 * Heavily buffered STDIO is used here, so that all the statistics
 * will be written with 1 sys-write call.  This is nice when more
 * than one copy of the program is running on a terminal;  it prevents
 * the statistics output from becomming intermingled.
 */
finish()
{
        putchar('n');
        fflush(stdout);
        printf("n----%s PING Statistics----n", hostname );
        printf("%d packets transmitted, ", ntransmitted );
        printf("%d packets received, ", nreceived );
        if (ntransmitted)
                if( nreceived > ntransmitted)
                        printf("-- somebody's printing up packets!");
                else
                        printf("%d%% packet loss",
                          (int) (((ntransmitted-nreceived)*100) /
                          ntransmitted));
        printf("n");
        if (nreceived && timing)
            printf("round-trip (ms)  min/avg/max = %d/%d/%dn",
                tmin,
                tsum / nreceived,
                tmax );
        fflush(stdout);
        exit(0);
}
```

11.3 Wireshark

Wireshark is a free and open-source packet analyser. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Originally named Ethereal, in May 2006 the project was renamed Wireshark due to trademark issues [54] [55].

Wireshark is cross-platform, using the GTK+ widget toolkit to implement its user interface, and using pcap to capture packets; it runs on various Unix-like operating systems including Linux, Mac OS X, BSD, Solaris, and Microsoft Windows. There is also a terminal-based (non-GUI) version called TShark. Wireshark, and the other programs distributed with it such as TShark, are free software, released under the terms of the GNU General Public License [55].

Wireshark is very similar to tcpdump, but has a graphical front-end, plus some integrated sorting and filtering options. Wireshark allows the user to put network interface controllers that support promiscuous mode in order to see all traffic visible on that interface not just traffic addressed to one of the interface's configured addresses and broadcast / multicast traffic [54] [55].

11.3.1 Features of Wireshark

Wireshark is software that "understands" the structure of different networking protocols. Thus, it is able to display the encapsulation and the fields along with their meanings of different packets specified by different networking protocols. Wireshark uses pcap to capture packets, so it can only capture the packets on the types of networks that pcap supports [54] [55].

1. Data can be captured "from the wire" from a live network connection or read from a file that recorded already-captured packets.
2. Live data can be read from a number of types of network, including Ethernet, IEEE 802.11, PPP, and loopback.
3. Captured network data can be browsed via a GUI, or via the terminal (command line) version of the utility, TShark.
4. Captured files can be programmatically edited or converted via command-line switches to the "editcap" program.
5. Data display can be refined using a display filter.
6. Plug-ins can be created for dissecting new protocols.
7. Voice over Internet Protocol (VoIP) calls in the captured traffic can be detected. If encoded in a compatible encoding, the media flow can even be played.
8. Raw USB traffic can be captured.

Wireshark' s native network trace file format is the libpcap format supported by libpcap and WinPcap, so it can exchange files of captured network traces with other applications using the same format, including tcpdump and CA NetMaster. It can also read captures from other network analysers, such as snoop, Network General's Sniffer, and Microsoft Network Monitor [54] [55].

11.4 MATLAB

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation. Using MATLAB, one can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran. MATLAB can be used in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modelling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions) extend the MATLAB environment to solve particular classes of problems in these application areas [57].

Key Features of MATLAB include [57]:

1. High-level language for technical computing.
2. Development environment for managing code, files, and data.
3. Interactive tools for iterative exploration, design, and problem solving.
4. Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration.
5. 2-D and 3-D graphics functions for visualizing data.
6. Tools for building custom graphical user interfaces.
7. Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java, COM, and Microsoft Excel.

11.4.1 The MATLAB Language

The MATLAB language supports the vector and matrix operations that are fundamental to engineering and scientific problems. It enables fast development and execution. With the MATLAB language, you can program and develop algorithms faster than with traditional languages because you do not need to perform low-level administrative tasks, such as declaring variables, specifying data types, and allocating memory. In many cases, MATLAB eliminates the need for 'for' loops. As a result, one line of MATLAB code can often replace several lines of C or C++ code. At the same time, MATLAB provides all the features of a traditional programming language, including arithmetic operators, flow control, data structures, data types, object-oriented programming (OOP), and debugging features [57].

For fast execution of heavy matrix and vector computations, MATLAB uses processor-optimised libraries. For general purpose scalar computations, MATLAB generates machine-code instructions using its Just-In-Time (JIT) compilation technology. This technology, which is available on most platforms, provides execution speeds that rival those of traditional programming languages [57].

Listed below is the Matlab source code for simulations conducted in the research work listed.

11.4.2 System Blocking Probability with SIP recognition – MATLAB Source Code

```
sum = 0;
a1 = 10; %number of lines;

for E = 1:20
    for i = 0:a1
        sum = sum + E^i/factorial(i);
    end
Pb1(E) = (E^a1/factorial(a1))/sum;
end


sum = 0;
a1 = 10; %number of lines;

for E = 1:20
    for i = 0:a1
        sum = sum + E^i/factorial(i);
    end
Pb2(E) = (E^a1/factorial(a1))/sum + 0.5;
end


sum = 0;
a1 = 10; %number of lines;

for E = 1:20
    for i = 0:a1
        sum = sum + E^i/factorial(i);
    end
Pb3(E) = (E^a1/factorial(a1))/sum + 0.2;
end

sum = 0;
a1 = 10; %number of lines;

for E = 1:20
    for i = 0:a1
        sum = sum + E^i/factorial(i);
```

```
    end
Pb4(E) = (E^a1/factorial(a1))/sum + 0.1;
end

plot (Pb1, 'g--^')
hold on
plot (Pb2, 'r--*')
hold on
plot (Pb3, 'm--o')
hold on
plot (Pb4, 'b--s')
ylabel('Blocking
Probability','fontname','garamond','fontsize',10,'fontweight','bold')
xlabel('Traffic                                                      Increase
(Erlang)','fontname','garamond','fontsize',10,'fontweight','bold')
```

11.4.3 Analysis of Changing *p* – MATLAB Source Code

```
%simulation of p
%1.4 Mhz = 6 Meg BW hence 120 lines max considering spectral efficiency and
voice coding

p1= 0.1;
Ct = 6;
R = 0.042;
a1 = round((Ct*(1-p1))/R);
E = 100;
sum = 0;
    for i = 0:a1
        sum = sum + E^i/factorial(i);
    end
Pb1 = (E^a1/factorial(a1))/sum


p2= 0.25;
a1 = round((Ct*(1-p2))/R);
sum = 0;
    for i = 0:a1
        sum = sum + E^i/factorial(i);
    end
Pb2 = (E^a1/factorial(a1))/sum

p3= 0.5;
a1 = round((Ct*(1-p3))/R);
sum = 0;
    for i = 0:a1
        sum = sum + E^i/factorial(i);
    end
Pb3 = (E^a1/factorial(a1))/sum


p4= 0.75;
a1 = round((Ct*(1-p4))/R);

sum = 0;
    for i = 0:a1
        sum = sum + E^i/factorial(i);
    end
Pb4 = (E^a1/factorial(a1))/sum

x = [p1, p2, p3, p4];
y = [Pb1, Pb2, Pb3, Pb4];
plot (x,y);
```

```
ylabel('Blocking
Probability','fontname','garamond','fontsize',12,'fontweight','bold')
xlabel('Factor "p"','fontname','garamond','fontsize',12,'fontweight','bold')
```

12 Appendix B - Publications

12.1 Publications in Current Research

1. Nageshar N, Van Olst R, "A Heuristic Analysis Approach to Admission Control (AC) for Voice in Packet Switched Wireless Networks," in IEEE Africon, Livingstone, Zambia, September 2011, pp. 1-6.
2. Nageshar N, Van Olst R, "Maintenance of Voice Quality Control in the Evolution to Packet Switched Wireless Networks," in South African Telecommunication Networks and Applications Conference (SATNAC), East London, South Africa, September, 2011.
3. Nageshar N, Van Olst R, "Regulation of Bearer / Service Flow Selection between Network Domains for Voice over Packet Switched Wireless Networks," in ITU Kaleidoscope, Cape Town, South Africa, December, 2011, pp. 175-180.
4. Nageshar N, Van Olst R, "Deep Packet Inspection (DPI) Initiated Quality of Service (QoS) for Over the Top (OTT) Voice in LTE Network," to be submitted for publication.

12.2 Publications in Previous Research

5. Nageshar N, Mneney S H, Sewsunker R, "Economic Analysis of a Satellite Terrestrial Telephony System for Rural SADC," in IEEE Africon, George, South Africa, October 2002, pp. 401–405.
6. Nageshar N, Mneney S H, Sewsunker R, "Economics Involved in providing Universal Telecommunications Access for the SADC Region," in South African Telecommunication Networks and Applications Conference (SATNAC), Champagne Sports Resort, Kwazulu-Natal, South Africa., September 2002.
7. Nageshar N, Mneney S H, Sewsunker R, "Universal Telecommunications Access using Satellite Platforms," in South African Telecommunication Networks and Applications Conference (SATNAC), Wild Coast Sun, South Africa, September 2001.

8. Nageshar N, Mneney S H, "The Southern African Development Community Telecommunications Satellite," in South African Telecommunication Networks and Applications Conference (SATNAC), Cape Town, September 2000.

**Note: Only publications in current research are printed in Appendix B.