Quality of Service Support for Voice over IP in Wireless Access Networks

by

Cristian Olariu

Under the supervision of:
Dr. Ruairí de Fréin
Prof. Liam Murphy

Thesis submitted in partial fulfilment of the requirements for the award of

Doctor of Philosophy

November 2013

Department of Computing, Mathematics and Physics
Waterford Institute of Technology

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy *is entirely my own work and has not been taken from the work of others save to the extent that such work has been cited and acknowledged within the text of my work.*

Signed:

ID number: W20050956

Date: 6^{th} of November, 2013

Abstract

Voice Over IP (VoIP) is increasingly displacing traditional voice telephony services, but the IP networks that carry the traffic are inherently prone to delays and losses that impact the voice quality. This thesis addresses those quality degradations with particular emphasis on wireless networks.

The scenarios considered in the thesis focus on the two most recent wireless access technologies, namely Long Term Evolution (LTE) and Wireless Mesh Networks (WMNs). These scenarios have in common the undesired effects of call quality degradation when the increased traffic volume across the network causes congestion. This thesis points out the causes of the congestion and proposes solutions tailored specially for each wireless access technology.

The solutions share a common concept, proposed and developed in this thesis, namely the Intermediate Mean Opinion Score (iMOS). The MOS is basically a call quality assessment methodology typically performed at the end-points. The novelty of iMOS resides in the fact that accurate call quality measurements are performed at intermediate points in the network rather than in the end-nodes. The iMOS proposed in this thesis uses the E-Model to estimate the quality of VoIP calls, based on network performance indicators. Measuring the call quality at intermediate points is driven by the operators' intent to guarantee high levels of VoIP quality of service.

The users' increased need for wireless access generates an unpredicted growth of traffic volume, which is directly reflected in the perceived VoIP call quality. The second part of this thesis addresses this problem and focuses on developing Call Admission Control (CAC) solutions designed to support the quality of service for VoIP while maximizing the amount of traffic allowed to pass through the network.

The iMOS concept and iMOS-based CAC solutions are validated through extensive simulations of realistic deployment scenarios of LTE femtocells and WMNs. The results highlight the existence of the call degradation problems and the proposed solutions are shown to have the capability of increasing the call quality and call capacity, and protecting against congestion which causes the undesired call quality degradation.

Acknowledgements

Firstly I would like to express my gratitude to the two research groups—the Telecommunications Software & Systems Group(TSSG) of Waterford Institute of Technology(WIT), and the Performance Engineering Laboratory(PEL) of University College Dublin(UCD)—which hosted me and provided the necessary funding and resources needed to conduct this research work. Many thanks especially to the FAME and CARMESH projects, and their creators, for facilitating the collaborations that helped this work.

I would like to thank Dr. Mícheál Ó Foghlú, Dr. Brendan Jennings and Dr. Ruairí de Fréin who acted sequentially as supervisors from TSSG/WIT and to Prof. Liam Murphy from PEL/UCD who actively maintained his supervisor role throughout the whole period.

I cannot forget to acknowledge the great support received from Dr. Philip Perry who guided my first steps in research and the great support received from Dr. John Fitzpatrick who shared his research expertise, vast knowledge in the field and contributed on all levels, being always available for a chat. Many thanks also to Dr. Yacine Ghamri-Doudane for his many comments and contributions to my research work.

I also wish to thank all my colleagues in the Performance Engineering Laboratory for being there, for their unconditional support, and for the great fun we had together along these years.

Many thanks go to my parents, Simion and Paraschiva, who made many sacrifices to support me throughout the period of my professional development.

Finally, I wish to thank my wife, Adriana, who understood very well the ups and downs of research and managed to cheer me up. Her dedication and support was infinite and I am forever grateful for her being always there for me.

Dublin, 6th of November, 2013 *Cristian Olariu*

CONTENTS

List of Figures **vii**

List of Tables **x**

List of Acronyms **xi**

Publications **xv**

1 Introduction **1**
1.1 Problem Statement 3
1.2 Overview of Proposed Solution 4
1.3 Contributions 5
1.4 Thesis Outline 6

2 Background **8**
2.1 VoIP 8
2.1.1 VoIP Transport 10
2.1.1.1 VoIP codecs 10
2.1.1.2 VoIP Transport Protocols 12
2.1.1.3 VoIP Traffic Metrics 13
2.1.1.4 VoIP QoS Protocols 15
2.1.2 VoIP Call Quality Assessment 16
2.1.2.1 The E-Model Algorithm 17
2.1.2.2 *A* 18
2.1.2.3 Id 19
2.1.2.4 Ie-eff 20
2.1.2.5 R to MOS 20
2.1.3 Speech Model 23
2.2 Femtocell Networks 24
2.2.1 LTE Architecture 26

2.2.2 LTE Femtocell Architecture 27
2.2.3 Femtocell Backhaul 31
2.3 Wireless Mesh Networks 33
2.3.1 WMN architecture 34
2.3.1.1 Components 34
2.3.1.2 Interfaces 34
2.3.1.3 Topologies 35
2.3.2 Routing 37
2.3.3 Quality of Service in 802.11 38
2.3.3.1 802.11e 39
2.3.3.2 Frame Aggregation in 802.11e 40
2.4 Summary 41

3 Literature Review 43
3.1 Delay Measurement and Estimation 43
3.2 Edge-to-middle VoIP Call Quality Monitoring Solutions 47
3.2.1 VoIP Backbone Monitoring 47
3.2.2 VoIP Monitoring Related to iMOS 49
3.3 VoIP Call Admission Control 51
3.4 Summary 55

4 iMOS — Estimating VoIP Call Quality at Intermediate Points 56
4.1 The iMOS Concept 56
4.2 Measuring the iMOS in DSL-backhauled Femtocell Deployments 59
4.2.1 Description 59
4.2.1.1 Targeted Femtocell Deployment Scenario 59
4.2.1.2 The Recency Effect 62
4.2.2 Simulation Parameters 64
4.2.3 Results Analysis 66
4.2.4 Summary 66
4.3 Measuring the iMOS in VoIP over WMN Deployments 68
4.3.1 Description 68
4.3.1.1 Targeted WMN Scenario 68
4.3.1.2 VoIP Capacity 71
4.3.2 The Delay Piggy-Backing Mechanism (DPBM) 73
4.3.3 Simulation Parameters 77
4.3.4 Results Analysis 79
4.3.4.1 Comparison methodology 79
4.3.4.2 VoIP Call Quality and VoIP Capacity Evaluation 80
4.3.4.3 Delay Measurement Accuracy Evaluation 81
4.3.4.4 Delay, Packet Loss and Jitter Evaluation 84
4.3.5 Summary 87

4.4 Summary 88

5 iMOS-based Call Admission Control 90
5.1 Call Admission Control for VoIP 90
5.2 iMOS-based CAC in DSL-backhauled Femtocell Deployments . . 91
5.2.1 Description 91
5.2.1.1 Targetted Femtocell Architecture 92
5.2.1.2 The DSLAM is the Bottleneck 94
5.2.2 iMOS-based Call Admission Control in Femtocell Networks 95
5.2.3 Simulation Parameters 100
5.2.4 Results Analysis 102
5.2.5 Summary 105
5.3 iMOS-based CAC in WMN-backhauled Femtocell Deployments . 107
5.3.1 Description 107
5.3.1.1 Targeted WMN & LTE Femtocell Architecture . 110
5.3.1.2 iMOS-based CAC Mechanism in WMN-backhauled LTE femtocells 110
5.3.2 Simulation Parameters 113
5.3.3 Results 116
5.3.4 Summary 120
5.4 Summary 122

6 Conclusion 123
6.1 Future Work 124

References 137

Appendix A NS-3 Code 138
A.1 NS-3 Implementation of the E-Model - ITU-T G.107 138
A.2 NS-3 Implementation of the Speech Model - ITU-T P.59 148
A.3 NS-3 Implementation of the PIFO Queue 152

LIST OF FIGURES

2.1 Mean Opinion Score (MOS) vs. Delay per codec 22
2.2 MOS vs. Jitter per codec 22
2.3 MOS vs. Loss per codec 23
2.4 Conversation Model from ITU-T/P.59 24
2.5 LTE Architecture 26
2.6 LTE Femtocell Architecture 28
2.7 LTE Femtocell User-Plane Protocol Stack 29
2.8 LTE Femtocell Control-Plane Protocol Stack 30
2.9 LTE Local-Femto Architecture 31
2.10 WMN Architecture 35
2.11 Types of WMN Topologies 37
2.12 Frame Aggregation in a Wireless Medium 41

3.1 Call Admission Control Taxonomy 53

4.1 MOS vs. Estimated RTT Delay 58
4.2 Typical deployment for LTE Femtocells with DSL backhaul . . . 59
4.3 The concept of iMOS in DSL-backhauled femtocell deployments 61
4.4 LTE Femtocell deployment, with multiple possibilities to connect the caller to the callee 62
4.5 Weighted MOS score mapped into corresponding call quality categories 63
4.6 Simulation Scenario 65
4.7 Targeted WMN Deployment Scenario 70
4.8 Simulation Topology and Call Placement 77

4.9 MOS versus number of injected calls on the uplink. The intersection of the threshold line (iMOS=3.6) and the curves represent the VoIP capacity measurements. *Legend: m represents the* $\mu(\mu(iMOS))$, *v1 represents* $\mu(\mu(iMOS)) \pm \mu(\sigma(iMOS))$ *and v2 represents the* $\mu(\mu(iMOS)) \pm \sigma(\mu(iMOS))$. *The circled values on the plot are VoIP capacity measurements.* 81
4.10 MOS versus number of injected calls on the downlink. The intersection of the threshold line (iMOS=3.6) and the curves represent the VoIP capacity measurements. *Legend: m represents the* $\mu(\mu(iMOS))$, *v1 represents* $\mu(\mu(iMOS)) \pm \mu(\sigma(iMOS))$ *and v2 represents the* $\mu(\mu(iMOS)) \pm \sigma(\mu(iMOS))$. *The circled values on the plot are VoIP capacity measurements.* 82
4.11 Accuracy of the piggy-backed delay value. *The points in the plot represent the difference between actual delay and the piggy-backed delay* 83
4.12 Delay versus number of injected calls on the uplink. *Legend: m represents the* $\mu(\mu(Delay))$, *v1 represents* $\mu(\mu(Delay)) \pm \mu(\sigma(Delay))$ *and v2 represents the* $\mu(\mu(Delay)) \pm \sigma(\mu(Delay))$. 84
4.13 Delay versus number of injected calls on the downlink. *Legend: m represents the* $\mu(\mu(Delay))$, *v1 represents* $\mu(\mu(Delay)) \pm \mu(\sigma(Delay))$ *and v2 represents the* $\mu(\mu(Delay)) \pm \sigma(\mu(Delay))$. . . 85
4.14 Packet loss versus number of injected calls on the uplink. *Legend: m represents the* $\mu(\mu(PacketLoss))$, *v1 represents* $\mu(\mu(PacketLoss)) \pm \mu(\sigma(PacketLoss))$ *and v2 represents the* $\mu(\mu(PacketLoss)) \pm \sigma(\mu(PacketLoss))$. 85
4.15 Packet loss versus number of injected calls on the downlink. *Legend: m represents the* $\mu(\mu(PacketLoss))$, *v1 represents* $\mu(\mu(PacketLoss)) \pm \mu(\sigma(PacketLoss))$ *and v2 represents the* $\mu(\mu(PacketLoss)) \pm \sigma(\mu(PacketLoss))$. 86
4.16 Jitter versus number of injected calls on the uplink. *Legend: m represents the* $\mu(\mu(Jitter))$, *v1 represents* $\mu(\mu(Jitter)) \pm \mu(\sigma(Jitter))$ *and v2 represents the* $\mu(\mu(Jitter)) \pm \sigma(\mu(PacketLoss))$. 86
4.17 Jitter versus number of injected calls on the downlink. *Legend: m represents the* $\mu(\mu(Jitter))$, *v1 represents* $\mu(\mu(Jitter)) \pm \mu(\sigma(Jitter))$ *and v2 represents the* $\mu(\mu(Jitter)) \pm \sigma(\mu(Jitter))$. . . . 87

5.1 Architecture of Femtocell Deployment over DSL backhaul networks 93
5.2 Voice over S1-U 94
5.3 Logical diagram of the proposed CAC and Dynamic Resource Allocation Mechanism 97
5.4 Femtocell Deployment Architecture with emphasis on MNO-to-ISP massage interchange 99
5.5 Simulation Setup 101

5.6 iMOS and Number of Active Calls versus Time when the iMOS-based CAC and Dynamic Resource Allocation Mechanism is Disable . . . 103
5.7 iMOS and Number of Active Calls versus Time when the iMOS-based CAC and Dynamic Resource Allocation Mechanism is Enabled . . . 104
5.8 The fluctuations of the current CIR value in the DSLAM's EF queue . . . 105
5.9 Topology of WMN-backhauled LTE femtocells . . . 109
5.10 Protocol stack of WMN-backhauled LTE femtocells . . . 111
5.11 Call duration and status; dropped calls are the calls which were initially accepted but the CAC decided later to drop, as their corresponding iMOS was low. . . . 115
5.12 Comparison between all combinations resulting from using the CAC, frame aggregation, and delay-piggy-backing mechanisms in enabled or disabled mode. . . . 117
5.13 Influence of CAC Loop Period on voice quality and capacity. . . 119
5.14 Influence of inter-node distance on voice quality and capacity. . . 119
5.15 Influence of CAC iMOS threshold on voice quality and capacity. 121

LIST OF TABLES

2.1 Comparison of VoIP Codecs 11
2.2 Quality of Service (QoS) Requirements per Application (L=Low, M=Medium, H=High) 14
2.3 MOS scale for subjective assessment 16
2.4 Recommended values for the Advantage Factor 19
2.5 R to MOS_{CQE} correspondence for estimative assessment 21
2.6 Evolution of 802.11 Wireless Technology 36
2.7 Enhanced Distributed Channel Access (EDCA) Access Category Mapping 40

4.1 Obtaining the Mean (μ) and Standard Deviation (σ) of iMOS . . 72
4.2 Simulation Setup 76

5.1 Simulation Setup 114
5.2 Possible combinations between the three mechanisms: CAC, Frame Aggregation (AGG), and the Delay Piggy-Backing Mechanism (DPBM) 116

LIST OF ACRONYMS

3GPP 3rd Generation Partnership Project . . . 26
3G Third Generation . . . 25
4G Fourth Generation . . . 26
A-MSDU Aggregated MAC Service Data Unit . . . 40
AC Access Category . . . 39
AMR-NB Adaptive Multi-Rate Narrow-Band . . . 10
AMR-WB Adaptive Multi-Rate Wide-Band . . . 10
AMR Adaptive Multi-Rate . . . 29
AM Aggregation Mechanism . . . 78
AP Access Point . . . 80
ARPA Advanced Research Projects Agency . . . 9
BNG Broadband Network Gateway . . . 92
BPCF Broadband Policy Control Function . . . 100
CAC Call Admission Control . . . 124
CDMA2000 Code Division Multiple Access 2000 . . . 25
CDMA Code Division Multiple Access . . . 25
CEM Customer Experience Manager . . . 57
CIR Committed Information Rate . . . 94
CSMA/CA Carrier Sense Multiple Access with Collision Avoidance . . . 39
CSN Circuit-Switched Network . . . 8
CS Circuit-Switched . . . 1
CW Contention Window . . . 39
DCF Distributed Coordination Function . . . 38
DiffServ Differentiated Services . . . 15
DPBM Delay Piggy-Backing Mechanism . . . 123
DSCP Differentiated Services Code Point . . . 15

DSLAM Digital Subscriber Line Access Multiplexer 64
DSL Digital Subscriber Line 124
E-UTRAN Evolved Universal Terrestrial Radio Access Network 27
EDCA Enhanced Distributed Channel Access 39
EF Expedited Forwarding 94
EIR Excess Information Rate 94
eNB evolved NodeB 26
EPC Evolved Packet Core 27
ETSI European Telecommunications Standards Institute 25
EWA Exponentially Weighted Average 63
EWMA Exponentially Weighted Moving Average 67
FIFO First-In-First-Out 74
FTTH Fiber To The Home 32
GPS Global Positioning System 45
GSM Global System for Mobile Communications 25
HAN Home Area Network 95
HeNBGW Home evolved NodeB Gateway 125
HeNB Home evolved NodeB 27
HNB Home NodeB 28
HSDPA High Speed Downlink Packet Access 25
HSS Home Subscriber Server 27
HSUPA High Speed Uplink Packet Access 26
IETF Internet Engineering Task Force 14
iMOS Intermediate Mean Opinion Score 123
IMT-2000 International Mobile Telecommunications-2000 25
IntServ Integrated Service 15
IPTV Internet Protocol TeleVision 12
IP Internet Protocol 8
IS-95 Interim Standard 95 25
ISP Internet Service Provider 30
ITU-T International Telecommunication Union-Telecommunication Standardisation Sector 16

ITU International Telecommunication Union ... 25
LFG Local Femto Gateway ... 31
LOS Line-of-Sight ... 32
LTE-A Long Term Evolution Advanced ... 59
LTE Long Term Evolution ... 26
MAC Medium Access Control ... 38
MIMO Multiple-Input/Multiple-Output ... 35
MME Mobility Management Entity ... 27
MNO Mobile Network Operator ... 123
MOS Mean Opinion Score ... 11
MPLS Multiprotocol Label Switching ... 15
MSDU MAC Service Data Unit ... 40
NAS Non Access Stratum ... 29
NLOS Non Line-of-Sight ... 32
NVP Network Voice Protocol ... 9
OFDMA Orthogonal Frequency-Division Multiple Access ... 27
OLSR Optimized Link State Routing Protocol ... 78
OWD one-way-delay ... 43
P-GW PDN Gateway ... 27
PAN Personal Area Network ... 33
PBX Private Branch Exchange ... 9
PCEF Policy and Charging Enforcement Function ... 27
PCM Pulse Code Modulation ... 11
PCRF Policy and Charging Rules Function ... 27
PC Personal Computer ... 8
PDN Packet Data Network ... 27
PESQ Perceptual Evaluation of Speech Quality ... 17
PIFO Push-In-First-Out ... 125
PLC Packet Loss Concealment ... 20
PSN Packet-Switched Network ... 8
PSTN Public Switched Telephone Network ... 8
PS Packet-Switched ... 1

QoS Quality of Service 13
RSVP Resource ReSerVation Protocol 15
RTCP Real-Time Control Transport Protocol 12
RTP Real-Time Transport Protocol 9
RTT Round Trip Time 12
R Transmission Rating Factor 17
S-GW Serving Gateway 27
SC-FDMA Single-Carrier Frequency-Division Multiple Access 27
SID Silence Indicator 115
SIP Session Initiation Protocol 9
SLA Service Level Agreement 125
SOHO Small Office/Home Office 32
SON Self-Organizing Network 91
SQM Service Quality Manager 57
TCP Transmission Control Protocol 12
TD-SCDMA Time Division-Synchronous Code Division Multiple Access . 25
ToS Type-of-Service 39
UDP User Datagram Protocol 9
UE User Equipment 24
UMTS Universal Mobile Telecommunications System 25
UTRAN Universal Terrestrial Radio Access Network
VAD Voice Activity Detection 11
VBR Variable Bitrate 23
VoIP Voice Over IP 123
WCDMA Wideband Code Division Multiple Access 25
WiFi Wireless Fidelity 33
WiMAX Worldwide Interoperability for Microwave Access 33
WLAN Wireless Local Area Network 54
WMN Wireless Mesh Network 123

PUBLICATIONS

Published

[1] Cristian Olariu, John Fitzpatrick, Yacine Ghamri-Doudane, and Liam Murphy. Provisioning Call Quality and Capacity for Femtocells over Wireless Mesh Backhaul. In *2013 IEEE 24th International Symposium on Personal, Indoor and Mobile Radio Communications: Mobile and Wireless Networks (PIMRC'13 - Mobile and Wireless Networks)*, London, United Kingdom, September 2013

[2] Cristian Olariu, John Fitzpatrick, Philip Perry, and Liam Murphy. A QoS based call admission control and resource allocation mechanism for LTE femtocell deployment. In *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, pages 884 –888, Las Vegas, USA, jan. 2012. doi: 10.1109/CCNC.2012.6181184

[3] Cristian Olariu, Adriana Hava, Piotr Szczechowiak, and Yacine Ghamri-Doudane. Prototyping Telematic Services in a Wireless Vehicular Mesh Network Environment. In *WCNC 2012 Workshop on Wireless Vehicular Communications and Networks (W5 - Wireless Vehicular Communications and Networks)*, pages 371–376, Paris, France, April 2012

[4] Cristian Olariu, Mícheál Ó Foghlú, Philip Perry, and Liam Murphy. VoIP Quality Monitoring in LTE Femtocells. In *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pages 501–508, Dublin, Ireland, 2011. IEEE

Submitted

[5] Cristian Olariu, John Fitzpatrick, Yacine Ghamri-Doudane, and Liam Murphy. A Delay-Piggybacking–based Packet Prioritisation Mechanism for Voice over IP in Wireless Mesh Networks. *Elsevier Computer Networks*, 2013

CHAPTER

ONE

INTRODUCTION

The transmission of voice, which has been studied for over 100 years, is still a highly investigated topic nowadays. In the early days of telephony, transmitting voice implied guaranteed services provided via dedicated communications mediums called *circuits*. In parallel with the existence of the Circuit-Switched (CS) telephony systems, another concept of transmitting voice emerged – Voice Over IP (VoIP).

VoIP works by converting the human voice into a stream of digital data which is packetized into Internet Protocol (IP) packets and sent over the network to the other end[s] of the call.

VoIP has seen a rapid increase over the last couple of decades and its wide adoption made it the strongest candidate in replacing the traditional telephony systems. One of the barriers for VoIP in becoming the *killer-application* of the Internet, is the lack in guaranteeing high Quality of Service (QoS) levels from end-to-end. Ironically, the strongest point of Packet-Switched (PS) networks–the shared medium approach–represents the main reason high QoS levels can not be guaranteed for, as VoIP packets are not treated differently to other traffic types by default. When mechanisms are used to prioritize VoIP traffic, it would be nearly impossible to allocate enough network resources to guarantee high QoS for all VoIP calls.

Since the appearance of VoIP, the issue regarding the guaranteeing of QoS was in the focus of VoIP's early adopters and since then, many solutions have been proposed. The ubiquitous solution is to prioritize VoIP packets over packets belonging to less time-critical applications, such as web browsing. This has generated the appearance of many standards and protocols, which are added to the IP protocol stack. However, the prioritization solutions work only when all forwarding equipment in the data path support the new protocols. This can be achieved in small-scale networks, but not in the ever-growing public Internet.

The success of IP in wired communications infrastructure has driven its widespread adoption also in the wireless communications infrastructure. The majority of the communications over licensed and unlicensed wireless mediums have both embraced the IP as the means of transferring payload. Most notably, the latest cellular technology–Long Term Evolution (LTE)–has adopted the PS instead of the CS approach used by its predecessors.

In the unlicensed wireless spectrum band, the IEEE 802.11 standard has seen the widest adoption. It has become so ubiquitous, that virtually all buildings, from universities to business offices and especially homes, host at least one 802.11-based Access Point (AP) serving Internet access to roaming users, while the Internet is provided to the APs through wired infrastructures. In order to extend the range of such wireless access networks, some APs can provide Internet connectivity to other APs, and so on. In this way, a large area can be covered by relaying Internet connectivity from one AP to another in a meshed fashion. This kind of wireless access infrastructure in also known as Wireless Mesh Network (WMN).

Wireless access networks are very attractive for *best-effort* delivery but time-critical applications such as VoIP worsen the QoS issue which is even more exacerbated when VoIP is carried over WMNs. This is mainly due to the large overhead involved when transporting VoIP packets which are small in size and are generated very frequently.

Femtocells are similar to 802.11 APs and provide wireless access in the cellular licensed spectrum. Femtocells appeared as a need to increase the overall system capacity of cellular networks and also to solve coverage issues in geographical locations where expensive macro towers do not reach. Another similarity to APs is that femtocells need to be provided with wired connectivity to the cellular core network, which is usually done through the user's Internet connection. As there is no guarantee that all nodes in the data-path are QoS aware, VoIP calls originating or terminating in femtocells can be faced with quality degradation.

1.1 Problem Statement

The Mobile Network Operator (MNO) associated with the femtocell has no control over the wired infrastructure backhauling femtocell traffic into its core-network. This backhaul infrastructure can behave unexpectedly and lead to VoIP call quality degradation sensed by the end-users, while the MNO may have no way to knowing the issue exists.

Alternatively, end-users may opt to use 802.11-based APs to carry their VoIP traffic. However, the wireless medium in this case is not as strictly regulated as in the case of femtocells, hence interference is a major challenge in these networks. The interference can be further translated into increased packet delay and loss ratio which are the main contributors in VoIP call quality degradation. The degradation is even further exacerbated in the context of WMNs.

The issues related to the above described scenarios lead to the following question: **How can VoIP call quality be monitored in the context of the latest wireless access technologies and what solutions can be employed when call quality degrades?**

Furthermore, this thesis investigates issues associated with specific wireless access technologies. Specifically:

- What entity in the femtocell backhaul causes the congestion leading to the quality degradation?
- Which network entity will be enabled to perform quality measurements and what actions can be taken based on these measurements?
- In WMNs, what specific mechanisms can be used to overcome the negative influence of the unlicensed wireless access medium?

1.2 Overview of Proposed Solution

This thesis presents a proposed solution to the issues raised in the previous section: to find a network entity in the path of the VoIP packets, to perform call quality measurements there, and to employ a feedback mechanism to correct potential causes of call quality degradation.

In the case of femtocells, this work leverages the fact that spectrum regulations have imposed a requirement that femtocells should be time-synchronised with the core-network. This allows any network entity on the path of VoIP packets to accurately determine the network delay for each packet. The delay is an important network metric used in the determination of call quality.

This work has identified the Home evolved NodeB Gateway (HeNBGW) as the network element belonging to the MNO where the call quality should be monitored. A Call Admission Control (CAC) mechanism is developed to temporarily block the access of new calls into the system until the quality is restored. In the case that after employing the CAC mechanism, the call quality is still low, a new mechanism is developed to request more resources in the network for the transitory period of low call quality.

In 802.11-based WMNs the component nodes are not time-synchronised with any other network entity on the path of the VoIP packets and, to alleviate this shortcoming, this work proposes a Delay Piggy-Backing Mechanism (DPBM) which enables the WMN gateway to obtain accurate network delay measurements using the VoIP packets traversing the WMN.

In both cases–femtocells and WMNs–the Mean Opinion Score (MOS) values are obtained in intermediate nodes. This is different than the traditional approach where the MOS is obtained at the end-nodes. Using the MOS values obtained at intermediate nodes is one of the outcomes of this work and is referred to as the Intermediate Mean Opinion Score (iMOS).

In both investigated deployment cases, an iMOS-based CAC mechanism is developed to prevent additional calls from being added to the existing ones, thus preventing the network issue to accentuate its effects over the VoIP traffic.

1.3 Contributions

The primary contributions of this thesis can be summarised as follows:

- **Defining the concept of iMOS** - Originally the call quality assessment methods were used to determine an estimative value of the MOS on an end-to-end basis. The iMOS is a new concept of MOS, proposed in this thesis, which enables other network nodes to obtain real time MOS values of ongoing VoIP calls. The iMOS values can further enable the development of new mechanisms and strategies that can cope with call quality degradation. The results obtained by simulating realistic VoIP deployments, show that the iMOS can be used to isolate network issues which cause VoIP call quality degradation.

- **Development of a CAC and dynamic resource allocation mechanism to protect femtocell calls in network congestion scenarios** - The measurement of iMOS is enabled in the HeNBGW where the quality of ongoing femtocell calls is monitored. When the quality degrades to a lower level than pre-established thresholds, the decision to activate CAC is taken. As this action may not be sufficient to re-establish the quality of ongoing calls, a request is made to increase the amount of resources in the forwarding nodes associated with the calls affected by quality degradation. In the investigated scenario, the network node where the congestion occurred belongs to a different provider than the VoIP service provider. A number of possible communication ways are proposed in this work through which the two service providers can collaborate to mitigate the issues.

- **Development of a DPBM in WMNs** - The packet delay is an important factor in the calculation of the iMOS and its measurement is difficult in situations where the involved nodes are not time-synchronised. This work proposes a DPBM able to provide high precision in the determination of packet delay at the WMN gateway. The developed mechanism also applies to other type of traffic and can be used by a wide range of mechanisms performing actions such as re-prioritization, re-routing, etc.

- **Development of an new enqueueing algorithm based on the Delay Piggy-Backing Mechanism** - The DPBM enables the WMN nodes in the path of VoIP packets to prioritise more delayed VoIP packets over less delayed VoIP packets, thus assuring fair distribution of call quality among the users of the same WMN. The results show that the difference of network delay between all calls is minimized, thus closing the gap between the quality of the best and the worst call.

Following is a list of contributions which support and extend the scope of the work presented as primary contributions:

- **Push-In-First-Out (PIFO) queue implementation in NS3** - The delay-piggy-backing mechanism uses the concept of PIFO queue in order to prioritise packets based on their cumulated queueing delay. The implementation is integrated with the NS3 stack and can be further used in other experiments where such queues are needed.
- **Realistic speech model implementation in NS3** - A set of simulation experiments are performed in order to validate the proposed solutions. While the VoIP applications in most simulators use random numbers to mimic human speech patterns, this work implements an actual human speech model based on a four state model [6].
- **Formula development for considering the human audio recency effect** - An Exponentially Weighted Average (EWA) formula is proposed to be used in weighting the MOS samples with emphasis on the last 30 seconds of the conversation, which is in relation to the human audio memory limitation.

1.4 Thesis Outline

Chapter 2 gives the background to the investigated research topic. An overview on VoIP is presented along with details on the methods used to assess the quality of VoIP calls. A detailed description of the femtocell network architecture is given followed by the description of 802.11-based WMNs and VoIP issues over WMNs.

Chapter 3 presents a literature review discussing existing methods used in assessing VoIP call quality and other solutions regarding the usage of CAC in the context of VoIP. A review of the solutions developed by other works aimed at increasing the VoIP capacity in a given shared network environment is presented towards the end of the chapter.

Chapter 4 describes in detail how to obtain the iMOS in the context of femtocell and WMN deployments. For each of the deployment scenarios, a complete set of simulation results is used to validate the accuracy and usefulness of the iMOS concept.

Chapter 5 presents two iMOS-based CAC solutions proposed in the context of femtocell and WMN deployment. In a Digital Subscriber Line (DSL)-backhauled femtocell deployment scenario, the possible bottleneck of the architecture is pointed out and a solution involving the HeNBGW employing a CAC and resource allocation mechanism is detailed. In a WMN-backhauled femtocell deployment scenario, the benefits of employing a delay piggy-backing mechanism are highlighted in the context of obtaining an accurate iMOS measurement based on which a mechanism takes CAC decisions.

Chapter 6 concludes this thesis with a summary of the solutions proposed and their corresponding results, followed by a discussion of potential future work.

CHAPTER

TWO

BACKGROUND

This Chapter will introduce the various elements of background information that are required to enable the reader to fully understand the contributions of this thesis.

2.1 VoIP

Voice Over IP (VoIP), also known as Internet telephony or Internet Protocol (IP) telephony, is a concept referring to transmitting voice over Packet-Switched Networks (PSNs), such as the Internet. In comparison, typical telephony is carried over Circuit-Switched Networks (CSNs) such as the Public Switched Telephone Network (PSTN).

VoIP involves using a VoIP-enabled device or a VoIP computer program running on a Personal Computer (PC) or a mobile phone. The acoustic voice signal is captured by the microphone and transferred to a voice codec which, at the sender side, has the role to sample the raw analogue audio signal and transform it into a digital stream of data which is further sent to be encapsulated as payload into IP packets. At the receiver side, the codec's role is reversed; the payload received into IP packets is extracted and combined into a digital stream of data which is converted back to an analogue signal and transferred to a speaker.

VoIP is among the oldest application designed for PSNs. The first mention of transporting voice over PSNs was done in the Network Voice Protocol (NVP) in 1973 by the Advanced Research Projects Agency (ARPA) [7]. Interestingly, the idea of VoIP preceded the protocols making VoIP possible nowadays, such as User Datagram Protocol (UDP) (1980) [8], IPv4 (1981) [9], H.323 (1996) [10], Real-Time Transport Protocol (RTP) (1996) [11] and Session Initiation Protocol (SIP) (1999) [12].

From a service provider perspective, the evolution of VoIP has passed through at least three phases:

- Business solutions: VoIP was initially regarded as a solution for companies to decrease the cost of maintaining the networking infrastructure by eliminating the PSTN-like internal network and using only the data infrastructure for both data and voice communications. The connection with the PSTN is made through the company's VoIP Private Branch Exchange (PBX). In this case VoIP is confined to the boundaries of the company's physical network.

- Closed service domain: Skype [13] was among the first to provide VoIP services over the Internet, and chose to use a distributed system based on a peer-to-peer protocol. Skype aims at the large public offering Skype-to-Skype calls for free. However, in order to place a call with a phone number in the PSTN, a fee is applied. Hence, the users are not limited by the boundaries of a network but by the service provider's domain.

- Federated VoIP: typically an Internet domain is used to provide access to webpages or hosting e-mail addresses. Federated VoIP adds functionalities such as VoIP. Similarly to e-mail, federated VoIP needs a dedicated server addressable on that domain. Additionally, the protocols used by federated VoIP are also used to provide communication services like video chat, conferencing, text chat and shared desktop. While the first two types of solutions are somehow limited by either the network or the domain, the federated solution crossed the limitations by enabling a plethora of inter-domain communication services.

2.1.1 VoIP Transport

Transporting voice on PSNs involves many technologies, ranging from signal processing to time-sensitive packet transport protocols. The key aspects of such technologies used by this work are highlighted below.

2.1.1.1 VoIP codecs

Human speech represents changes of the local air pressure in the surrounding atmosphere. In all voice transmissions, these changes are captured by a microphone and transformed into an analogue electrical signal. Any analogue signal has in theory infinite bandwidth, but in practice the bandwidth is limited using filters. The majority of information within human speech can be found in the 100Hz to 4000Hz band. Signal sampling is required in order to transform the analogue signal into a digital data stream. In signal theory, the Nyquist rate [14] specifies that the minimum sampling rate required to avoid aliasing should be equal or bigger than twice the bandwidth of a band-limited signal. Thus, for human speech a usual sampling rate is 8000Hz.

In VoIP, the conversion between analogue and digital domains is done by COders/DECoders (Codecs). A large variety of codecs exists with the main differences between them being the level of compression, bandwidth efficiency, and speech quality. These features require extra processing time at either the coder or decoder side, so that the more complex codecs increase the mouth-to-ear delay of the audio signal. However, this amount of time delay is rather insignificant for human perception. The extra time needed for processing is named *look-ahead* delay.

The concepts of frame and look-ahead delay are present in complex codecs which use compression and Table 2.1 shows the most known audio codecs used in VoIP. Most of the VoIP codecs sample the audio signal at 8000Hz with the exception of Adaptive Multi-Rate Wide-Band (AMR-WB) which samples at 16000Hz; the AMR-WB is considered to be an high definition codec due to its high sampling rate. Table 2.1 also shows that the bitrate of the codecs can range from 4.75kbps for Adaptive Multi-Rate Narrow-Band (AMR-NB), which is very small, to 64kbps for G.711. As such, a frame is composed of multiple samples which are transformed by a compression algorithm into smaller

Table 2.1:
Comparison of VoIP Codecs

Codec	Sampling (kHz)	Bitrate (kbps)	Frame size + look-ahead (ms)	MOS
AMR-NB	8	4.75 ... 12.2	20 + 5.0	4.14
AMR-WB	16	6.6 ... 23.85	20 + 5.0	4.30
G.711 (PCM)	8	64	0.125 + 0.0	4.30
G.723.1	8	5.3, 6.3	30 + 7.5	3.65
G.726	8	16 ... 40	0.125 + 0.0	3.85
G.729	8	8	10 + 5.0	3.92
GSM FR	8	13	20 + 2.5	3.50
GSM EFR	8	12.2	20 + 2.5	3.80
iLBC	8	15.2	20 + 5.0	4.14

number of bytes needed to encode the digital data. The frame size in Table 2.1 refers to the time duration of the audio signal included in one frame. G.711 and G.726 are codecs which do not use compression, thus the frame concept does not really apply and the corresponding frame size values are actually the amount of time captured in one sample period. As no compression is used by G.711 and G.726 there is no look-ahead delay. The far-most right column shows the maximum achievable VoIP call quality which is graded using the Mean Opinion Score (MOS) scale.

G.711 Pulse Code Modulation (PCM) [15] is probably one of the most well known codecs used in many audio applications and it is the main codec used in the PSTN. No compression is used and its data rate is constant during audio transmissions. One of the most popular codecs using compression is G.729 [16]. The bandwidth used by G.729 is smaller than the bandwidth used by G.711, however due to inter-frame dependencies introduced by the encoding algorithm, G.729 and similar codecs are prone to packet loss.

Voice codecs can be made more bandwidth efficient by employing Voice Activity Detection (VAD) algorithms [17]. VAD is an algorithm used by the codec to determine when the microphone captures actual speech or only background noise. When the VAD algorithm determines there is only background noise, the codec will output very small-sized packets basically informing the other party to play background noise, thus reducing the usage of network bandwidth during silent periods.

2.1.1.2 VoIP Transport Protocols

Further is presented how the captured voice samples combined into voice frames are transported over PSNs.

Transmission Control Protocol (TCP) [18] is the most widespread transport protocol used in PSNs. One of the primary features of TCP is guaranteed delivery by using receipt acknowledgement reports sent by the destination party of a transmission, back to the sender. If packet loss occurred during the transmission, the lost packets are re-transmitted thus guaranteeing the integrity of the transmitted data. Another important feature of TCP is its algorithm used to prevent network congestion by dynamically adjusting the transmission throughput. TCP's features can lead to relatively large transmission delays and this is not desired for real-time applications, such as VoIP.

UDP [8] is another widespread transport protocol used by applications where the timely delivery of data is more important than its integrity. Lost packets are not re-transmitted and the destination of the packets does not send receipt acknowledgements. Commonly VoIP and Internet Protocol TeleVision (IPTV) fall in the category of applications using UDP as transport protocol.

Real-Time Transport Protocol (RTP) [19] is a protocol used to uniquely identify a media stream. RTP headers are attached to media packets and contain sequencing information in order to help the destination application rebuilding the data-stream.

Real-Time Control Transport Protocol (RTCP) [19] is the sister protocol of RTP. RTCP provides statistics and control information about the RTP flow it is associated with. The statistics include information such as packet count, lost packet count, Round Trip Time (RTT), and jitter. This information can be used by the real-time applications to adapt the parameters of the media stream to network conditions. Such adaptations may include changing the audio codec or the rate at which the content is being transmitted.

The RTP headers extracted from VoIP packets are used by the E-Model in the calculation of the MOS. A complete description of the MOS calculation process is made in Section 2.1.2.

2.1.1.3 VoIP Traffic Metrics

There are four basic traffic metrics that are widely used by the majority of the Quality of Service (QoS) protocols:

- **Bandwidth** is the amount of traffic (measured in bits) that traverses a point of the network in a given period of time (measured in seconds); the bandwidth is thus measured in bits per second (bps).
- **One-way-delay** is a measure of the end-to-end packet delay as defined by Almes et al. [20] and is comprised of a number of delay factors occurring at each node involved in the delivery of the packet:
  - **Processing delay** is the time needed by the network equipment to process the packet's headers and determine the next required packet action, e.g. the next hop. Another delay due to processing occurs upon packet reception at the physical layer when the received packet is checked for transmission errors.
  - **Transmission delay** is the time required for a sending source to push all the bits of the packet onto the transmission medium; it is tightly correlated with the data-rate of the link.
  - **Propagation delay** is the time spent by the bits of a packet traversing the propagation medium such as copper wires, optical fibre, or the ether.
  - **Queueing delay** is the time spent by a packet in the outgoing queue of a node waiting for the transmission medium to become available.
- **Jitter** is the variation of the packet delay between a particular set of packets belonging to the same packet stream and is described in detail by Demichelis and Chimento [21]. The variation can occur due to changes affecting the network, such as alternative routing or congestion.

- **Loss** is the proportion of packets sent by the source but not received at the destination and is typically measured as a percentage of number of packets lost over the total number of packets sent as defined by Almes et al. [22]. One of the main causes of packet loss is queue overflow; this occurs when the outgoing queue is full at the time of packet arrival. Packet loss can occur also in wireless environments where the propagation medium may suffer from poor transmission conditions caused by channel interference.

All these traffic metrics–bandwidth, one-way-delay, jitter, and loss–are defined by the Internet Engineering Task Force (IETF) which is an organization developing and promoting Internet standards.

Some of the possible types of applications enabled by the PSN and the associated requirements in relation to the metrics discussed above are shown in Table 2.2. It can be seen that VoIP imposes the most strict requirements in terms of one-way-delay, jitter, and loss, but as shown in Table 2.1 the bandwidth requirements for VoIP are low. A streaming traffic type implies that there is no rate control, regardless of the network state, whereas an elastic traffic type adapts its rate to match the network condition. In other words, the two traffic types are characterised by UDP and respectively TCP type of traffic.

Table 2.2:
QoS Requirements per Application (L=Low, M=Medium, H=High)

Application	Bandwidth	Delay	Jitter	Loss	Type
VoIP	L	H	H	H	Streaming
Video Conference	H	H	H	M	Streaming
Streaming VoD	H	M	M	M	Streaming
Streaming Audio	L	M	M	M	Streaming
E-Mail	L	L	L	H	Elastic
File Transfer	M	L	L	H	Elastic

2.1.1.4 VoIP QoS Protocols

The delivery of demanding applications such as the VoIP and video conferencing in Table 2.2 needs to satisfy expectations. The objective of QoS protocols is to ensure that the specified traffic metrics are maintained within acceptable boundaries.

IETF have been mainly involved in the development of protocols designed to provide guaranteed QoS levels, as follows:

- Resource ReSerVation Protocol (RSVP) [23, 24] is a protocol used to carry the reservation request initiated by an application in need for guaranteed resources. The request is carried along the path within the network and each node attempts to meet the specification of the request. RSVP is used by a few of the QoS control architectures such as Integrated Service (IntServ), Differentiated Services (DiffServ), and Multiprotocol Label Switching (MPLS).

- IntServ [25] is an IETF standard designed to provide fine grained flow-based QoS control of traffic within a network. IntServ defines a set of functions used by the nodes in a network to manage per-flow traffic in a coordinated fashion.

- DiffServ [26] is another IETF protocol designed to provide QoS guarantees per-packet. As such, the Differentiated Services Code Point (DSCP) field of the IP header is used to differentiate traffic into different categories. Each node on the path prioritizes packets belonging to certain classes over other classes in order to minimize the delay for time-sensitive applications, such as VoIP.

- MPLS [27, 28] is an IETF standard which attempts to mimic the behaviour of CSNs in PSNs. Functionally, MPLS works by attaching labels to packets which are interpreted by the routers as QoS requirements which further can be translated into selecting a shorter routing path for time-sensitive applications.

2.1.2 VoIP Call Quality Assessment

The International Telecommunication Union-Telecommunication Standardisation Sector (ITU-T) report P.800 [29] introduced a scoring system to assess the quality of a speech that has been transmitted via telephone lines. The score ranges from 1 to 5 as shown in Table 2.3. Traditionally when human test subjects are involved in grading a speech's quality using this scale, the result is the *Mean Opinion Score (MOS)*.

Table 2.3:
MOS scale for subjective assessment

Quality	MOS	Impairment
Excellent	5	Imperceptible
Good	4	Perceptible but not annoying
Fair	3	Slightly annoying
Poor	2	Annoying
Bad	1	Very annoying

Three different methods are available for determining the MOS of a voice call:

- Subjective (S): human subjects are involved in grading the speech's quality;
- Objective (O): does not involve humans to assess the speech but uses an objective model to predict the speech quality based on the differences between the original and the received speech signal;
- Estimative (E): uses a model which considers communication transmission factors, whereas neither the original nor the received signal are considered.

The accuracy of Subjective MOS scores is high as the assessment is done by human subjects by averaging the scores reported by all participants. The major limitation in this case is the scarce availability of human subjects. The process has high cost and obviously cannot be performed in real time. Therefore objective and estimative methods have been developed.

The most popular model for objective speech assessment is Perceptual Evaluation of Speech Quality (PESQ) (ITU-T P.862 [30]). A reference audio sample is injected into a telephone network and the output is recorded. The PESQ methodology compares the reference and the recorded output and the results of the comparison is directly related to the quality of the transmission. Similar to the subjective assessment, the objective assessment can not be used in real-time. These two methods do not capture the influence of the network parameters, such as delay, jitter, and loss, on the speech quality. As previously shown, VoIP quality is correlated with the traffic metrics mentioned above, thus a different method has to be used in order to assess the VoIP quality.

2.1.2.1 The E-Model Algorithm

The most popular estimative model is based on transmission parameters and is the E-Model (ITU-T G.107 [31]). The E-Model algorithm is an ITU-T standardised computational model for subjective call quality assessment [31]. It is widely accepted as an accurate tool for transmission network planning. The E-Model operates under the assumption that perceived quality impairments are additive. By combining both codec and network impairments, it produces the *Transmission Rating Factor (R)* (Equation 2.1).

$$R = R_0 - I_s - I_d - I_{e\_eff} + A \tag{2.1}$$

Equation (2.1) is comprised of following elements:

- R_0 represents the signal-to-noise ratio obtained by considering the circuit and room noise;
- I_s is a combination of all impairments which occur more or less simultaneously with the voice signal, such as quantization or loudness level;
- I_d represents the impairments caused by the delay of voice signals such as talker echo, listener echo, and absolute signal delay (mouth to ear delay);
- I_{e_eff} represents the impairments caused by low bit-rate codecs and packet loss;

- A is the advantage factor and represents the user's willingness to accept lower call quality in exchange for the advantage of access.

Equation (2.1) takes into account many of the traditional telephony parameters which cannot be measured in a VoIP call such as R_0 and I_s. One of the first studies on the E-Model with emphasis on voice transmitted over PSNs was done by Clark et al. [32]. Clark et al. made the observation that many of the E-Model parameters can be assumed using the default values as specified by [31]. As such, the difference $R_0 - I_s$ takes the effective value of 94 and Equation (2.1) takes the following form:

$$R = 94 - I_d - I_{e\_eff} + A \tag{2.2}$$

The E-Model is called upon the receipt of every VoIP packet. It is known that the accuracy is low for the R values obtained from the first arrived VoIP packets. This is influenced by the low measurement accuracy of packet loss and jitter, which are metrics based on the link between consecutive packets.

The actual NS-3 implementation of the E-Model used in the simulations carried out in this work, is presented in Appendix A.1. The remaining terms of Equation (2.2) are detailed below.

2.1.2.2 *A*

The advantage factor A is the only additive parameter in the E-Model's formula. A represents the user's willingness to accept lower call quality in compensation for being able to place calls in unusual circumstances, for example being in a remote geographical area.

Based on information found in the E-Model [31], Table 2.4 shows the maximum values A can take depending on the call use-case. Further details on how A's values will evolve, based on the technological progress in telecommunications, is presented in [33].

Table 2.4:
Recommended values for the Advantage Factor

A	**Type of Communication System**
0	Wired connections (e.g. PSTN)
5	Low speed mobility (e.g. indoors)
10	High speed mobility (e.g. vehicle)
20	Remote areas (e.g. multi-hop satellite connection)

2.1.2.3 Id

I_d is related to the impairments caused by the delay between the time when the sound is injected in the communications system and the time it reaches the end-point where it is played out. In telephone systems this delay is usually called mouth-to-ear delay.

In addition to the network delay described in Section 2.1.1.3, there are two more delays introduced in VoIP calls:

- Coding/Packetization delay – is the delay introduced at the source of the VoIP packets by the codec; it represents the time needed to perform the conversion of the analogue signal into digital, the look-ahead delay, and the time needed to compose the VoIP packet.
- Decoding/Jitter Buffer Delay – is the delay introduced at the destination of the VoIP packets by the decoder in order to decompose the VoIP packets and extract the payload, decode the payload into audio samples, and eventually the waiting time imposed by a de-jitter buffer; the de-jitter buffer is acting like a queue for audio samples with the aim on reducing the influence of network delay variation on the play-out quality.

The ITU-T recommends a total mouth-to-ear delay lower than 150ms [31], in order to preserve interactivity in a duplex phone call. Values higher than 150ms start to degrade the call quality, making interactivity impossible for values above 400ms.

2.1.2.4 Ie-eff

I_{e_eff} is reflecting the effect of packet loss on the voice quality. For some of the lower rate codecs which introduce inter-packet dependencies the effect of the packet loss is exacerbated. In the E-Model this is denoted with I_e. Typical values of I_e for three of the most used codecs are: 0 for G.711, 11 for G.729, and 5 for AMR-NB.

Some codecs have a feature called Packet Loss Concealment (PLC) which is an interpolation process used to artificially reconstruct lost packets using previous and later audio content. In the E-Model this codec feature is captured by the Bpl factor. Typical values for Bpl for three of the most used codecs are: 4.3 (without PLC) or 25.1 (with PLC) for G.711, 19 for G.729, and 10 for AMR-NB.

Finally, the I_{e_eff} can be calculated as a combination of I_e, Bpl, and percentage packet loss (Ppl)[31]:

$$I_{e\_eff} = I_e + (95 - I_e) \cdot \frac{Ppl}{Ppl + Bpl} \tag{2.3}$$

2.1.2.5 R to MOS

Based on subjects' involvement in the actual speech, there are three situations possible:

- listening-only (L), when the subjects assess the speech coming out from the speaker;
- talking (T), when the subjects grade the talking side only, which for example can be influenced by the echo signal;
- and conversational (C), when subjects are involved in an active conversation.

In order to obtain the MOS for the Estimated Conversational speech Quality (MOS_{CQE}), a conversion formula is provided in [31] which converts R values to MOS:

$$MOS_{CQE} = \begin{cases} 1 & \text{if } R < 0, \\ 1 + 0.035 \cdot R + 7 \cdot 10^{-6} \cdot R(R-60)(100-R) & \text{if } 0 \leq R \leq 100, \\ 4.5 & \text{if } R > 100. \end{cases} \tag{2.4}$$

The conversion from R to MOS is also given in Table 2.5 where the quality categories are divided into steps of 10 on the R scale in the range from 50 to 90 (lower limit).

Table 2.5:
R to MOS_{CQE} correspondence for estimative assessment

R-value (lower limit)	**MOS (lower limit)**	**User Satisfaction**
90	4.34	Very satisfied
80	4.03	Satisfied
70	3.60	Some users dissatisfied
60	3.10	Many users dissatisfied
50	2.58	Nearly all users dissatisfied

Figure 2.1 depicts the influence of the delay on the MOS for three of the most used codecs for VoIP. It can be seen that the lower the codec's rate, the lower its MOS is (confirmation from Table 2.1).

Figure 2.2 depicts the influence of the packet delay variation or jitter, on the MOS for the same three codecs. The same order between the codecs is maintained as in Figure 2.1.

Figure 2.3 depicts the influence of the packet loss on the MOS for the same three codecs with the difference that G.711 can be used with its PLC feature enabled or disabled. Here the trend changes compared to the trend presented for the delay and jitter. Specifically, G.729 performs better than AMR and G.711 without PLC (Bpl=4.3) given the fact that G.729 is a very low rate codec. However, G.711 with PLC (Bpl=25.1) outperforms the other codecs when its PLC feature is enabled.

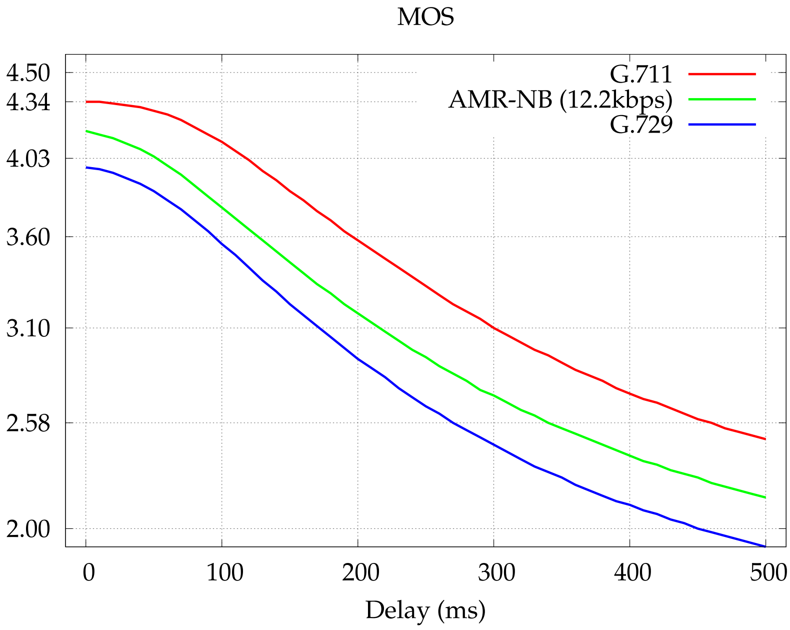

Figure 2.1: MOS vs. Delay per codec

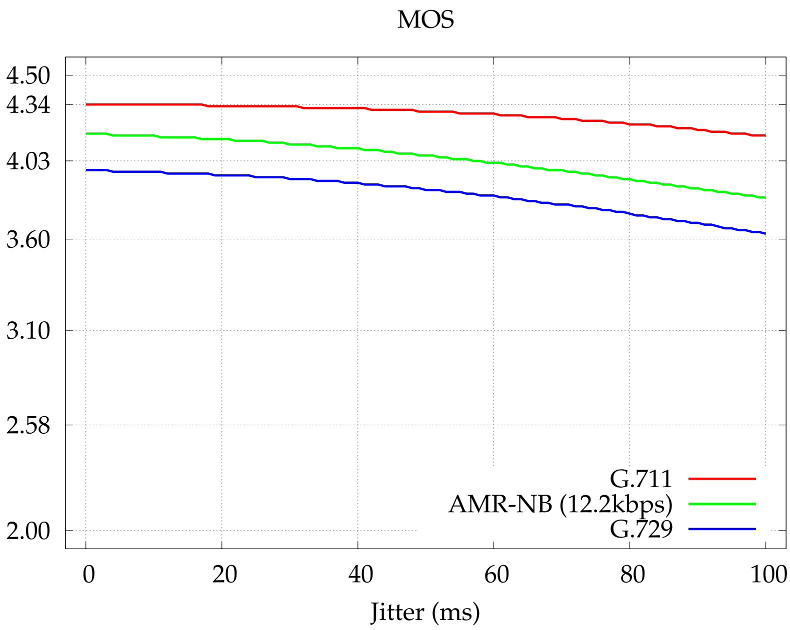

Figure 2.2: MOS vs. Jitter per codec

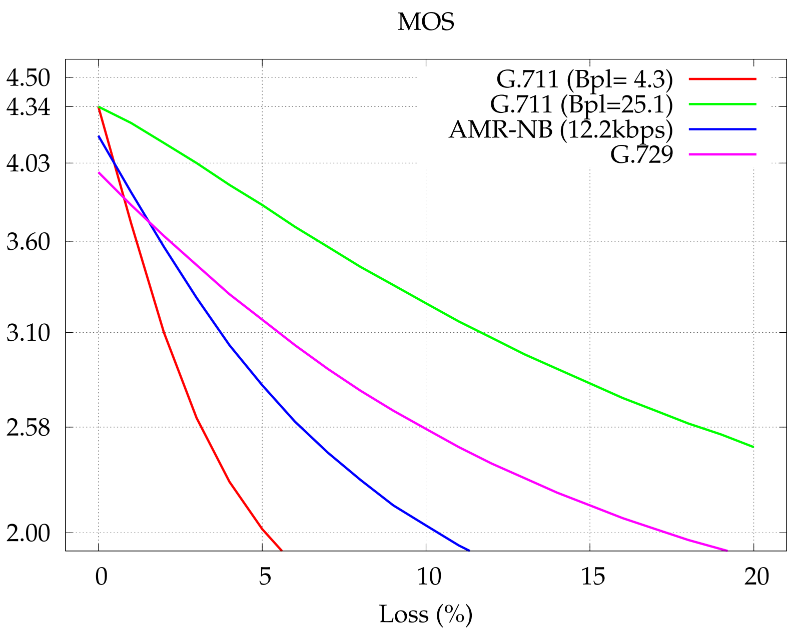

Figure 2.3: MOS vs. Loss per codec

2.1.3 Speech Model

It is not feasible to test the call quality of a phone conversation using human subjects, as this solution is not scalable. As a consequence of this limitation, the conversations have to be either simulated or synthesised in test-beds. For testing Variable Bitrate (VBR) codecs such as AMR and G.729, the VoIP packet traffic pattern–and consequently the call quality–is highly influenced by the human conversation pattern. The conversation model from ITU-T's P.59 report [6] mimics the pattern of a conversation between two humans. This model improves the accuracy of the results obtained from tests where realistic conversations are needed.

Figure 2.4 is an example of a 50 seconds full-duplex conversation based on the ITU-T's conversation model. The model is a four state transition model; the states are: *Single Talk*$_1$ (A-talking, B-silence), *Single Talk*$_2$ (A-silence, B-talking), *Double Talk*, and *Mutual Silence*. The period length of a state and the probabilities determining which state to change to, have been extracted from actual human conversations. This model was implemented and incorporated into the network simulators used in this work, in order to generate realistic

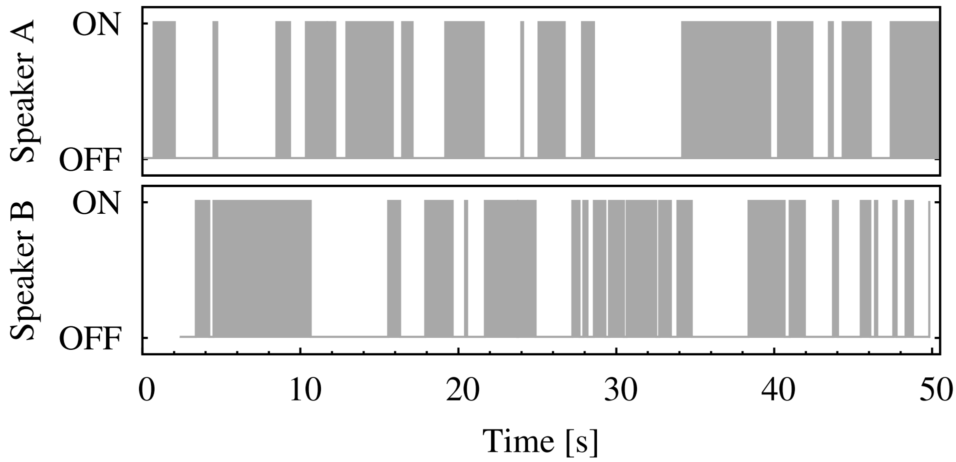

Figure 2.4: Conversation Model from ITU-T/P.59

speech activity for full-duplex VoIP calls.

The actual NS-3 implementation of the ITU-T P.59 speech model is presented in Appendix A.2.

2.2 Femtocell Networks

Typically in cellular communications the wireless link is provided to the User Equipment (UE) by a cell-tower deployed in the area where the user roams. The range of the cell-tower can cover an important area and the transmission power is high enough to penetrate walls in order to provide indoor service coverage. However, the indoor service suffers because of the signal attenuation caused by the building. One possible solution to that is deploying the cellular antenna indoors and reducing its transmission power. Although in the early days of cellular communications this solution remained at the concept stage, nowadays such solution exists and is commonly known under the term of *femtocell*.

Femtocells are small-range, fully-featured, low-power cellular base stations. Typically femtocells are deployed for home or small-office usage. Compared to macro-towers, femtocells have:

- shorter range, typically tens of meters;

- lower transmission power;
- lower energy consumption;
- lower number of served users, typically around ten users simultaneously;
- the backhaul is provided by the user, and is typically out of the femtocell service provider's control.

Historically the femtocell concept was revealed by Alcatel in 1999 and 3 years past until Motorola built the first prototype of a Third Generation (3G) home base station in 2002. In 2003 *picoChip*[1] created the first commercial 3G chipset which was used a year later by Ubiquisys[2] and 3Way Networks[3] to build commercial 3G home base stations. It was in 2005 that the term *femtocell* was first used instead of *home base station*. The femtocells were the main focus of the 3GSM conference in 2007 and 2008. Although femtocells were available since 2004, only in 2008 Sprint USA[4] commercialised these for the public.

It is important to see the femtocells in the context of cellular standards for a better overview on the subject. As such, the first generation of cellular communications used analogue signals and was followed by the digital cellular system of the second generation of cellular communications. The Global System for Mobile Communications (GSM) is among the well known digital cellular systems developed in Europe by the European Telecommunications Standards Institute (ETSI). The 3G standards comply with the International Mobile Telecommunications-2000 (IMT-2000) specifications developed by the International Telecommunication Union (ITU). Among the deployed 3G compliant systems the Universal Mobile Telecommunications System (UMTS) and Code Division Multiple Access 2000 (CDMA2000) systems are the best known. The UMTS system uses the Wideband Code Division Multiple Access (WCDMA) air interface in Europe and Time Division-Synchronous Code Division Multiple Access (TD-SCDMA) in China, and the CDMA2000 system uses Code Division Multiple Access (CDMA) and Interim Standard 95 (IS-95). The 3G standards improved the data access for mobile clients to packet access. High

[1] www.picochip.com
[2] www.ubiquisys.com
[3] 3Way Networks was acquired by Airvana - www.airvana.com
[4] www.sprint.com

Speed Downlink Packet Access (HSDPA) and High Speed Uplink Packet Access (HSUPA) were developed to offer data access at 21 Mbps for downlink and respectively 11 Mbps.

Fourth Generation (4G) improved the 3G system, by bringing enhancements to the UMTS system architecture and changes to the air interface. Specifically the new architecture is shifting from its CSN behaviour to a PSN behaviour. This shift transformed circuit-switched voice calls into VoIP calls. This new cellular system is commonly known as Long Term Evolution (LTE) and its architecture was designed with higher emphasis on the packet access as can be seen in the following section.

2.2.1 LTE Architecture

The LTE architecture diagram is depicted in Figure 2.5, as described by the 3rd Generation Partnership Project (3GPP) in their Release 8. The architecture can be split into three domains:

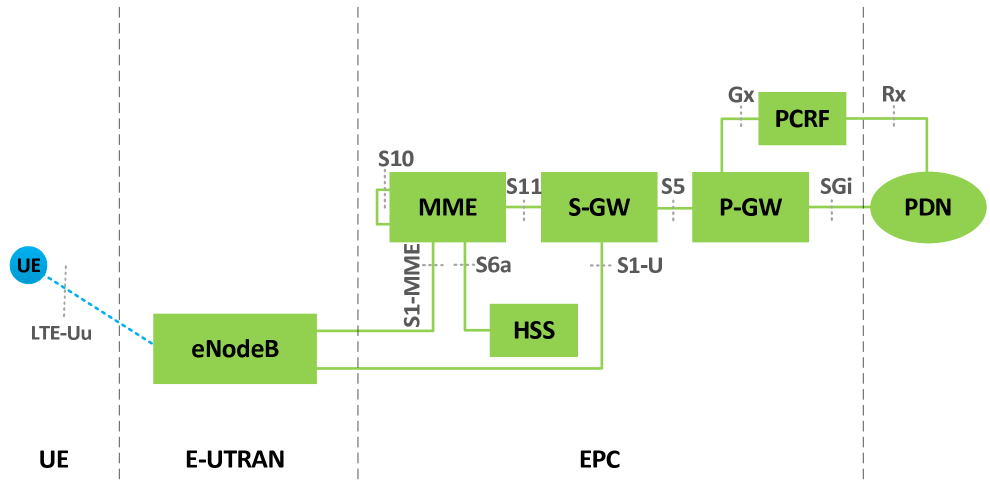

Figure 2.5: LTE Architecture

- **User Equipment (UE):** is comprised of the mobile devices enabled to work on the LTE-Uu interface connecting the UE to the evolved NodeB (eNB); the LTE standard changed the characteristic of the air interface from WCDMA to Orthogonal Frequency-Division Multiple Access (OFDMA) for the downlink and Single-Carrier Frequency-Division Multiple Access (SC-FDMA) for the uplink;

- **Evolved Universal Terrestrial Radio Access Network (E-UTRAN):** is composed only form the eNBs to which the UEs connect. An eNB, is the basestation, or cellular tower, covering a relative large area with cellular connectivity.

- **Evolved Packet Core (EPC)**: is comprised of multiple entities responsible for the overall control of the UEs:

  - the Mobility Management Entity (MME) performs the processing of the control messages flowing between the UEs and the EPC. Here the establishment, maintenance and release of the bearers is handled together with performing security management of the connections.
  - the Home Subscriber Server (HSS) is maintaining the information about the user's subscription such as the QoS profile and the Packet Data Networks (PDNs) the user can connect to.
  - the Serving Gateway (S-GW) is the anchor-node for data services provided to UEs moving between eNBs. In addition, the S-GW performs the administrative function of collecting information for billing and lawful interception.
  - the PDN Gateway (P-GW) is the entity responsible with the allocation of IP addresses for the UEs and also for performing filtering of downlink data into different QoS bearers.
  - the Policy and Charging Rules Function (PCRF) is responsible for taking decisions related to policy control and also for controlling the flow-based charging functionalities of the Policy and Charging Enforcement Function (PCEF) (not shown) residing on the P-GW.

2.2.2 LTE Femtocell Architecture

Figure 2.5 described the LTE architecture and Figure 2.6 shows the position of the LTE femtocell in the LTE architecture. In 3GPP parlance, the femtocell is referred to in the LTE standard by Home evolved NodeB (HeNB) which

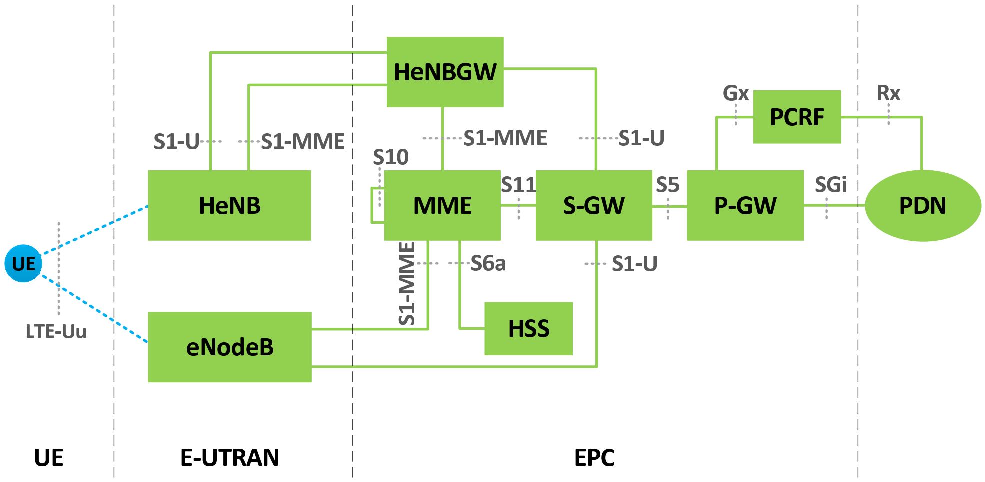

Figure 2.6: LTE Femtocell Architecture

is a term evolved from the previously used Home NodeB (HNB) in UMTS femtocells.

In Figure 2.6 the UE can connect to either the HeNB or eNB usually based on a strongest signal criteria. When connected to the HeNB the user typically notices an increase of the received signal strength. However, what the user does not necessarily notice, is the fact that the backhaul changed.

The HeNB requires to be provided with an Internet connection usually available at the user's premises. Since this type of backhaul is considered to be insecure for sensitive communication, such as voice calls, the LTE standard requires this communication to be done through a secure tunnel. As such, an IPsec tunnel is established between each HeNB and the Home evolved NodeB Gateway (HeNBGW) in order to securely transfer the data or voice packets between the UEs and the EPC.

The IPsec tunnel is terminated at the HeNBGW which is the entry point into the EPC. The HeNB traffic can be split in two main categories: the user- and the control-plane. From the HeNBGW the control-plane traffic is forwarded to the MME over the S1-MME interface, and the data-plane traffic is forwarded to the S-GW over the S1-U interface. For Internet access the S-GW routes the uplink requests over the S5 interface to the P-GW which further forwards them over the SGi interface to the PDN. The downlink traffic flows in the reverse order to the uplink traffic.

The LTE femtocell protocol stack of the user-plane is depicted in Figure 2.7. The UE encapsulates the payload from the application into IP packets and forwards it down the stack to the layers specific to the LTE-Uu interface. The packets are further sent over the wireless channel to the HeNB from where the packets are treated as normal IP packets. These are securely sent on the S1-U interface over the Internet to the S-GW in the EPC.

The application which is enabled to transmit and receive packets using the protocol stack described above can be any application residing on the UE. Typically such application is in the category of communication applications, such as e-mail, web-pages, video content, audio content, etc. The main application though is the call application, which is used by the user to place voice calls. The application uses a codec which works as described in Section 2.1.1.1 to obtain the voice in a digital format suitable for network transmissions. The support of the Adaptive Multi-Rate (AMR) codec is a requirement in the current LTE standard.

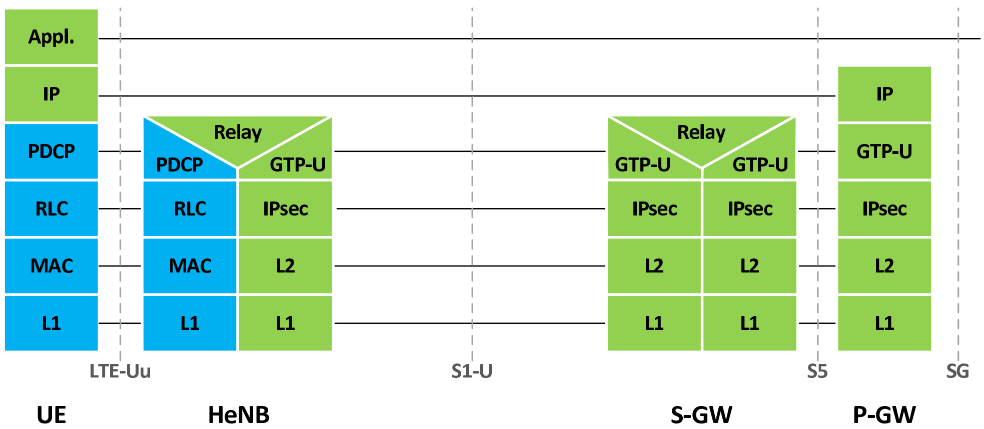

Figure 2.7: LTE Femtocell User-Plane Protocol Stack

The LTE femtocell protocol stack of the control-plane is depicted in Figure 2.8. Similarly to the user-plane, the control-plane communicates with an entity in the EPC, i.e. the MME. The payload in this case is comprised of signalling messages between the UE and the EPC described by the Non Access Stratum (NAS) protocols [34].

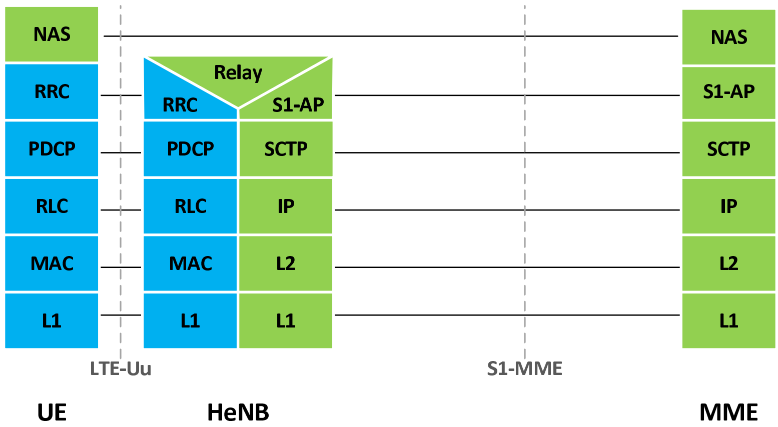

Figure 2.8: LTE Femtocell Control-Plane Protocol Stack

Figure 2.6 shows the LTE femtocell architecture where the UE sends and receives traffic via the HeNB by traversing a backhaul network placed between the HeNB and the HeNBGW. This backhaul network is typically a part of the Internet as the user plugs the HeNB to the Internet connection provided by an Internet Service Provider (ISP). However, when the user accesses the Internet services via the HeNB, the traffic thus generated has to traverse the EPC. It is obvious that traversing the Internet to reach the EPC in order to have access to the Internet is not optimal from a few points of view:

- User's view: the user already pays the Internet bill and it would be unreasonable to pay again for the Internet access when using the HeNB; this is usually solved by the subscription type which instructs the PCEF not to charge the user for data access when using the HeNB. However, sending all traffic thought the EPC can only increase the latency on the IP services.
- HeNB provider's view: data services are charged when the user is connected to the eNodeB, but not charged when using the HeNB; still EPC resources are used to serve the user, and no revenue is made.

In order to alleviate the issues described above Zdarsky et al. [35] have proposed a modified LTE femtocell architecture in which a new network entity can be deployed to proxy some functionalities of the EPC locally, more close to the location where the femtocells are deployed. Zdarsky et al. target the enterprise use-case where multiple femtocells can serve an entire office-building. Their proposed architecture is depicted in Figure 2.9. It can be seen that two of the EPC functionalities have been proxied to the new entity called Local Femto Gateway (LFG):

- The proxy MME intercepts and manipulates signalling messages on the S1-MME interface and controls the proxy S-GW function. In this way, mobility can be handled locally thus offloading the EPC from having to handle mobility events caused by in-building roaming.
- The proxy S-GW intercepts the data traffic flowing on the S1-U interface and re-routes some of the flows to the local IP services or directly to the Internet.

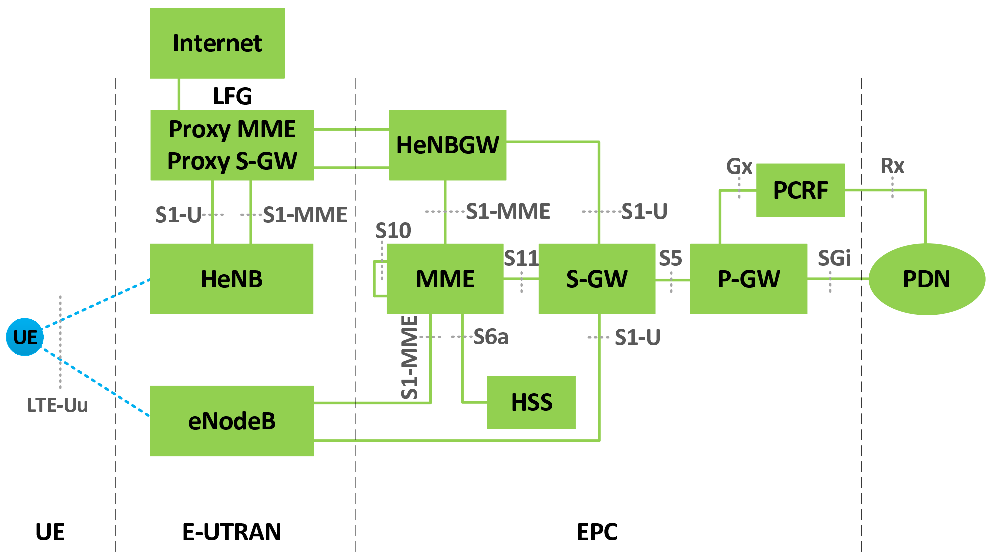

Figure 2.9: LTE Local-Femto Architecture

2.2.3 Femtocell Backhaul

Based on the connection type, the femtocell backhauls can be divided in wired and wireless backhauls. Wired backhaul links are provided to femtocells deployed in residential premises of Small Office/Home Offices (SOHOs). Wireless backhaul links are provided to femtocells deployed in a variety of cases, such as very remote areas, where a physical link could not reach, e.g. tropical forests.

Following are the best known and widely deployed wired backhauls:

- Digital Subscriber Line (DSL): is an Internet access technology built on the infrastructure which was already deployed to provide fixed telephony services.
- TV cable: is another Internet access type built on the infrastructure used by the TV service providers. The speeds of these type of connections are higher than the speeds achievable via the xDSL infrastructure.
- Fiber To The Home (FTTH): is an Internet access technology based on optical communications. Typically the optical fibre reaches only the distribution points placed close to user premises. Copper-wired connections, such as Ethernet cables, are further used for the last few meters needed to reach the user's home router.

While the wired backhaul solutions are mainly used for home and SOHO deployments, the wireless backhaul solutions are gaining momentum. Various opportunities and use-cases started to emerge as the commercialization of femtocell reached enough maturity.

The wireless backhaul solutions can be terrestrial or satellite-based. The terrestrial backhauls can be also split in two main categories based on the frequency range:

- Non Line-of-Sight (NLOS): wireless communications, when there is no line-of-sight between the sender and receiver, are possible in bands allocated at frequencies lower than 6 GHz; this category includes standards such as 802.11a/b/g and WiMAX.
- Line-of-Sight (LOS): when the frequency is higher than 6 GHz, material absorption is nearly 100% hence the antennas have to be in the line-of-sight of each other to be able to communicate.

The satellite backhaul solutions have become more available due to a decrease in usage price. As such, a satellite connection can be used to provide backhaul for femtocells deployed in very remote places where deploying Internet via copper-wire or optical fibre would have a bigger cost than the satellite solution. A special case were this kind of deployment is clearly beneficial is the disaster-recovery scenario. Femtocells can be deployed on intervention vehicles and the backhaul is provided through a satellite connection.

2.3 Wireless Mesh Networks

Wireless Mesh Networks (WMNs) are networks composed of multiple nodes with wireless links connecting them in a combined effort to extend the range of wireless services in a geographical area. The nodes involved are usually equipped with multiple wireless interfaces to facilitate user access and provide backbone connectivity.

Although this wireless access technology existed for more than a decade, WMN deployments have only gained momentum in the past few years, once the price for embedded hardware and wireless chipsets dropped considerably.

The prevalent wireless access technology used in WMNs is in unlicensed spectrum. The most known such technology is described in the 802.11 family of protocols. 802.11 protocols are nowadays supported by any new built smart-phone, laptop, tablet, or other mobile devices. Apart from 802.11 which is commonly known as Wireless Fidelity (WiFi), 802.15 and 802.16 protocols can be also used as underlying technology for WMNs. 802.15 is the protocol family describing Personal Area Networks (PANs) which are of short range and the most commonly know technologies from this family are Bluetooth and ZigBee. 802.16, on the other hand, standardized communications for much larger range than 802.11 and 802.15 and the most known technology belonging to this family is Worldwide Interoperability for Microwave Access (WiMAX).

It is important to note that neither of the above mentioned protocols, 802.11, 802.15 and 802.16, were initially designed to support meshing functionalities. As the WMN paradigm gained momentum, working groups have been created around those protocols to amend these protocols with meshing functionalities.

2.3.1 WMN architecture

2.3.1.1 Components

The nodes involved in a working WMN can be split in two categories and are depicted in Figure 2.10:

- **The mesh routers** are the nodes which constitute the WMN backbone infrastructure; there are three main functionalities a mesh router may have:
  - relay: each mesh router has to be equipped with at least one interface through which it can collaborate with other nodes to create the backbone infrastructure;
  - gateway: nodes equipped with an interface through which access to external networks can be established; typically, this is done via a wired interface;
  - access point: grants wireless access to roaming users.
- **The clients** are the users of the WMN which are able to connect to the mesh routers which have enabled the *access point* functionality.

2.3.1.2 Interfaces

The number of interfaces and the technology used by these can vary from deployment to deployment. Studies [36] have shown that multiple interfaces installed in the mesh routers are beneficial for the backbone infrastructure. However the topology plays an important role in the overall performance of the WMN.

A common problem in a WMN is interference from neighbouring nodes. The interference can be diminished by using different wireless channels when communicating with different nodes. This can be achieved with multiple interfaces installed on a mesh router and employing a wireless technology which features multiple orthogonal channels.

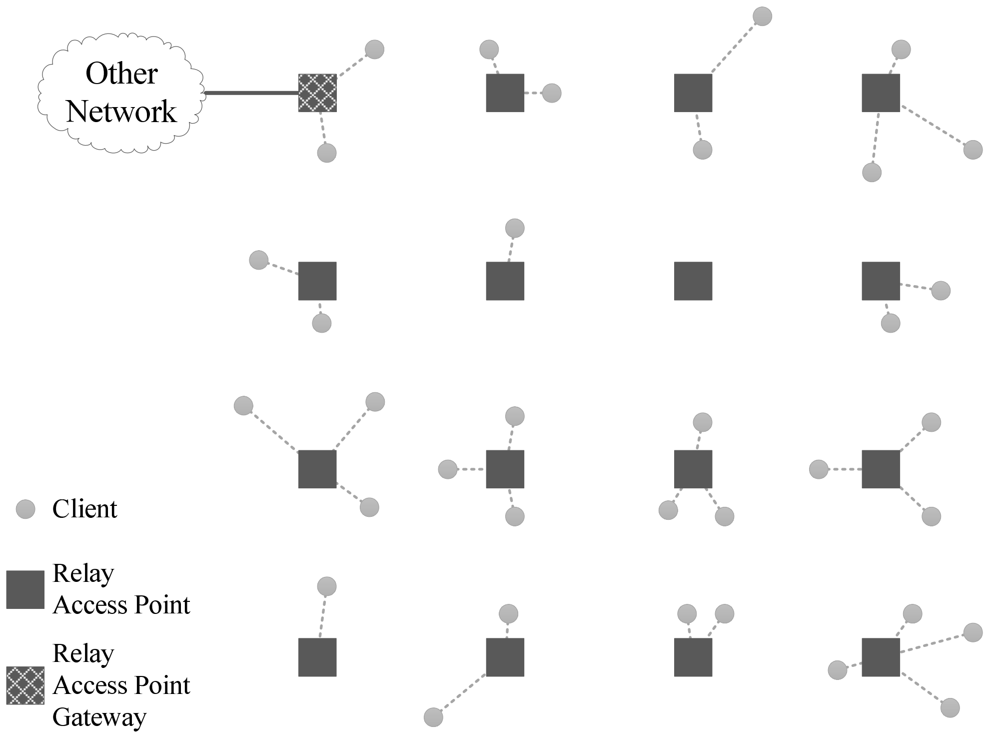

Figure 2.10: WMN Architecture

The 802.11 family provides a variety of protocols which evolved over time becoming the de-facto technology in most of the mobile equipment built nowadays. Table 2.6 shows the main features of the members of the 802.11 wireless family. It can be seen that the maximum achievable data rate increased over time, now reaching speeds unimaginable at the time of the first member of the standard family in 1997. With the introduction of 802.11n [37] in 2009, the Multiple-Input/Multiple-Output (MIMO) feature was added to the standard. MIMO basically enables two nodes to exchange information using multiple interfaces simultaneously. This new addition allowed a boost in data rate, however the main driver of this concept is the decrease in manufacturing price of the wireless routers.

2.3.1.3 Topologies

The physical deployment of WMNs involves the placement of mesh routers in the area where the service is made accessible. There are many possible deploy-

Table 2.6:
Evolution of 802.11 Wireless Technology

Year	Standard	Frequency (GHz)	Channel Bandwidth (MHz)	Max Data Rate (Mbps)
1997	802.11	2.4	20	2
1999	802.11b	2.4	20	11
1999	802.11a	5	20	54
2003	802.11g	2.4	20	54
2009	802.11n	2.4 5	20 40	1x1: 150 2x2: 300 3x3: 450
2013	802.11ac	5	20 40 80 160	2x2 (80MHz): 866 4x4 (80MHz): 1733

ment topologies and actual deployments are influenced by the geographical surroundings of the served area. However, many of the deployment topologies are developed from geometrical forms, such as lines, triangles, squares and hexagons. Random placement of mesh routers has been proven inefficient by Robinson and Knightly [36].

In the research community, the topologies chosen for numerical analysis or simulation works, are mainly geometrically aligned topologies, as depicted in Figure 2.11. The line topology is used when demonstrating the interference range or measuring the network's throughput capacity. The line topology is preferred also in scenarios where the routing does not play an influencing role. The more complex topologies, such as triangle and squares are used to provide multiple routing paths, thus focusing on the ability of different routing protocols to choose the optimum paths based on the traffic and network conditions.

The position and number of gateways plays an important role in the performance of a WMN. Many studies such as [38] have shown that any complex

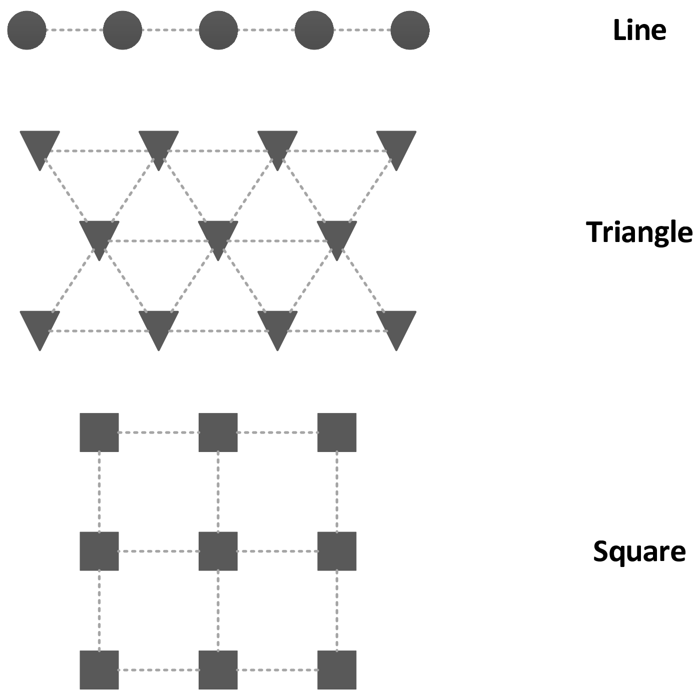

Figure 2.11: Types of WMN Topologies

topology can be simplified to a subset of the network served by only one gateway.

2.3.2 Routing

Any WMN needs a routing protocol which provides the necessary information to each node which has to send or forward traffic to a specific destination. Typically the necessary information any node needs for routing is the next hop to which a specific packet needs to be directed. This routing information is stored in the routing table of the node.

There are two different approaches employable by a node in a WMN in order to populate its routing table:

- Reactive routing – the node starts to flood the network with requests on information about a specific destination in order to discover the path.

The neighbouring nodes which receive the requests may reply directly, if the route for the destination is known, or forward the request, if the path is not known yet. When the path to the destination was built in this way, the packet flow starts. An obvious drawback of reactive routing is the relatively large route discovery time which introduces an initial delay to the transmission.

- Pro-active routing – one node in the network is selected to act as the root node which regularly initiates the procedure of checking for neighbours, discovering all possible destinations in the network, and refreshing the routing table of all nodes. In this way no initial/additional delay is caused, but there is an important traffic overhead caused by constant polling.

Examples of known reactive routing protocols are AODV [39], DSR [40], MSR [41], SrcRR [42]. Examples of pro-active routing protocols are OLSR [43], BATMAN [1] and BABEL [44].

2.3.3 Quality of Service in 802.11

As multimedia is a popular content consumed by mobile users nowadays, studies have shown that multimedia traffic needs to be guaranteed in terms of delay, packet loss, and jitter. Subsequently many standards were developed in this regard. Section 2.1.1.4 presented different QoS architectures and different protocols designed to offer QoS support for different traffic types.

In the 802.11 wireless environments the QoS parameters are directly affected by the quality of the packet transmission. The IP packets are wrapped into 802.11 Medium Access Control (MAC) frames. The MAC protocols are in control of defining timing and scheduling of sending events. A common problem in 802.11 environments is that the probability of two nodes sending in the same time is not zero, thus leading to collisions. A collision means the medium is being used simultaneously by two nodes and the resulting signal is a compound which can not be used by the receiver.

The 802.11 protocol [45] which specifies the parameters and scheduling of the transmission is the Distributed Coordination Function (DCF). The

[1] http://www.open-mesh.org/projects/open-mesh/wiki

DCF treats all IP traffic equally. It implements a timing system of which all 802.11-compliant nodes must be aware. Nodes must sense the wireless channel before transmitting according to specifications mentioned in the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. If the medium is busy, the node will continue to sense the medium until it becomes idle. If the medium is idle, the node can send a frame after waiting a randomly chosen backoff time. The backoff time is uniformly distributed over the interval $[0, CW - 1]$ where the Contention Window (CW) is specified in number of time-slots. The initial value of the backoff time is specified by CW_{min}. The backoff counter is decremented after each time-slot in which the medium was found to be idle. If, after the backoff time elapsed, the medium is busy, the backoff time is doubled but not allowed to exceed a certain threshold specified by CW_{max}. Else, if the backoff time elapsed and the medium is not busy, the node can access the medium.

2.3.3.1 802.11e

DCF was not initially designed to support different traffic types, hence all traffic is treated equally. All nodes have the same chance of acquiring access over the transmission medium regardless of how time-sensitive is the traffic which is transmitted. The 802.11e [46] protocol was designed to alleviate this shortcoming.

The DCF mechanism was amended by the Enhanced Distributed Channel Access (EDCA) mechanism which introduced different treatment for packets belonging to different traffic types. The traffic type of a packet is specified by the application layer before sending the packet to the lower levels. This is clearly specified in the Type-of-Service (ToS) field of the IP packet header. The field is interpreted by the EDCA at the MAC layer and mapped into an Access Category (AC) as shown in Table 2.7.

The EDCA mechanism specifies different values for the CW_{min} and CW_{max} per AC. Each AC has its dedicated packet queue and each queue works in the same way as described for the DCF but with different parameters. In this way, virtual contention between different queues of the same node is possible if the backoff timer of two queues of the same EDCA elapse simultaneously.

Table 2.7:
EDCA Access Category Mapping

Traffic Type	Priority	AC
Best Effort	0	0
Best Effort	1	0
Best Effort	2	0
Video	3	1
Video	4	2
Video	5	2
Voice	6	3
Voice	7	3

The queue with the highest AC gets priority over the lower-priority queues which treat this event as a busy channel situation.

2.3.3.2 Frame Aggregation in 802.11e

The aggregation mechanism is based on encapsulating multiple packets or frames into a single one. This approach considerably increases the performance and capacity of WMNs by combining frequent transmissions of small-payload packets into less frequent transmissions of a larger payload. Frame aggregation was initially specified by the 802.11e standard [46] and was further improved by the 802.11n standard [37]. Figure 2.12 shows the structure of a normal and an aggregated 802.11 MAC frame. An 802.11 MAC frame can carry an Aggregated MAC Service Data Unit (A-MSDU) which encapsulates MAC Service Data Units (MSDUs), each MSDU having its own sub-frame header.

An end-to-end or a hop-by-hop scheme is usually used for aggregation. In an end-to-end approach, the aggregating node selects from the outgoing queue only those packets having the same destination. The aggregated packet is not de-aggregated until it reaches the destination. This method gives good results when the network is highly loaded.

In the hop-by-hop approach, the aggregating node selects from the queue those packets having the same next-hop. At each hop, the aggregated frame

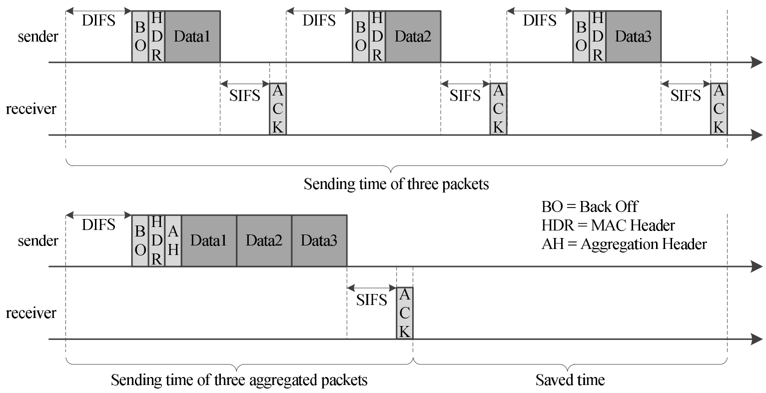

Figure 2.12: Frame Aggregation in a Wireless Medium

is de-aggregated and the resulting packets are again considered individually. This approach increases the complexity and processing load on the mesh nodes, but yields better results considering all possible network load conditions. In Figure 2.12, it can be observed that the sub-frame headers contain the source and destination address, meaning that the 802.11 frame aggregation feature supports the hop-by-hop aggregation scheme.

The aggregation mechanism is triggered by monitoring threshold crossing. One possible threshold is the maximum number of bytes an A-MSDU can have. For example, if the queue has available for aggregation enough packets to fill an A-MSDU, then the process is triggered and the A-MSDU is made available at the future moment when the MAC layer senses the wireless medium as idle. Another possible threshold is the moment a countdown timer expires; a new aggregation process is triggered regardless of whether the queue has packets available for aggregation or not.

2.4 Summary

This introductory chapter highlighted some of the background pre-requisites needed in order to follow the discussion later in this Thesis.

Three main topics have been discussed in this chapter, namely VoIP, Femtocell Networks, and Wireless Mesh Networks.

The history of VoIP highlighted this Internet application as one of high interest in the area of communications, with emphasis on the Quality of Service aspects affecting VoIP call quality. An entire section was dedicated to measuring VoIP call quality. The presented E-Model accounts for all the networking factors influencing the VoIP call quality. The model is a fundamental component of the work developed in Chapter 4.

Further in this chapter two attractive wireless access technologies, that are currently raising interest for both the users and the service providers, were presented. In the licensed wireless spectrum, the LTE femtocell-based wireless access network and its architecture was presented, and the 802.11-based Wireless Mesh Networks were presented as the candidate of the unlicensed spectrum, with emphasis on VoIP as the application under consideration. These are the networking schemes explored in Chapter 5.

The concepts presented here are also needed for a complete understanding of the literature review presented in Chapter 3.

CHAPTER

THREE

LITERATURE REVIEW

This Chapter discusses the related work found in the literature on packet delay measurement and estimation, VoIP monitoring, and VoIP CAC mechanisms.

3.1 Delay Measurement and Estimation

This section describes the related works done in the area of measuring the network delay of IP packets. This review is in relation to the work developed here in the area of network delay measurement methods, which are used as a direct input to the calculation of the iMOS.

The reader is kindly reminded about Section 2.1.1.3, where the one-way-delay (OWD) was described as a network traffic metric composed of four elements: the processing delay, the transmission delay, the propagation delay, and the queueing delay. While the majority of the following related works are focused on measuring the end-to-end OWD, the mechanisms developed in this work focus on methods to obtain accurate delay measurements between an end point and an intermediate node.

The complexity of the methods found in the literature depends on the accuracy of the delay measurement. Different levels of accuracy are required for different type of applications. These applications range from OWD-based Service Level Agreements (SLAs) [47] to mechanisms designed to monitor and minimise the effects of the OWD on the performance of real-time applications, such as VoIP.

There is a tremendous amount of work dedicated to measuring the OWD, and most are comprised in a few notable surveys, such as [47] and [48]. Wang et al. [47] mention that the OWD can be used for SLA validation between network service providers and customers or between neighbouring network service providers. Another usage of the OWD can be, according to the same authors, learning about the underlying properties or characteristics of the current networks, such as network topology, traffic patterns, and protocol distribution. The OWD measurements can be fed into measurement-based simulation systems to guide the capacity planning, application tuning or performance lifting.

Both surveys, [47] and [48], agree that the RTT is an inaccurate estimation of the OWD. As already discussed, the OWD is obtained from halving the RTT measurements. The key argument showing the RTT estimation shortcoming is path asymmetry. In IP networks different paths may exist for the forward and reverse direction between two nodes, hence the performance of these paths is different, however this can not be captured by the RTT measurement.

As typically the OWD is measured by using probes, Wang et al. [47] argue that these probes compete for network resources with packets transporting important payloads. A direct consequence of that is the potential of creating disturbances in applications using these payloads.

It is possible to use a synchronisation-based or a non-synchronisation-based OWD measurement method. The former uses timestamping to compute the OWD as the difference between the sending and the receiving times, and the latter uses statistical models which are able to estimate with a certain error the OWD.

De Vito et al. [48] focus their survey on the the synchronisation-based OWD methods. As the method's name suggests, the nodes between which the OWD is measured have to be time-synchronised. This means that the internal clocks of the two nodes have to be perfectly synchronised to obtain accurate OWD measurements. Hence, the accuracy of the OWD measurements is directly influenced by the accuracy of the time-synchronisation methods.

De Vito et al. mention that there are three predominant time-synchronisation solution widely used:

- **NTP [49]**: Is a protocol which uses probes to synchronise a client's clock to a precise time-keeping server. It does not require dedicated hardware, however it may take up to a few hours until the desired level of accuracy is obtained. A typical accuracy level obtained using NTP is 500 microseconds.
- **Global Positioning System (GPS)**: This solution uses the widely known GPS system by attaching GPS receivers to all nodes which are to be synchronised. The accuracy is reached in a much shorter period than using NTP, however at the extra cost of hardware support. A typical accuracy level obtained using GPS is in the range of tens of microseconds.
- **IEEE 1588 Standard [50]**: Is a high precision, hardware supported method of assuring time synchronisation in measurement and control applications, but also in telecommunications networks. The networking and computing resources needed by this protocol are minimal, since it uses dedicated hardware. The highest accuracy for this protocol is in the range of 100 nanoseconds with extra hardware support.

In Section 4.2 the nodes between which the OWD needs to be measured are time-synchronised. Specifically, these nodes are the femtocell and the femtocell gateway. Section 4.2 will show that the LTE standard imposes the time synchronisation on these nodes, and in practice this is done using NTP.

In Section 4.3 a network topology is used where synchronizing nodes is proven to be difficult, and as a consequence a non-synchronisation-based OWD measurement solution, called the Delay Piggy-Backing Mechanism (DPBM), is developed to alleviate this shortcoming. A few other works are present in the literature which use non-synchronisation-based methods to obtain the OWD. Gurewitz et al. [51] introduced a novel approach for the estimation of OWDs. It was based on conducting multiple one-way measurements among pairs of nodes, and optimizing the value of a global objective function that is determined by the overall network topology instead of just by individuals.

The authors gave a detailed procedure for estimating the deterministic components in OWDs. The variable OWD estimation is based on measuring and analysing the link between neighbouring nodes. Their work is limited in the sense that from the components of the OWD, only the transmission and the propagation delays are modelled.

In a work conducted by Wei-Xuan and Shun-Zheng [52], a model for both the constant and the stochastic parts of the OWD is proposed. The constant parts of the OWD are the transmission and the propagation delays, and the stochastic part is composed of the queueing delay. They use an active method by injecting probes into the network, and employ an iterative Fourier-to-time reconstruction algorithm on the measured intra-gaps of the end-to-end OWD, to determine the distribution of the queueing delay. The complexity of their method is high and is performed on an end-to-end basis.

Ngamwongwattana and Thompson [53] performed another work on obtaining an estimation of the OWD without having the nodes time-synchronised, however their work does not use probes, and instead use the RTP packets received in a VoIP call. Their method is comprised of two phases, namely Sense and Sync. With this method a good estimation of the OWD, however without including an estimation of the propagation delay. Their work has two major limitation for which it would not be suitable as a replacement for the DPBM mechanism proposed in this work. Specifically, their method is strictly end-to-end and the OWD estimations obtained during the first seconds of a call are not reliable.

The non-synchronisation-based OWD estimation works presented above are not suitable as a replacement for the DPBM as these methods either lack in simplicity, or are end-to-end only, or use probes which may contend with the actual VoIP traffic. Section 4.3 will show that the DPBM is very lightweight and uses actual RTP packets to piggy-back delay information instead of using active probes, in order to obtain the necessary information to compute the OWD. No other works have tackled to OWD measurements in the same way it is addressed in this work.

3.2 Edge-to-middle VoIP Call Quality Monitoring Solutions

This subsection describes other works done in the field of call quality monitoring. This is related to Chapter 4 of this Thesis, where the call quality is measured by means of the E-Model [31], with the important difference that in this work the E-Model is used in an intermediary node, rather than on end-nodes as it is typically done.

3.2.1 VoIP Backbone Monitoring

Prior to any important VoIP deployment, a few studies were carried out in the research community to assess the performance of VoIP support in Internet backbones. It is worth noting that some of the studies are dating from more than a decade ago when the typical speed of an Internet connection was in the range of few hundreds of kilobits per second. Thus, actual VoIP deployments did not really exist and as a consequence VoIP traffic had to be injected artificially in the investigated network. Later, when the Internet was widely adopted across the Globe, a few studies were focused on investigating data obtained from real VoIP conversations carried on a provider's backbone network.

Markopoulou et al. [54] have investigated over a long period of time the network delay and packet loss suffered by VoIP calls injected in the backbone networks of seven different providers connecting five US cities. The monitoring was done in an end-to-end fashion, storing data for offline call quality assessment using the E-Model [31]. A few key factors were discovered after analysing the resulting data. First, Markopoulou et al. discovered that even overprovisioned networks can cause important degradation on VoIP calls. In a second observation, the paper noted that many of the networks need to provide the means of prioritizing VoIP traffic over regular data traffic. In another observation, the paper concluded that the network delay was accountable for most of the call quality degradation. That only highlights that the networks of those days were not ready to support VoIP yet as the delay was considerably large due to slow electronics in the networking equipment. However, in time the networks became faster and the influence of the delay was much reduced, so that nowadays packet loss is the main reason for call quality degradation.

Another end-to-end quality assessment of injected VoIP calls is presented by Jiang and Schulzrinne [55]. Their assessment approach is different than a typical VoIP call quality measurement. Instead, the authors have defined a new metric, the *end-to-end availability*. The assessment involved the usage of 14 nodes spread across the Globe. VoIP calls were initiated between these nodes and data was collected over a very long period of time. Although some of the VoIP calls were routed through research networks, the *end-to-end availability* was not significantly better than the quality of those routed through commercial networks. Their results showed that a significant percent of unavailability was caused by long packet loss outages, while the effects of short bursts of packet loss can be compensated by PLC-enabled voice codecs.

As the Internet has evolved and many users have turned their attention to the cheaper telephony solution, some Internet websites provided the functionality of VoIP call quality testing. By running such a website for a very long period, Sylor et al. [56] have collected monitoring data related to calls originated at the end-users and terminated at one ore more of the seven servers deployed across the Globe. Among a full characterisation of the VoIP call quality analysed at a global scale, two conclusions are generally applicable: first, it was concluded that not only jitter should be used in SLAs, but also loss caused by late packet discards, but this metric is more difficult to measure as it is codec dependent. A second interesting conclusion was that the G.711 performed much better than G.729 which was supposed to perform similarly to G.711 with much less bandwidth usage.

Once VoIP reached enough maturity to be a generally accepted Internet application, the need to monitor the data attached to it has driven the research community to investigate different monitoring architectures. Such a work was conducted by Kobayashi and Ishibashi [57]. They proposed a new architecture which was built on some of the existing building blocks related to monitoring, such as access control list (ACL)-based metering functions adaptable to existing commercial routers and switches. One of the main challenges in VoIP monitoring is to provide the correlation between the signalling protocol used by the VoIP application, such as SIP [12] or H.232 [10], and the actual audio stream. Compared to the previous presented works where the monitoring was done in and end-to-end fashion, Kobayashi and Ishibashi envisioned the architecture to be end-point agnostic and applicable to multiple types of network deployments.

All works presented above were based on results obtained by injecting VoIP calls in a backbone network, hence no user behaviour, such as call dropping due to low quality, was really taken into account. A better analysis can be obtained by monitoring real VoIP traffic.

Such analysis is presented by Birke et al. in [58] and [59]. In their study, the authors have obtained data from one important ISP in Italy which provides their data, VoIP and IP television services over a fully IP architecture. Birke et al. used a packet sniffer in one of the backbone nodes of the ISP to collect only the packet headers of VoIP packets needed in the calculation of the equivalent VoIP call quality. The estimation of the network delay used in their work was obtained after post-processing of the data. Their results showed important statistical data regarding the distribution of the call duration and a comparison between DSL users and FTTH users. However, when compared to the solution presented in this thesis, their offline post-processing technique can not be used for taking real-time Call Admission Control (CAC) decisions.

3.2.2 VoIP Monitoring Related to iMOS

Here, a few studies on passive monitoring VoIP at intermediate nodes are presented. Passive monitoring refers to the investigation of existent VoIP traffic, as opposed investigations using injected VoIP traffic. The first framework proposal of monitoring VoIP not only at the end-nodes, but also were possible at network gateways was presented by Clark et al. [32]. The study focused on obtaining the simplified E-Model, by assigning default values to some of the E-Model parameters, as those parameters were related to calls placed using a PSTN network. This study was the first to present the short version of the E-Model formula. Further, monitoring VoIP not only at the end-nodes, but also were possible at network gateways was proposed by Clark et al. but their work focused more on investigating each remaining parameter of the E-Model, such as I_d and I_e.

One important factor Clark et al. considered is the *recency* effect presented by France Telecom [60]. The major contribution of France Telecom's study was the finding of two time values related to human reception of variation in call quality. Specifically, on average, a human subject can detect a variation from good to worse call quality in 5 seconds and the reverse in 15 seconds. The value of 5 seconds was considered in Chapter 5 when proposing the period of the cycle in which the Intermediate Mean Opinion Score (iMOS)-based CAC gathers data for examination. Another important finding of France Telecom's study was the observation that human subjects have a limited audio memory, thus when asked to rate an audio recording or a call the decision is taken mainly based on the last 30 seconds of the audio stream.

Another passive VoIP monitoring framework was described by Conway [61]. Instead of using network metrics such as delay, packet loss ratio and jitter, Conway proposed the usage of PESQ. PESQ is a technique used to rate an audio stream by comparing the original stream with the stream obtained at the reception point. PESQ was presented in Section 2.1.2. The content of the sniffed VoIP packets could not be compared with the original for obvious reasons. However, the headers of the packets were saved and during a later investigation, the payload was replaced with the content of the known audio source. In this way real VoIP traffic could be post-processed using PESQ. However this method could not be used by real-time CAC mechanisms.

The works presented in this subsection are related to passive VoIP monitoring solutions on which this Thesis is also focused. However, there are substantial differences between the reviewed works and this thesis:

- some algorithms are not suited for real-time monitoring because the processing of the ongoing VoIP calls involves techniques which demand high processing power;
- none of the works actually presents a viable solution of measuring the one-way-delay; although the RTT can be used instead, the two flows composing a VoIP call are not always symmetric in terms of network delay, and this can lead to erroneous decisions when these are taken fully or partially based on the network delay.

3.3 VoIP Call Admission Control

Call Admission Control (CAC), in the VoIP context, is essentially a process through which a network entity determines whether or not to accept a new call request. There is tremendous work done in the area of defining different types of criteria which can be employed by the CAC process. This section aims at categorising these criteria, highlighting their strengths, weaknesses and limitations, and finally fitting this work into the big picture of VoIP CAC.

A few key surveys in the area exist [62–67] and their key findings include reasons why CAC exists and why it is necessary, and how can CAC be categorized.

Even though a network is designed to meet a given performance for a nominal traffic load, the actual traffic surpasses significantly the amount initially considered. Overloaded networks perform poorly in delivering acceptable QoS to all traffic types, especially to VoIP calls. In the first phase of the overload, the packet delay increases significantly and if no corrective actions are taken, packet loss will occur with detrimental consequences on VoIP calls, as shown in Section 2.1.2. One obvious solution is to design a mechanism able to timely detect network overload events, temporarily stop the acceptance of new traffic in the network, and eventually employ a solution to clear the backlog of packets in the networks' queues.

A particular issue arises in networks transporting VoIP, specifically the addition of only a single call over the network's capacity tears down the call quality of all ongoing calls in the network. This issue has been reported in many works in the area [63, 67–73].

An alternative to CAC is resource over-provisioning, which means provisioning of network capacity beyond the traffic load expected, so that applications behave as if the network is unloaded. Over-provisioning represents a simple initial solution, yet it does not scale with greater deployment of VoIP services, is highly expensive in terms of wasted network resources, and it does not handle well failure scenarios [65, 67, 68].

All the issues summarised above

- traffic surpassing the load expectation,

- the call quality drops when one single call is added over the network's capacity,
- and the unsuitability of the over-provisioning due high cost and low scalability

support the need of CAC, with special emphasis on the VoIP use-case.

Figure 3.1 shows a taxonomy of CAC mechanisms designed for networks transporting VoIP. As the number of works in the CAC for VoIP area is very high, the taxonomy presented below is mainly focused on CAC mechanisms specifically tailored for issues around wireless access networks.

Control-messages exchange takes places prior to the establishment of a new VoIP call, to inform the involved parties about the imminence of a new call. CAC typically is set to accept or reject the new requests based on different criteria, with the ultimate goal to maintain specified levels of call quality for all calls in the network. The first division of CAC solutions specifies where the admission decision is taken. Two categories have been identified, specifically in the end-nodes, or in one or more intermediate nodes.

It was observed that all CAC mechanisms for wireless access have in common the mapping of network performance parameters into call quality metrics. Some of the network performance parameters are obtained directly through measurements of metrics directly affecting the call quality (i.e. delay, packet loss, and jitter), or mathematically modelled from metrics specific to the wireless access medium to obtain an estimate of a metric directly affecting the call quality. These solutions can be either active or passive, based on whether they use probes or actual VoIP traffic to determine the network performance.

The active measurement-based solution involves the usage of probe-packets, which are injected in the network to obtain network performance metrics such as loss [74–76], delay[1] [71], and throughput [77]. An end-node initiates the probing upon receiving a call request and the result of the probe monitoring will enable the CAC to take a favourable decision if the performance is higher

[1] The delay is obtained by halving the RTT.

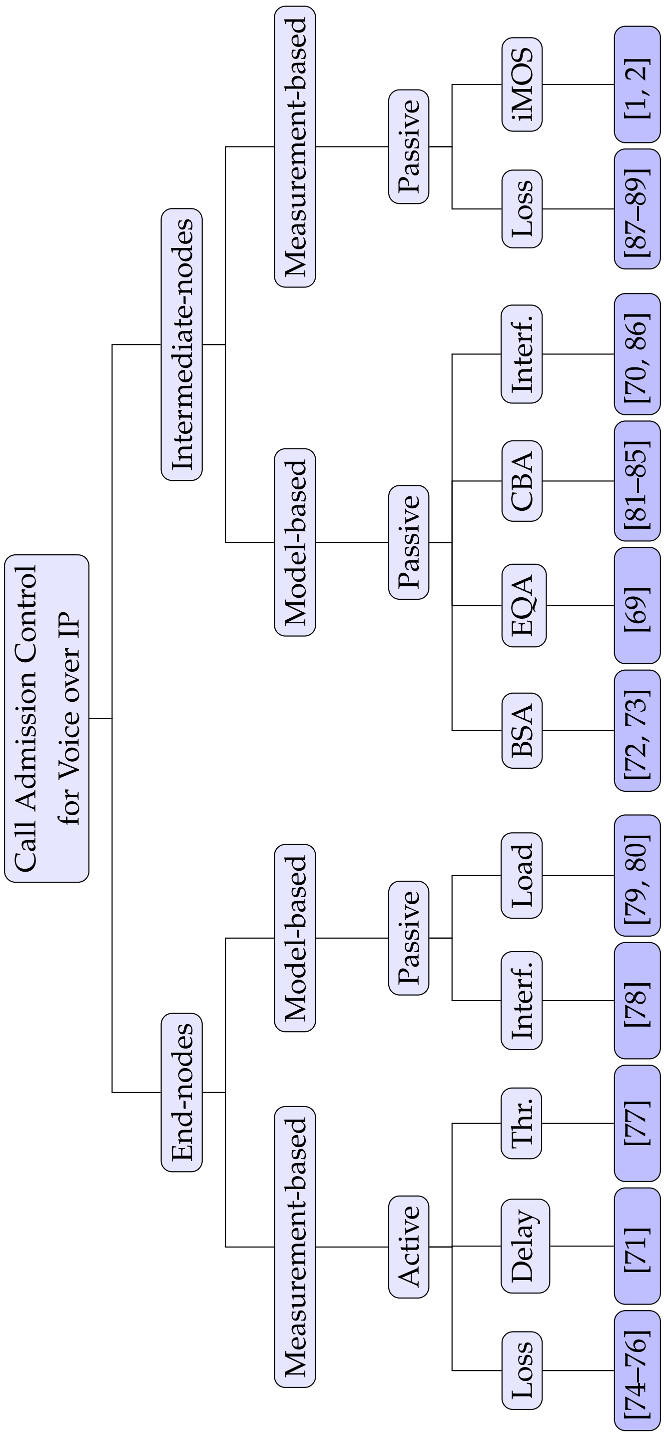

Figure 3.1: Call Admission Control Taxonomy

than a pre-established threshold, or a negative decision otherwise. This solution is easy to implement and scalable, however the injection of *dummy* traffic in the network unnecessarily increases the network load, sometimes with detrimental effects on real traffic. Another negative effect of this solution is increasing of call setup time.

The model-based solutions are typically passive and have been proposed in order to address the shortcomings of the active measurements. Specifically the interference [78] and load [79, 80] of the wireless channel are monitored continuously, and the end-node learns to map the current status of the network to call quality. This enables the end-node to increase its accuracy on predicting the call quality, and reject future call requests when the estimate falls under a pre-established threshold.

The CAC decision can be also made by an intermediate entity. This entity intercepts the exchange of control messages, and a CAC decision is made based on a predictive model or on network performance measurements. The CAC solutions in intermediate nodes are typically passive, as the high complexity and low scalability of establishing probing messages between the intermediate node and the end-nodes would raise even more issues. The CAC solutions at intermediate nodes are preferred to the end-to-end solutions, as they give a network operator or a VoIP service provider the ability to better provision for user satisfaction across all users of the network, not only for a single user as it is the case for end-node CAC solutions.

The model-based CAC solutions in intermediate nodes are mainly studied in Wireless Local Area Networks (WLANs), where the CAC mechanism is placed in the access point, and takes decisions regarding calls originated or destined to roaming clients. These solutions focus on estimating the call quality from the access point's buffer utilization ration (BSA) [72, 73], or from the wireless channel busyness (CBA) measured in the access point [69], or from the theoretical network capacity (EQA) [69], or from the interference level around the access point [70, 86]. All these are estimates, and need to be implemented in the access point where computational resources are scarce, and ultimately are inaccurate in measuring the call quality.

An alternative to model-based CAC solutions, is performing performance measurements on VoIP traffic flowing through intermediate nodes. The works in [87–89] have focused on measuring and mapping the packet loss to call quality. New calls are accepted if the current measured packet loss is lower than an admission threshold. As shown in Section 2.1.2, the call quality depends on multiple factors besides the packets loss, namely the voice codec chosen, the delay, and jitter. The work proposed in this Thesis, fills the gap, by providing a measurement-based solution to monitor the actual MOS at intermediate nodes. The solutions proposed here are termed iMOS-based CAC solutions, and are presented in Chapter 5.

3.4 Summary

This Chapter gave an overview on the works done in the field of measuring the one-way-delay (OWD), VoIP call quality monitoring, and VoIP Call Admission Control (CAC) solutions. The OWD is used to assess the network performance in general, but also as a reflection of VoIP call quality in particular. A large variety of solutions exist in the literature on how to measure the OWD and these have been discussed in terms of complexity, cost, and differences to the solution proposed in this work. The VoIP call quality monitoring architectures proposed in the literature have been reviewed in the middle part of this Chapter, with emphasis on the similarities with the iMOS monitoring solution proposed in this work. Finally, a taxonomy of the CAC solutions proposed in the literature was presented in the last part of the Chapter, with emphasis on the gap-filling contribution represented by this work in the area of VoIP CAC.

CHAPTER

FOUR

IMOS — ESTIMATING VOIP CALL QUALITY AT INTERMEDIATE POINTS

4.1 The iMOS Concept

Section 2.1.2 presented the MOS concept and Section 2.1.2.1 showed how the E-Model [31] is used to compute the MOS by using metrics typical to packet-switched networks, such as delay, packet loss, and jitter. The usage of the E-Model implies that the MOS will be obtained at the end-nodes. As such, the estimated quality of the VoIP conversation is known only by the caller and the callee.

The iMOS is the MOS estimation obtained at an intermediate point which is located in the path of the packet flow[s] of a VoIP conversation. The iMOS thus implies the existence of a node which intercepts VoIP packets from which it extracts the RTP headers needed in the calculation of packet loss. By using VoIP packets belonging to the same VoIP conversation, the intermediate node calculates the jitter for that packet flow. However, obtaining the delay is not as simple as on the end-nodes. On the end-nodes, the reports from the RTCP packets are used to compute an estimate of the network delay. This is commonly known as the RTT estimation and was presented in Section 2.1.1.2. However, the RTCP reports can not be used in the intermediate node, due to their dependency to local information stored on the end-nodes, such as the starting time of the RTT measurement procedure. It is also known that the accuracy of the delay estimation based on RTT measurements is prone to errors [90].

Figure 4.1 shows the influence of the delay estimation error on the MOS. The MOS versus one-way-delay is plotted with lines, and the areas around the lines represent the error domain of the MOS, when the one-way-delay is measured with an accuracy level of 10%, 25%, and 50%. While at 10% the estimation error may be difficult to perceive by a human, its detrimental influence on MOS-based mechanisms leads to employing incorrect actions. An example of a MOS-based mechanism is a CAC mechanism which determines the admission or rejection of VoIP calls based on MOS measurements. Such CAC mechanisms tailored for the specifics of femtocell and WMNs are presented in Chapter 5.

There are many applications of the iMOS, which in other words can be considered as an accurate mid-point call quality metric. Below, the most important are highlighted:

- Isolating call-quality related problems by dividing the network in $n+1$ parts, if there are n iMOS monitors. In Section 4.2 it is shown that issues can arise in the infrastructure provided by a third party and the MOS measurements performed in the user's device are not sufficient to determine the network part causing those issues.
- Enabling the VoIP service provider to develop iMOS-based Customer Experience Manager (CEM) or Service Quality Manager (SQM) applications, possibly in connection to Service Level Agreement (SLA) compliance and negotiation.

The next section will describe how the iMOS is calculated using accurate delay measurements obtained in DSL-backhauled femtocell networks, and Section 4.3 will describe how the delay can be estimated using the proposed DPBM.

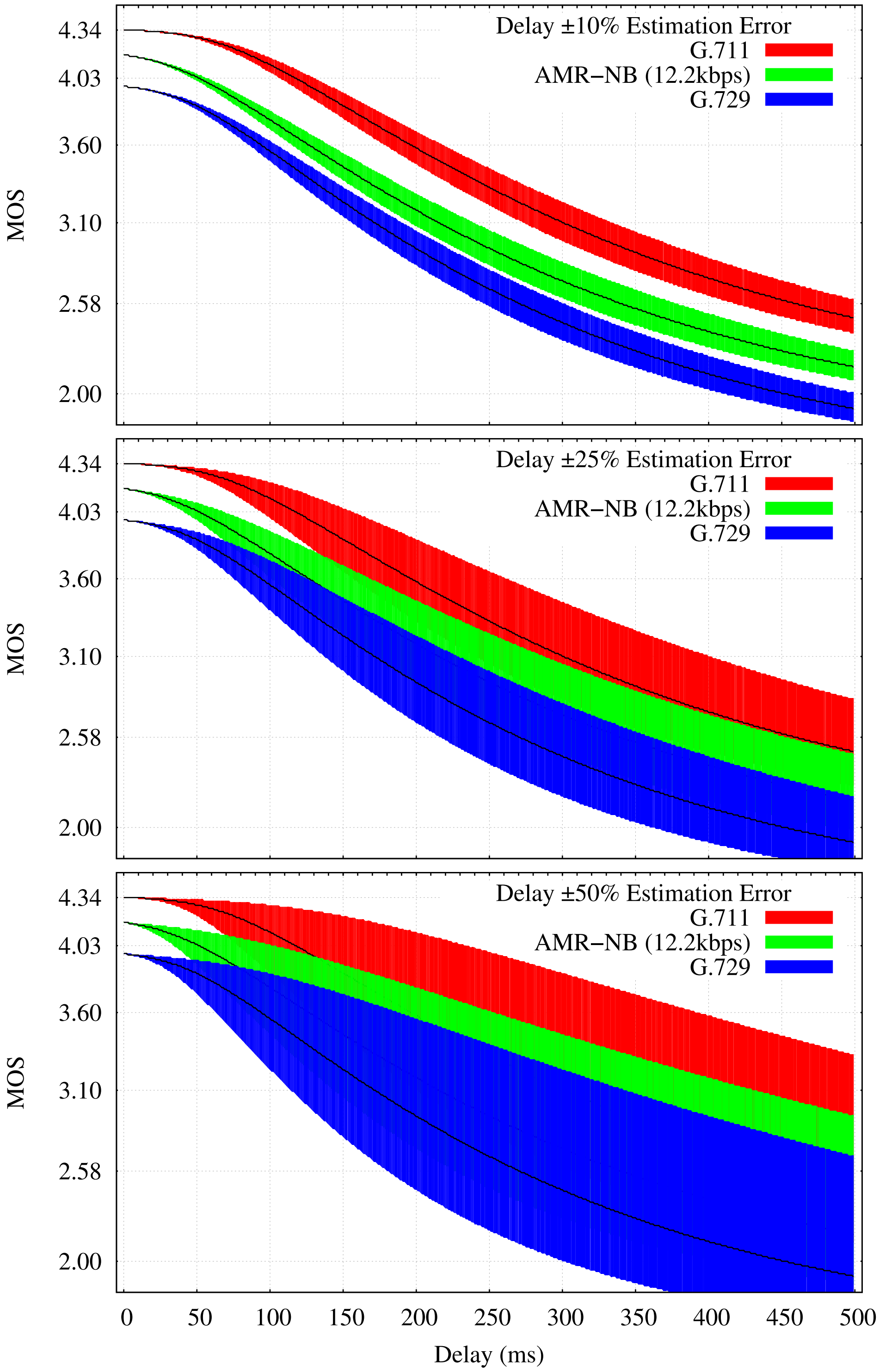

Figure 4.1: MOS vs. Estimated RTT Delay

4.2 Measuring the iMOS in DSL-backhauled Femtocell Deployments

4.2.1 Description

4.2.1.1 Targeted Femtocell Deployment Scenario

The use of small cells deployed in residential premises is often referred to as femtocell technology. This offers a way to improve cellular coverage for indoor users and also increases network capacity with minimal infrastructure costs. Such femtocells have been deployed for 3G and LTE cellular networks with future deployments expected to be based on Long Term Evolution Advanced (LTE-A) standards.

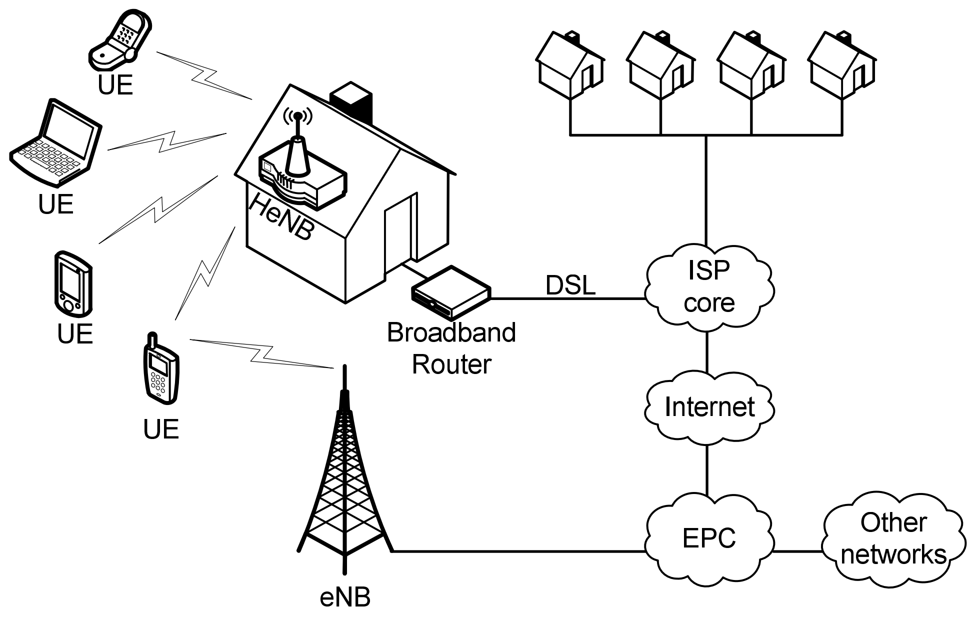

Figure 4.2: Typical deployment for LTE Femtocells with DSL backhaul

Femtocells typically use the subscriber's broadband connection to carry the traffic to and from the cellular network [91]. In 3GPP's Release 8 [34], femtocells are denoted with Home evolved NodeB (HeNB). The terminology and full LTE architecture was presented in Section 2.2. Figure 4.2 depicts a typical

LTE femtocell deployment with the femtocell backhaul provided over DSL. It can be seen that a femtocell can provide service to any device equipped with a cellular interface (e.g. cellphone, laptop, PDA). All these types of equipment are generically called User Equipment (UE).

The femtocell's backhaul connectivity is provided over a residential Internet connection, which in this work is assumed to be a DSL connection. The DSL broadband router is shared with other UEs of the same residence. The traffic originated in the UEs is in direct contention for Internet bandwidth with the other femtocell traffic and may also contend with other types of traffic in the access network. The ISP's core network and the Internet backbone network are traversed by VoIP packets which finally reach the Evolved Packet Core (EPC).

A call placed over a femtocell is, from a backhaul perspective, identical to a VoIP call. As the migration towards femtocell deployments will coincide with an usage increase of high-bandwidth multimedia applications, the broadband access network will become a major bottleneck.

From a data traffic point of view, VoIP is regarded as a stream of relatively small packets that are time sensitive. Network delay, jitter, and packet loss rate are taken into account whilst measuring the overall or instantaneous estimative quality of a conversation. Although MOS is used to calculate the quality of a conversation on a end-to-end basis, this work proposes this metric to be calculated also at intermediate network nodes. In this way, the degradation of the call between the transmitting UE and the intermediate node can be measured.

As mentioned in Section 4.1, the packet delay needed for the MOS calculation can be estimated using the RTT. Another method is based on the difference between local time and the time obtained from packets' timestamps, in this case the local time of the two nodes must be synchronised. This synchronisation requirement is met by femtocells, which must meet rigorous regulations in order to comply with the LTE air-interface standard [34, 92]. Another component of the LTE architecture which meets these synchronisation requirements is the HeNBGW [34]. Based on these findings, the delays can be easily calculated using RTP [19] timestamps correlated with UTC timestamps contained in corresponding RTCP packets [19]. The details presented above, enable the HeNBGW to measure the MOS between any femtocell which is attached to the EPC via that particular HeNBGW.

The synchronisation requirements imposed by the LTE standard have driven the femtocell operators to employ time synchronisation techniques. Real femtocell traffic was investigated in this work, and the finding is that at least one operator uses NTP [49]. One important observation is that the NTP packets are exchanged very frequently, mainly to compensate for frequency shifts in the femtocell's local oscillator. These frequent exchanges do not impose serious problems on a wired backhaul, however on a wireless-based backhaul, as presented in Section 4.3, the overhead is disruptive for real-time applications. The number of these messages can be up to 240 packets per minute as shown in [93].

Figure 4.3 depicts the HeNB (femtocell), the HeNBGW, and the other components linking the caller to the callee. The HeNB is connected to the ISP's DSL core network via a DSLAM. Further up is the Internet backbone which serves as a conveyor for the VoIP traffic towards the EPC. The entry point in the EPC is the HeNBGW, where the iMOS is computed for the VoIP traffic flowing away from the femtocell; that is the femtocell uplink VoIP traffic, if the femtocell is used as a reference point. The VoIP traffic traverses the EPC and is forwarded to the other end-node of the call.

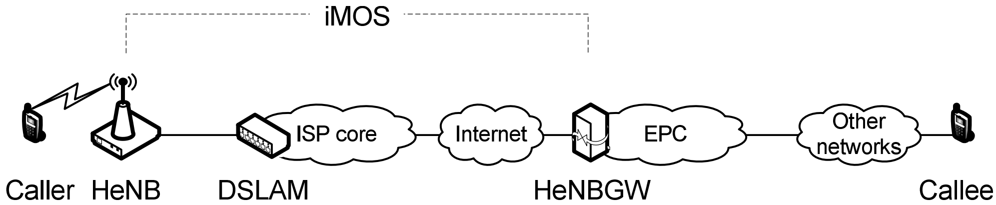

Figure 4.3: The concept of iMOS in DSL-backhauled femtocell deployments

Multiple topology combinations can be derived by analysing the path between the PDN Gateway (P-GW) and the callee, as depicted in Figure 4.4:

- The first is a HeNB-to-HeNB call where both HeNBs belong to the same Service Provider. It can be observed that the femtocell backhaul is symmetrical, both the caller and the callee using a DSL connection for their femtocells. In this case the HeNBGW will compute iMOS values for four

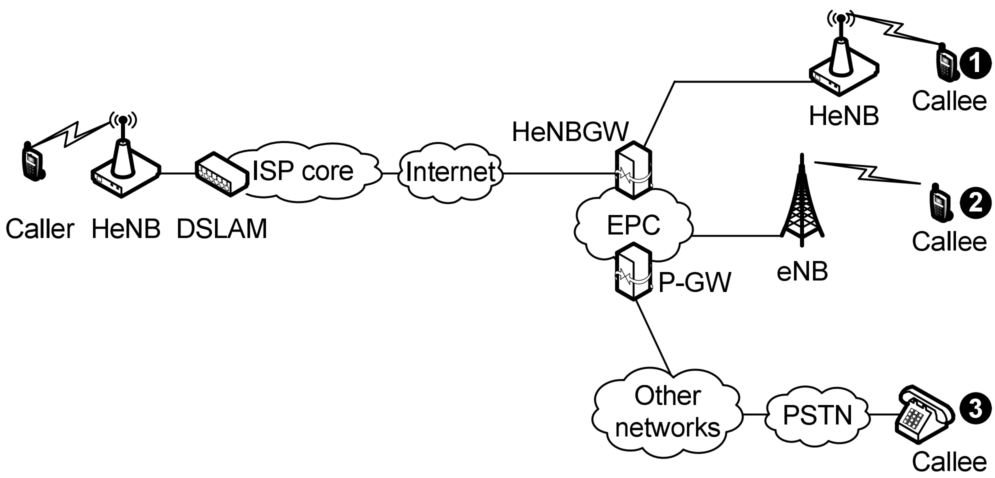

Figure 4.4: LTE Femtocell deployment, with multiple possibilities to connect the caller to the callee

VoIP packet streams: two characterizing the link between itself and the *caller* and two for the link between itself and the callee.

- The second scenario involves a callee in the same Provider's network, attached to a base-station (eNB).
- The third scenario depicts a situation where the callee uses the fixed telephone network to participate in the call.

For consistency with the presented DSL-backhauled femtocell context, the focus in this work is mainly on scenario 1 in Figure 4.4.

4.2.1.2 The Recency Effect

MOS is a metric which estimates the subjective ranking given by a human subject to a voice conversation. User's MOS report is obtained after the conversation is over. In [94] it is shown that humans are able to recall only the last 30 seconds of the conversation. That further means that the MOS reported by a human subject is mainly based on the call quality experienced towards the end of the conversation.

In order to enhance the accuracy of the iMOS measurements, this work proposes equation (4.1) which applies an Exponentially Weighted Average (EWA) on the history of iMOS samples, giving higher weights to the previous 30 seconds of the conversation. The reader is reminded that the E-Model computes a MOS value for every received VoIP packet, as mentioned in Section 2.1.2.1. The iMOS values are stored as $(timestamp, iMOS)$ pairs. The first arrived VoIP packet is assigned with 0 as timestamp and the last VoIP packet is assigned t_f. In other words, t_f represents the time elapsed since the arrival of the first VoIP packet of the monitored conversation. This work proposes a 1 to 100 scale for the weighting function (4.1). t is the iterator through the timestamp values. The exponential function is chosen as it maps better to the pattern describing the behaviour of the human audio memory [95]. Specifically, the human audio memory limitation of 30 seconds determines a subject to base their call quality assessment only on the last 30 seconds of the conversation. This effect is captured by equation (4.1) proposed by the author of this thesis. $H(x)$ is the Heavyside function, detailed in equation (4.2).

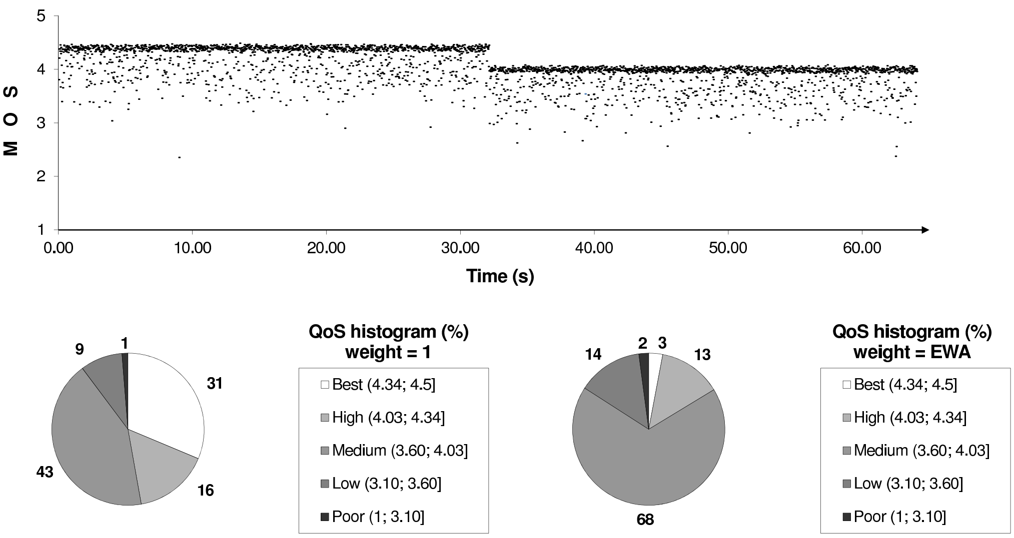

Figure 4.5: Weighted MOS score mapped into corresponding call quality categories

$$w(t) = H(30 - t_f) \cdot \left(e^{4.6 \cdot \left(1 - \frac{t_f - t}{30}\right)} \right) + \\ + H(t_f - 30) \cdot \left(e^{4.6 \cdot \left(1 - H(t_f - 30 - t) - H(t - (t_f - 30)) \cdot \frac{t_f - t}{30}\right)} \right) \tag{4.1}$$

$$H(x) = \begin{cases} 0 & if\ x < 0, else \\ 1 & for\ x \geq 0 \end{cases} \tag{4.2}$$

The numerical example depicted in Figure 4.5, helps the reader to better grasp the effects the human recency effect on the estimative call quality assessment methodology. This example is particularly chosen so that the MOS values cover all 5 call quality categories (Table 2.3). Pie-like plots are used to show the histogram of the MOS values as a percentage of the total captured samples. It can be observed that there is a quality degradation suffered at the conversation midpoint. The histogram on the left represents the default case in which the human recency effect is not considered. The histogram on the right captures the effect of the human recency effect on the perceived call quality. The difference between the two histograms highlights that estimative computational models can lead to erroneous results, unless all human factors are weighted into the mathematical formulae.

4.2.2 Simulation Parameters

Further, the impact of multimedia streaming on the quality of femtocell VoIP calls, is assessed. In particular, the congestion at the Digital Subscriber Line Access Multiplexer (DSLAM) where femtocell calls contend for access with other traffic, is examined. In the investigated scenario (Figure 4.6) it is assumed that the femtocell service provider has no control over the DSLAM or over any other equipment belonging to the ISP. It is also assumed that the DSLAM is a non-QoS network equipment, which means that services such as *diffserv* are not enabled on the DSLAM.

The scenario depicted in Figure 4.6 is simulated using Qualnet 4.5 [96], which is an open source network simulator. The source code required a small modification: by default Qualnet implements a half-duplex VoIP call; that was

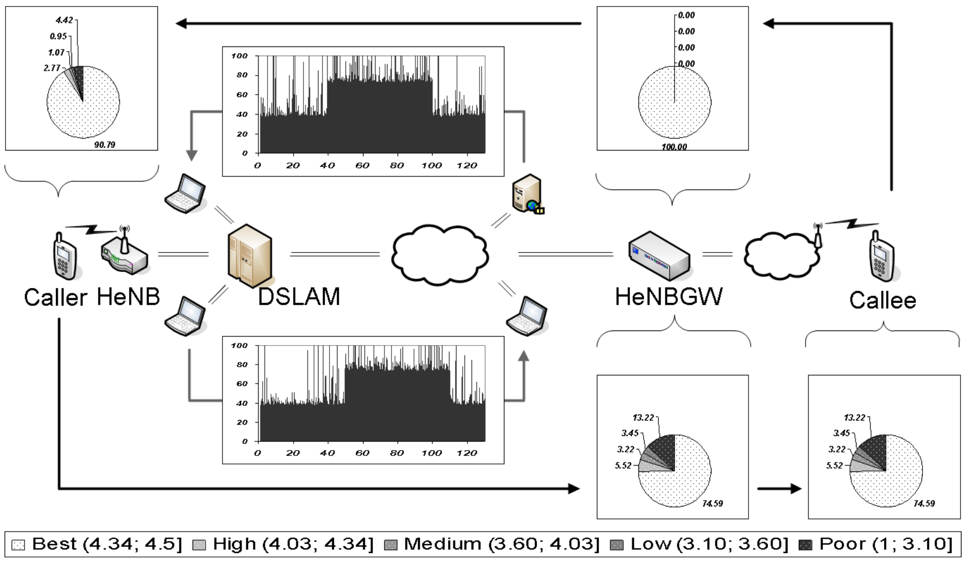

Figure 4.6: Simulation Scenario

modified to support full-duplex VoIP calls. The voice codec used is G.711 for reasons of simplicity and ease of analysis, as G.711 is a CBR voice codec. The results are expected to be broadly similar for any VBR voice codec, such as AMR or G.729.

In Figure 4.6 a VoIP call is initiated between the caller and the callee. This is scenario 1 from the the diagram showing the multiple call connectivity in Figure 4.4. The DSLAM forwards both femtocell and background traffic. The background traffic is used to emulate traffic congestion at the DSLAM and is simulated with the help of VBR sources. The DSLAM forwards the femtocell and the background traffic to another network entity, named MAN, which denotes the combined networks of the ISP and Internet Backbone. The HeNBGW forwards the VoIP call further towards the callee. The iMOS is calculated for both VoIP packet flows, since the call is full-duplex and the callee uses femtocell services from the same provider.

A burst of multimedia background traffic load is used to cause congestion at the DSLAM. The burst starts at the 40^{th} second and ends at the 100^{th} second of the simulation, on caller's downlink direction. On the uplink direction a similar background burst pattern is delayed by 10 seconds.

4.2.3 Results Analysis

The simulation results are shown on the same diagram showing the simulation topology in Figure 4.6. The histograms show the call quality assessment after the duplex VoIP call ended.

On the caller's downlink, it can be seen that the 100% of the iMOSs obtained at the HeNBGW, for packets generated by the callee, fall in the *Best* quality category. However, the histogram obtained from the MOSs measured in the caller's UE show that the call quality has dropped, lowering the percentage of *Best* call quality to roughly 90%. Therefore the link between the HeNB and its HeNBGW must have caused the quality degradation. The reason for this degradation is the multimedia background traffic which created congestion at the DSLAM. Since no priority mechanism is implemented in the DSLAM, the congestion is causing impairments on the voice quality.

A similar situation occurs on the uplink VoIP traffic. The difference in this case is the burst for the background traffic occurs 10 seconds later than on the downlink direction. These 10 seconds of congestion would be perceived negatively by a human subject, which is influenced by the recency effect. As such the formula proposed in equation (4.1) captures the effect of the burst and reflects it in the call quality histogram. A much higher degradation is reported by the callee's UE because of the congestion caused by the recent burst and the human recency effect.

The results show that the proposed scheme can locate the cause of VoIP call quality degradation. Such information may then allow remedial actions to be taken.

4.2.4 Summary

The work in Section 4.2 shows how the iMOS concept described in Section 4.1 can be used in a LTE femtocell deployment using DSL as backhaul for the femtocells. The time synchronisation requirements imposed by the LTE standard is the key factor enabling the HeNBGW to measure the iMOS.

Another contribution of this section is the development of a formula which includes the human recency effect in the VoIP call quality estimation. More weight is given to the quality estimation samples acquired towards the end of the conversation. The same methodology can be used also in real time, simultaneous with the VoIP call, by employing the formula as an Exponentially Weighted Moving Average (EWMA) instead of an EWA. The results of using the formula as a EWMA would be similar to the results obtained by asking the human subject to grade the conversation at any point in time of the live conversation.

The iMOS can be used to drive CEM applications and may be used by the HeNBGW to make admission control decisions at the HeNB. Section 5.2 shows how the iMOS-based admission control is done in a similar LTE femtocell deployment scenario.

4.3 Measuring the iMOS in VoIP over WMN Deployments

This section focuses on another wireless access technology, specifically on 802.11-based WMNs. The description starts by pointing out the necessity of computing the iMOS for VoIP calls backhauled over WMNs. Then details are given on how the iMOS is actually obtained in a network topology which does not benefit from the time synchronisation requirement, as it is the case for femtocell deployments. A collateral benefit of obtaining the iMOS is highlighted by proposing a novel mechanisms called the DPBM. The DPBM is a mechanism able to prioritise VoIP packets based on their cumulative delay suffered in the WMN. NS-3 simulations have been used to validate both the iMOS concept and the proposed DPBM in a VoIP-over-WMN scenario.

4.3.1 Description

Section 2.3 provided the necessary background on the architecture and standards used in WMNs. The following WMN scenario gives more motivation around deploying WMNs with emphasis on VoIP as the use case and the issues arising when provisioning call quality for VoIP calls.

4.3.1.1 Targeted WMN Scenario

WMNs represent a solution allowing telecom- or WiFi-operators to provide large areas with wireless data connectivity at relatively low cost. Thanks to minimal cabling required, these types of network are of particular benefit for transient deployments in which the network must be rapidly and cheaply deployed. Such a solution is capable of providing coverage in areas for which it is not feasible or economically viable to deploy more traditional wireless coverage. In cases where these networks are used to support voice communication it is essential that the capacity is maximised while providing high levels of user QoS.

WMNs primarily rely on IEEE 802.11 technology, more commonly called WiFi. This choice is motivated by the low cost of hardware and the unlicensed spectrum in which it operates. However, IEEE 802.11 was not initially designed for real-time multimedia applications such as VoIP; this problem is further exacerbated in a WMN environment due to the multi-hop nature of such networks.

A large amount of work can be found in the literature proposing various solutions to improve the voice call quality and number of supported VoIP calls over a WMN. The majority of these solutions focus on packet aggregation, which works by combining multiple small VoIP packets into a single frame before transmission, hence lowering the number of frame transmissions required, which further reduces the probability of collision. The reader is encouraged to visit Section 2.3.3.2 for more details on the 802.11-specific frame aggregation feature.

As explained in Section 2.1.2, VoIP call quality is affected by a number of factors. These include the voice codec, the absolute end-to-end delay (mouth-to-ear delay), packet delay variation (jitter), and packet loss. Both packet loss and jitter are easily obtained by examining the RTP header information over which VoIP calls are transported. However, the absolute delay is hard to determine, firstly because the RTP header information does not contain absolute delay information and secondly because the WMN nodes are not time synchronised. The reader was presented in Section 4.2 with the details about how the packet delay can be obtained when the nodes are synchronised, and how this delay feeds into the calculation of the iMOS.

As already emphasized at the beginning of this Chapter, intermediate nodes are not able to avail of the RTT measurement to calculate the iMOS. For WMN deployments, this issue is addressed by another contribution of this work, which is the development of a novel method of obtaining the absolute delay of VoIP packets traversing a WMN.

The reference WMN topology used throughout this Section is depicted in Figure 4.7. The focus of this work is not on the client-to-AP link, but rather on the relay-to-relay links which interconnect the mesh nodes. These links carry the traffic between APs and the network's gateway. This is done in a multi-hop fashion between relays; for this reason a significant delay may build-up and loss may occur due to bottlenecks and congestion across the WMN.

The price for wireless hardware has dropped over the last years, making it feasible to install multiple wireless interfaces in a single mesh node. In

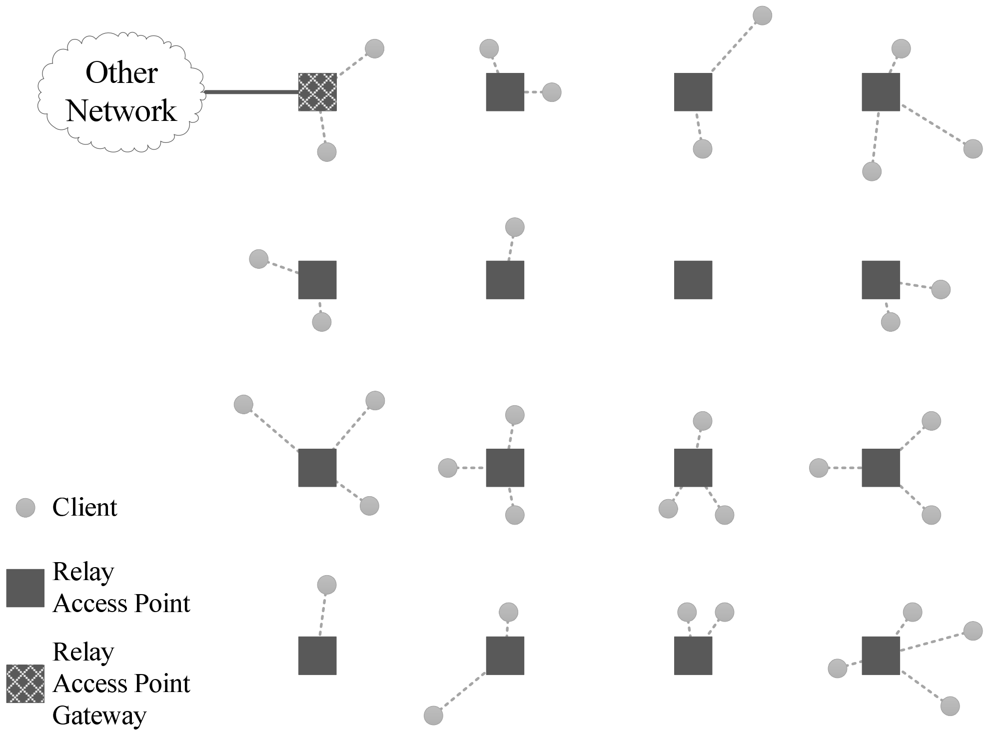

Figure 4.7: Targeted WMN Deployment Scenario

the case of a WMN utilising 802.11 equipment, it is possible to use multiple radio interfaces to increase the system's capacity by allowing multiple wireless communication channels to be used simultaneously between relays.

4.3.1.2 VoIP Capacity

VoIP capacity is the maximum number of calls a network can support with guaranteed call quality. The call quality threshold is typically specified as a MOS value. The VoIP capacity analysis requires the iMOS reported by the WMN's gateway as an input. Since two packet streams are required to transport the VoIP packets between the caller and callee, the iMOS and the derived VoIP capacity is measured on each direction. Considering the caller is a user of the WMN, then the VoIP traffic flowing away from the caller is the uplink traffic and the VoIP traffic flowing towards the caller is the downlink traffic.

Some aspects are omitted in the literature regarding how the actual VoIP capacity is measured. The majority of previous works consider only the mean voice call quality over all calls in the network, as the threshold for VoIP capacity. This produces inaccurate call capacity measurements as in this case some calls may have very poor call quality while others have very high call quality. This makes it extremely difficult to state what the actual capacity is. In this work, besides the mean of the call quality, the standard deviation of the mean call quality is also considered in order to improve the accuracy of the VoIP capacity measurement within the WMN.

In order to highlight the differences between different statistical metrics which can be used in calculating the overall iMOS, we provide in Table 4.1 a numerical example of hypothetical iMOS measurements. The rows in Table 4.1, show iMOS samples for the downlink (DL) and the uplink (UL), for M calls transported over the WMN. The overall mean quality per call, per call direction, is obtained by computing the mean of the samples ($\mu(iMOS)$). The variations of the quality per call, per call direction, are captured by the standard deviation of the samples ($\sigma(iMOS)$). The overall mean quality of all calls supported by the WMN is obtained by computing the mean of the means ($\mu(\mu(iMOS))$), also known as the grand mean. The variation of the mean quality among all calls is captured by the standard deviation of the means ($\sigma(\mu(iMOS))$).

Table 4.1: Obtaining the Mean (μ) and Standard Deviation (σ) of iMOS

	$iMOS$		$\mu(iMOS)$		$\sigma(iMOS)$	
Call	**DL**	**UL**	**DL**	**UL**	**DL**	**UL**
$call_1$	3.90, 3.90, ..., 3.80	3.73, 3.81, ..., 3.69	3.88	3.78	0.4	0.6
$call_2$	3.82, 3.80, ..., 3.80	3.70, 3.73, ..., 3.75	3.81	3.71	0.2	0.3
...	...	...	...	...	...	...
$call_M$	3.85, 3.80, ..., 3.75	3.75, 3.70, ..., 3.65	3.80	3.70	0.5	0.5
		$\mu(\mu(iMOS))$	3.74	3.84		
		$\sigma(\mu(iMOS))$	0.04	0.06		
		$\mu(\sigma(iMOS))$			0.33	0.53

It is important to consider the mean of the standard deviations ($\mu(\sigma(iMOS))$) as this captures the overall variation of all calls' quality. This value is higher than the standard deviation of the means ($\sigma(\mu(iMOS))$). The numerical example supports this fact which is demonstrated by the following arguments: a few users may already experience call quality which falls below the iMOS threshold, before the moment when the mean of the overall call quality ($\mu(\mu(iMOS))$) has reached the threshold; hence the minimum between the standard deviation of the means ($\sigma(\mu)$) and the mean of the standard deviations ($\mu(\sigma)$) of the call quality should be considered as the iMOS threshold of the VoIP capacity. It is worth noting that other works do not consider these metrics and primarily focus only on the mean call quality values ($\mu(\mu(iMOS))$). This work addresses these shortcomings.

Further, the VoIP capacity is obtained by monitoring the overall mean quality of all ongoing calls, while the number of calls is linearly increased, usually by one call at a time. The network's VoIP capacity is reached when the measured call quality falls under the specified iMOS threshold. A common procedure in the literature [97–99] is to consider the threshold around 3.6 on the MOS scale, or the equivalent of 70 on the R scale (see Section 2.1.2.1).

4.3.2 The Delay Piggy-Backing Mechanism (DPBM)

This subsection provides a detailed overview of the proposed DPBM with a particular focus on the targeted WMN scenario. Another key element of the proposed mechanism is the algorithm which governs how packets are enqueued at each hop and hence this is also described.

As a packet traverses a network, the main contributor to the overall packet delay is usually the waiting time in the outgoing queues. This delay is directly related to the queue size, data rate and the MAC mechanism employed. The delay can be kept to a minimal value in the case of wired networks such as ethernet-based networks. However, in the case of 802.11-based wireless networks, the waiting time is influenced by the load and congestion of the wireless medium.

802.11 utilises a CSMA/CA-based DCF for medium access control. DCF uses a listen-before-transmitting approach with an exponential random back-off timer. This back-off introduces a medium access delay which can increase significantly as more nodes attempt transmission hence creating congestion over the medium. These waiting times are directly reflected in the absolute packet delay, and are further exacerbated in the case of 802.11-based WMNs due to the use of multiple hops from the user to the gateway.

A QoS-enabled WMN is assumed in this work, in order to maximise voice call quality. As such, each mesh node has four different outgoing queues on each of its interfaces. Each of the four queues is destined for one of the four ACs defined by the 802.11e protocol [46]. The ACs are: AC_BK for *Background* traffic, AC_BE for *Best Effort* traffic, AC_VI for *Video* traffic, and AC_VO for *Voice* traffic.

The DPBM proposed in this work measures the cumulative queuing delay for each VoIP packet which passes through the AC_VO queues of the WMN nodes encountered. Just before the scheduler dequeues a VoIP packet for transmission, the total amount of waiting time of that VoIP packet in that WMN node, is attached to the packet itself in a *delay field*. The value of the *delay field* is initialised to zero by the first WMN relay which received the VoIP packet from a WMN client. At each relay the queuing delay is added to the total delay in the *delay field* such that, at each relay, the *delay field* contains the total delay experienced by the VoIP packet since entering the WMN. Each relay uses the *delay field* to insert the VoIP packet at a specific location in the outgoing queue. The packets are enqueued in such a way, that old packets are placed closer to the head of the queue. In this work, Push-In-First-Out (PIFO) queues are used for the AC_VO queue in order to leverage the potential of the DPBM.

In a standard First-In-First-Out (FIFO) queue, packets are always enqueued at the last position, if space allows, whereas in a PIFO queue, packets are enqueued at a position calculated based on a comparison criteria. In the case of the DPBM being proposed here, the criteria for comparison is the delay value found in the *delay field*. Algorithm 4.1 shows how the PIFO queue works in the DPBM implementation. The actual NS-3 code used to implement a VoIP PIFO queue in this work, is presented in Appendix A.3.

Algorithm 4.1: Pushing a VoIP packet into the PIFO queue

input: *cp* (current VoIP packet), *Q* (destination queue)

```
if Q is empty then
    enqueue cp at head of queue Q;
    RETURN;
else
    for p ← Q.begin() to Q.end() do
        if TS_WMN(cp) > TS_WMN(p) then
            if length(Q) == MAX_SIZE then
                drop Q.end();
            push cp in Q;
            RETURN;
        else
            p++; // increment queue index
    if length(Q) < MAX_SIZE then
        enqueue cp, the youngest packet, at Q.end();
    else
        drop cp, the youngest packet;
    RETURN;
```

The algorithm receives two parameters as input, the current VoIP packet (*cp*) that has to be pushed-in, and the VoIP queue (Q) where *cp* is to be placed. Packets already belonging to Q (if any) are denoted with *p*. If Q is empty, *cp* is placed at the first position; else, the insertion position is determined by iterative comparisons between the delay value of *cp* and the ones of the packets belonging to Q via the TS_{WMN} (Time Spent in the Wireless Mesh Network) function.

The TS_{WMN} function returns the content of the *delay field* when called for *cp*. It returns the content of the *delay field* added to the time interval during which p waited in Q from its enqueueing moment until now, when called for p. When *cp's* delay is bigger than the delay of a p, then *cp* is *pushed-in* right before that p, but not before *Q's* current length is verified. If Q's current length is equal to its maximum size, then the last packet of Q, which is also the youngest packet in Q, is dropped. If this *for* loop reaches the end of Q, then, if Q's current length is less than its maximum size, then *cp* is added to Q. Otherwise, *cp* is dropped as Q is full and all its packets are older than *cp*.

PIFO queues require more computational power during the *push-in* phase than normal FIFO queues. The authors of [100] present a possible implementation of PIFO queues. Their performance evaluation showed that current hardware is able to cope with the increased processing power imposed by PIFO queues. However, PIFO queues were not attractive in the past as the extra processing power needed would render the equipment financially prohibitive.

As previously mentioned, the global cumulative packet delay is a piece of information contained in the packet itself in the *delay field*. Two options are available for placing the cumulative delay value into a VoIP packet. First by modifying an existing packet header to accommodate a new field, or second, creating a new proprietary packet header. The first option is more desirable, as there are many packet headers which support experimenter fields, allowing to piggy-back some information over the network using existing packet headers.

For VoIP, an end-to-end delay of more than 150ms has negative effects on the voice call quality, whereas an end-to-end delay of more than 400ms significantly degrades the conversation interactivity [31]. Considering these values, a *delay field* that can store delay values up to 256ms is well suited; hence, an eight-bit field is required.

Table 4.2: Simulation Setup

Parameter	Value
Simulator	NS-3.10 [101]
Topology	Grid 4x4
Distance between nodes	125 m
Number of interfaces	2 for AP-to-AP
	1 for AP-to-MN
WiFi Mode	802.11a for AP-to-AP
	802.11g for AP-to-MN
WiFi Data Rate	6 Mbps
Access Method	CSMA-CA
Propagation Model	LogDistancePropagationLossModel
	Path Loss Exponent: 3
	Reference Distance: 1 (m)
	Reference Loss: 46.667 (dB)
Error Rate Model	YansErrorRateModel
Remote Station Manager	ConstantRateWifiManager
WiFi interfaces queue size	50 packets
Routing Algorithm	Fixed routes, pre-discovered by OLSR
Aggregation Type	Hop-by-hop
Call Type	Full-duplex
Call Duration	120 seconds
Voice codec	G.729a+VAD
	20 bytes @ 20 milliseconds
Call Quality Assessment Method	E-Model [31]
	Ie = 11
	Bpl = 19
	A = 0
Speech Model	ITU-T/P.59 [6]
VoIP capacity threshold	R=70 (MOS=3.6)
Number of simulation seeds	5

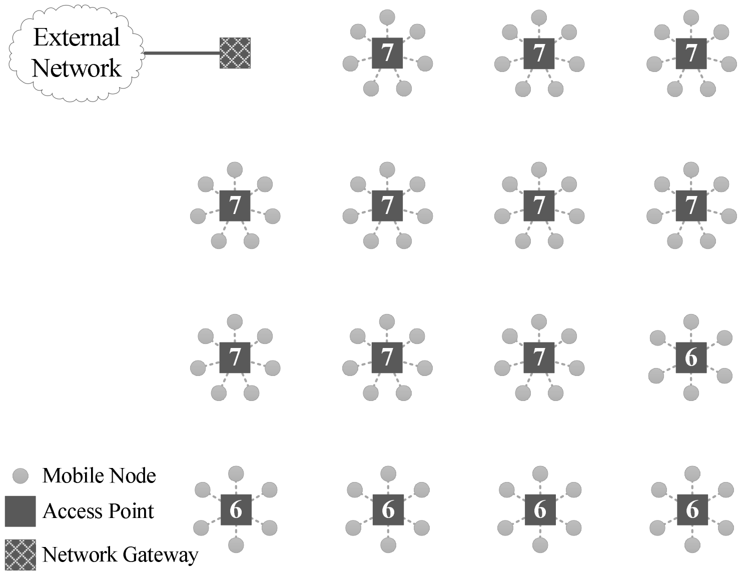

Figure 4.8: Simulation Topology and Call Placement

4.3.3 Simulation Parameters

This section describes the simulation scenario and all relevant parameters used to validate the proposed mechanism and produce the results presented in Section 4.3.4. Table 4.2 provides all pertinent values used to parameterise the simulations.

The network topology is a 4 by 4 grid with the WMN gateway located at one corner (Figure 4.8). The inter-node distance is 125 meters and the physical link data rate is set to 6Mbps between relays. The relatively large inter-node distance ensures that relays can not communicate with other relays located diagonally.

Each relay is equipped with three interfaces: two for relay-to-relay communication and one interface for AP-to-user communication. The relay-to-relay interfaces are configured such that one interface supports the uplink traffic while the other the downlink traffic.

The default queue size in NS3 is 400 packets per Access Category (AC), however this is not a realistic queue size. In this work a queue size of 50 packets per AC was used. This represents the current default queue size used by the most widespread wireless drivers, *MadWiFi* [102] and *ath5k* [103].

The routing table entries are statically assigned. Any WMN routing protocol would introduce statistical artefacts, hence hindering the drawing of statistically significant conclusions from the results. However, it is worth noting that Optimized Link State Routing Protocol (OLSR) was used for initial route discovery but was prevented from making route modifications after VoIP calls started to be injected.

One full-duplex VoIP call is established by each user, with a call length of 120 seconds which is in accordance with reports showing data about the average phone call duration [58]. Each call is established between a caller and a callee which is located outside of the WMN. No VoIP calls are placed on the gateway as these VoIP packets would be directly forwarded to the wired interface's packet queue and would not be transported over the WMN, and therefore may bias the results.

There are four possible mechanism combinations which are used for evaluating the effects of the DPBM and the Aggregation Mechanism (AM). These are:

- *A: DPBM Disabled & AM Disabled*;
- *B: DPBM Enabled & AM Disabled*;
- *C: DPBM Disabled & AM Enabled*;
- *D: DPBM Enabled & AM Enabled*;

The simulations are carried out in batches, one simulation batch per number of injected calls. The number of injected calls is incremented from 1 to 100 in steps of one, resulting in a total of 100 simulation batches. In each batch, each of the four mechanism combinations mentioned above is simulated five times, each time with a different simulation seed. The total number of simulations performed in this work is:

$$100\ injected\ calls * 4\ mechanism\ combinations * 5\ simulation\ seeds = 2000\ simulations$$

During each simulation run, there is a ramp-up period during which calls are randomly initiated with different starting times to avoid traffic source synchronisation issues. However, the region of interest is when all calls are simultaneously active. Each call is active for 120 seconds and is denoted as *Call Duration* in Table 4.2.

Figure 4.8 shows the call distribution when 100 calls run concurrently. This distribution is one instance of the approximative 312×10^{15} possible combinations of distributing 100 calls over 15 nodes[1]. Testing all possibilities is not feasible, given the very large amount of time and processing resources necessary. Hence, we chose to use a uniform distribution, which would not bias the results towards the best or the worst case scenario.

4.3.4 Results Analysis

4.3.4.1 Comparison methodology

From the four cases of mechanism combinations between the DPBM and AM described above, A is the case where neither of the two mechanisms is enabled. This reference case, which shows the influence of a plain 802.11e-based WMN on VoIP calls, is used as the basis for comparison for the other three cases. The comparison is done for the overall mean of the call quality ($\mu(\mu(iMOS))$), VoIP capacity, and for the variation of the call quality represented with a grey area around the mean. Specifically, the variations are: $\mu(\mu(iMOS)) \pm \mu(\sigma(iMOS))$ for the mean of the standard deviations, and $\mu(\mu(iMOS)) \pm \sigma(\mu(iMOS))$ for the standard deviation of the means, as explained in section 4.3.1.2 with the help of Table 4.1.

Figure 4.9 is used as the reference plot for the following explanations. To determine which overall mean call quality ($\mu(\mu(iMOS))$ depicted with m in the plot) is higher, the size of the area between the curve representing the $\mu(\mu(iMOS))$ and the x axis are compared. This comparison is done for cases $B,\ C,\ and\ D$ against case A. The results of the comparisons are depicted in the plots as the percentage of improvement under the label m. The size of the areas around the mean are used to determine which variation is smaller. The results of the variation comparisons are depicted under the label *v1* which represents

[1]The formula used to calculate the number of combinations possible for 100 similar calls to be distributed over 15 nodes is $C_{n-1}^{c+n-1} = \frac{(c+n-1)!}{(n-1)!c!}$, where c=100 calls and n=15 nodes.

the $\mu(\mu(iMOS)) \pm \mu(\sigma(iMOS))$ and $v2$ which represents the $\mu(\mu(iMOS)) \pm \sigma(\mu(iMOS))$ as the percentage of improvement compared to case A.

The call quality analysis for the uplink (Figure 4.9) and downlink (Figure 4.10) is presented in separate plots, as there is an obvious difference between the two cases, as described below.

In addition to presenting the results for the call quality on the uplink and downlink, the performance analysis of the other three metrics is presented. Specifically for the delay, packet loss, and jitter (Figures 4.12, 4.13, 4.14, 4.15, 4.16, 4.17), using the same comparison procedure. It is worth noting that the percentages depicted in the plots represent relative improvements to case A.

4.3.4.2 VoIP Call Quality and VoIP Capacity Evaluation

Figure 4.9 depicts the results obtained for the call quality and VoIP capacity on the uplink direction. This shows that the call quality, variations of call quality, and VoIP capacity increased for cases C and D with respect to case A, however there was no significant increase between themselves. This means that, in the uplink direction, the DPBM had very little impact. This is due to the traffic pattern on the uplink, where all VoIP packets are forwarded towards a single node–the network gateway. There are few packets available for aggregation on the Access Points (APs) further away from the gateway, hence only a small improvement can be observed.

However, in the downlink direction as shown in Figure 4.10, the situation changes. The aggregation mechanism takes advantage of the fact that VoIP packets are always available for aggregation at the gateway. It can be seen that, between cases C and D, the mean call quality remains relatively similar, however the variations around the mean are smaller when the DPBM is used to enhance the effects of the aggregation mechanism, by about 7% for both $v1$ and $v2$. A smaller variation in case D indicates that the DPBM achieves a more fair distribution of call quality perceived by the users across the network. The other important effect of our proposed DPBM is the increase in VoIP capacity. The VoIP capacity is determined at the intersection of the m, lower part of $v1$ are and lower part of $v2$ are with the VoIP capacity threshold, i.e. MOS=3.6. It can be seen in Figure 4.10 that case D shows an improvement of 3 extra calls, or 6% capacity improvement, over case C.

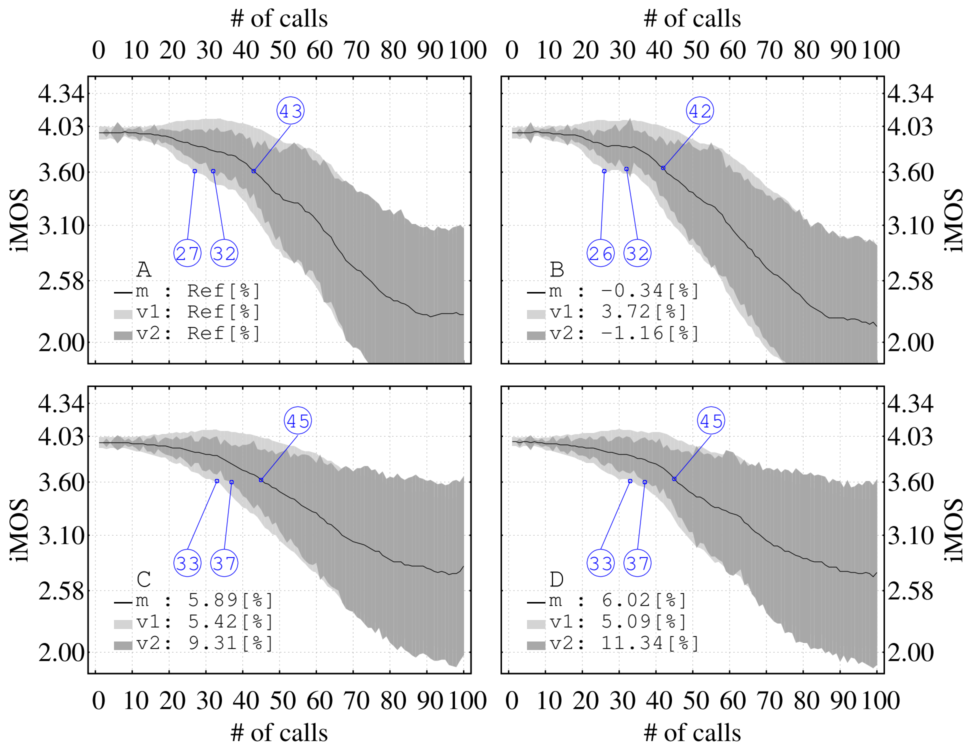

Figure 4.9: MOS versus number of injected calls on the uplink. The intersection of the threshold line (iMOS=3.6) and the curves represent the VoIP capacity measurements. *Legend: m represents the* $\mu(\mu(iMOS))$, *v1 represents* $\mu(\mu(iMOS)) \pm \mu(\sigma(iMOS))$ *and v2 represents the* $\mu(\mu(iMOS)) \pm \sigma(\mu(iMOS))$. *The circled values on the plot are VoIP capacity measurements.*

It can be seen that different ways of measuring the VoIP capacity can lead to different results, as shown in Figures 4.9 and 4.10. When intersecting the quality threshold, i.e. MOS=3.6, with the *m* curve and the lover parts of the *v1* and *v2* areas, the resulting VoIP capacity has different values, and the lowest value is more accurate. This value corresponds to the $\mu(\mu(iMOS) - \mu(\sigma(iMOS))$ curve, and should be considered when performing VoIP capacity analysis in future works in the field.

4.3.4.3 Delay Measurement Accuracy Evaluation

Figure 4.11 shows the accuracy of the delay measurements obtained at the WMN's gateway for the uplink and downlink, for cases B and D where the

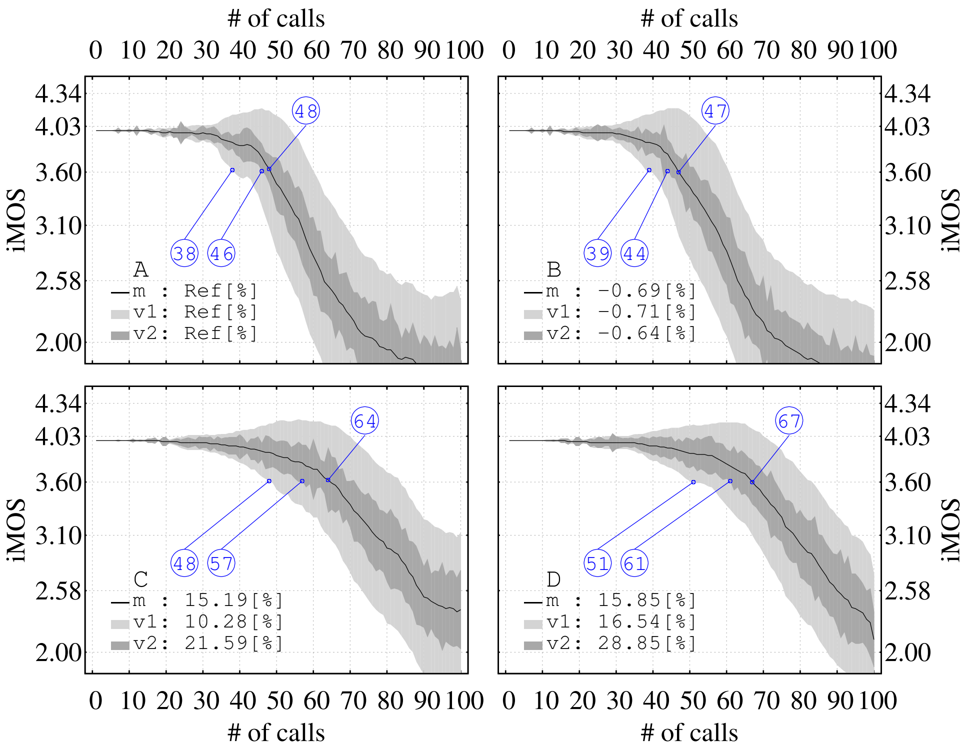

Figure 4.10: MOS versus number of injected calls on the downlink. The intersection of the threshold line (iMOS=3.6) and the curves represent the VoIP capacity measurements. *Legend: m represents the* $\mu(\mu(iMOS))$, *v1 represents* $\mu(\mu(iMOS)) \pm \mu(\sigma(iMOS))$ *and v2 represents the* $\mu(\mu(iMOS)) \pm \sigma(\mu(iMOS))$. *The circled values on the plot are VoIP capacity measurements.*

DPBM is enabled. The accuracy is measured as the difference between the grand mean of the actual delay and the grand mean of the delay obtained from the delay field. This is enabled because the simulator keeps all nodes synchronized, thus the actual delay can be measured as timestamp difference. However, in a real WMN deployment this measurement can not be obtained as the nodes are not time synchronised. Even if the nodes are time synchronised, that would impose a large overhead, as it could be seen in Section 4.2.1.1 and [93].

It can be seen that the delay difference is negative for a total number of injected calls lower than 50. The DPBM adds at least 1 millisecond to the delay field, even when a packet does not wait at all in the AC_VO queue. As a result, the value in the delay field is equal to the number of hops in low

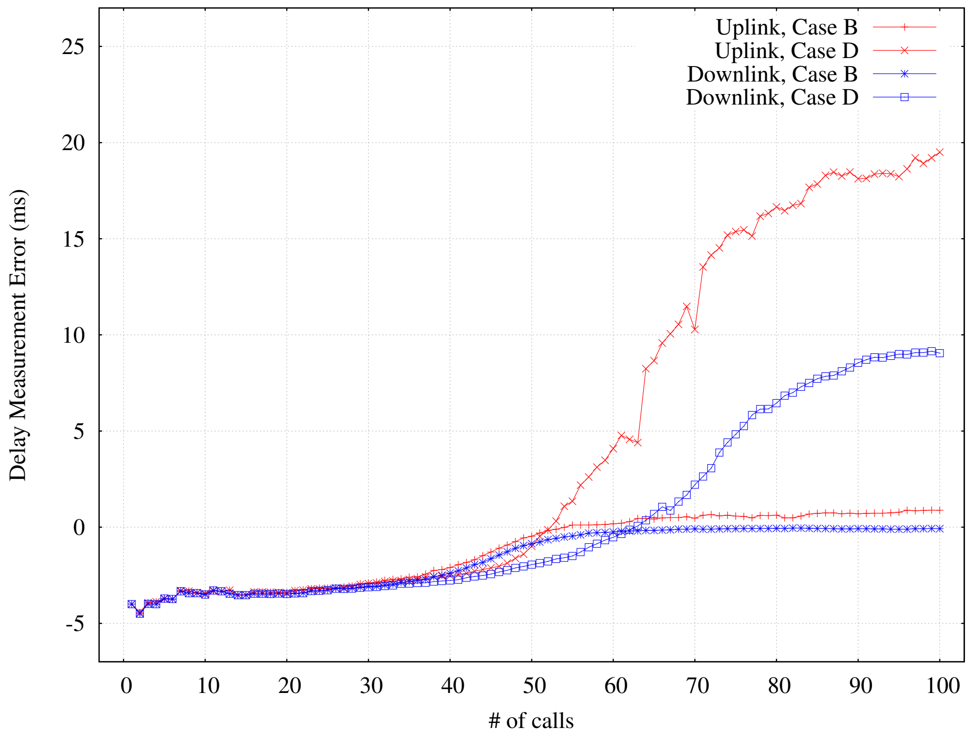

Figure 4.11: Accuracy of the piggy-backed delay value. *The points in the plot represent the difference between actual delay and the piggy-backed delay*

network load conditions. In the same low load conditions, the actual packet delay shows values close to 0. This explains the negative values when the total number of injected calls is lower than 50.

The delay difference for case B where only the DPBM is enabled when the network load is around capacity, on both uplink and downlink, is very low which means that the proposed DPBM is able to provide accurate delay measurements for the iMOS calculation process.

In case D, where both the DPBM and aggregation mechanism are enabled, the delay difference is positive, with values below 20 milliseconds, with lower differences on the downlink than on the uplink. In other words, the actual delay is higher than the delay retrieved from the delay field. The difference represents the time needed to transfer or to retransmit the aggregated frame. This time is not captured by the proposed DPBM, however an extension to the DPBM to include these time is proposed as future work.

It is noteworthy that as the network approaches its VoIP capacity the delay field contains very accurate delay measurements, thus assuring the iMOS measurement is also accurate.

4.3.4.4 Delay, Packet Loss and Jitter Evaluation

The performance of the delay, packet loss, and jitter, on the uplink and downlink, are depicted in Figures 4.12 and 4.13, Figures 4.14 and 4.15, and Figures 4.16 and 4.17, respectively. It can be observed that on the uplink, the results show a similar pattern to the call quality analysis, which showed that the DPBM provides little help to the aggregation mechanism due to limited availability of VoIP packets.

However, on the downlink path of these cases, the results show an overall improvement of case *D* of about 7% in terms of metric variation around the mean. Another aspect worth noting is the smaller slope of the *m* curves for the delay, packet loss, and jitter, for case *D*. This means that the DPBM lowered the negative effects of these metrics on the perceived call quality.

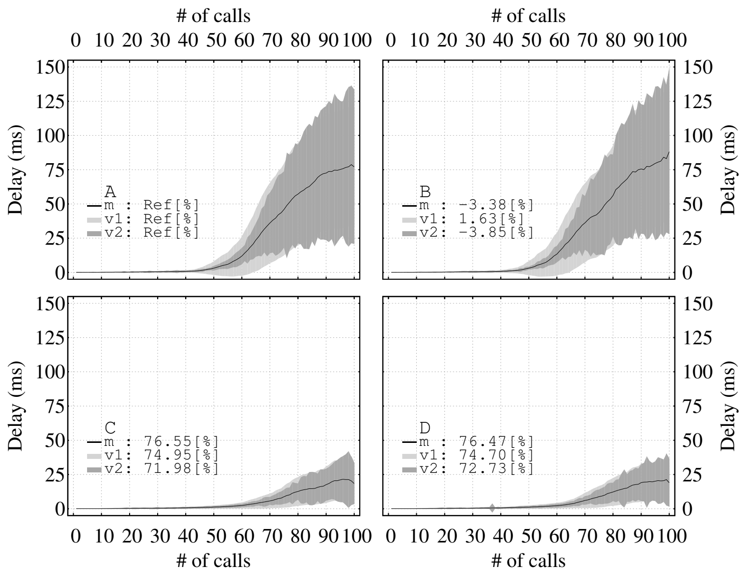

Figure 4.12: Delay versus number of injected calls on the uplink. *Legend: m represents the $\mu(\mu(Delay))$, v1 represents $\mu(\mu(Delay)) \pm \mu(\sigma(Delay))$ and v2 represents the $\mu(\mu(Delay)) \pm \sigma(\mu(Delay))$.*

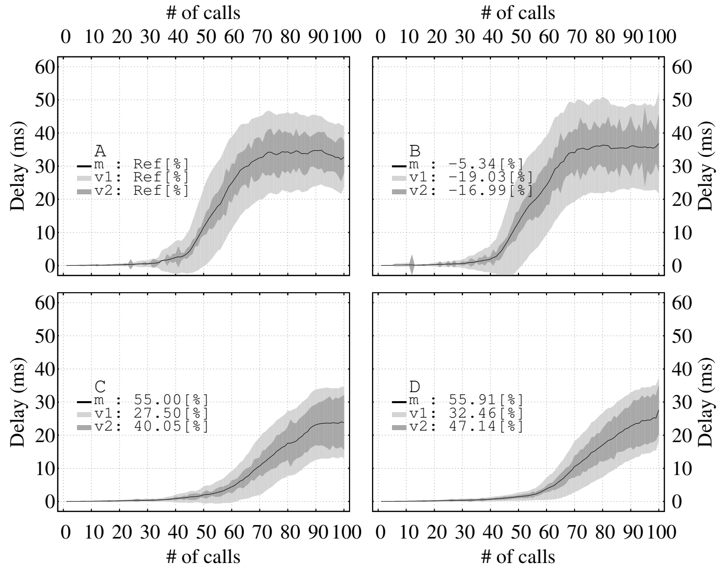

Figure 4.13: Delay versus number of injected calls on the downlink. *Legend: m represents the $\mu(\mu(Delay))$, v1 represents $\mu(\mu(Delay)) \pm \mu(\sigma(Delay))$ and v2 represents the $\mu(\mu(Delay)) \pm \sigma(\mu(Delay))$.*

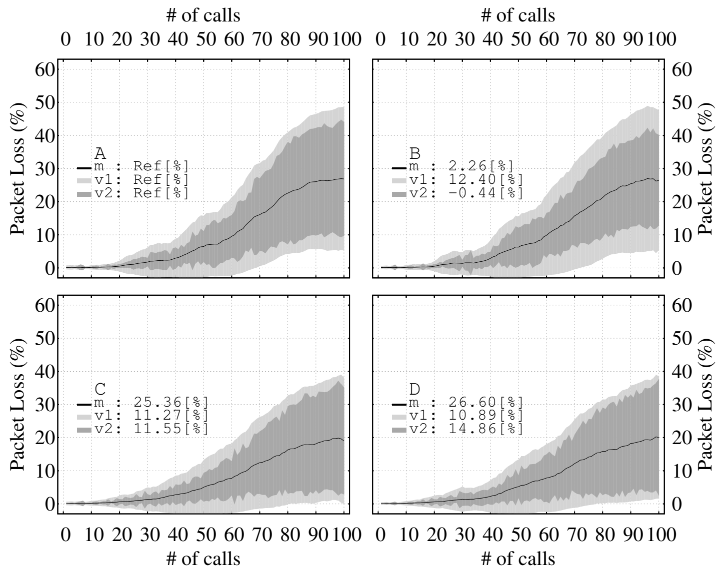

Figure 4.14: Packet loss versus number of injected calls on the uplink. *Legend: m represents the $\mu(\mu(PacketLoss))$, v1 represents $\mu(\mu(PacketLoss)) \pm \mu(\sigma(PacketLoss))$ and v2 represents the $\mu(\mu(PacketLoss)) \pm \sigma(\mu(PacketLoss))$.*

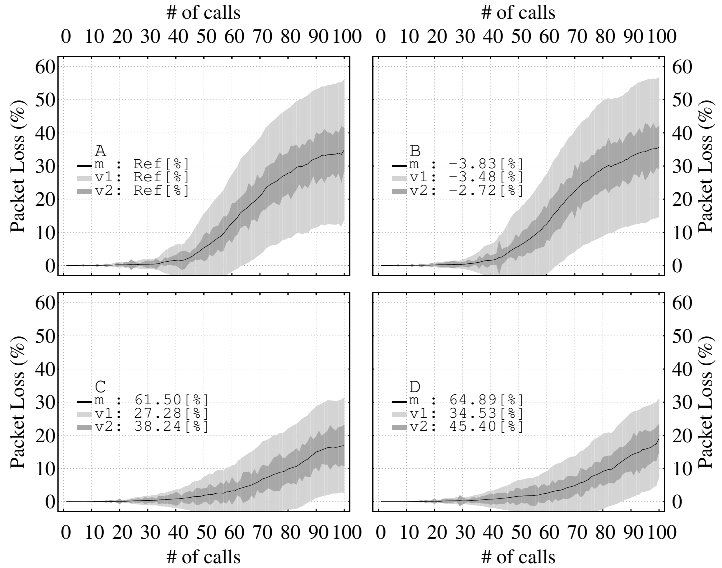

Figure 4.15: Packet loss versus number of injected calls on the downlink. *Legend: m represents the* $\mu(\mu(PacketLoss))$, *v1 represents* $\mu(\mu(PacketLoss)) \pm \mu(\sigma(PacketLoss))$ *and v2 represents the* $\mu(\mu(PacketLoss)) \pm \sigma(\mu(PacketLoss))$.

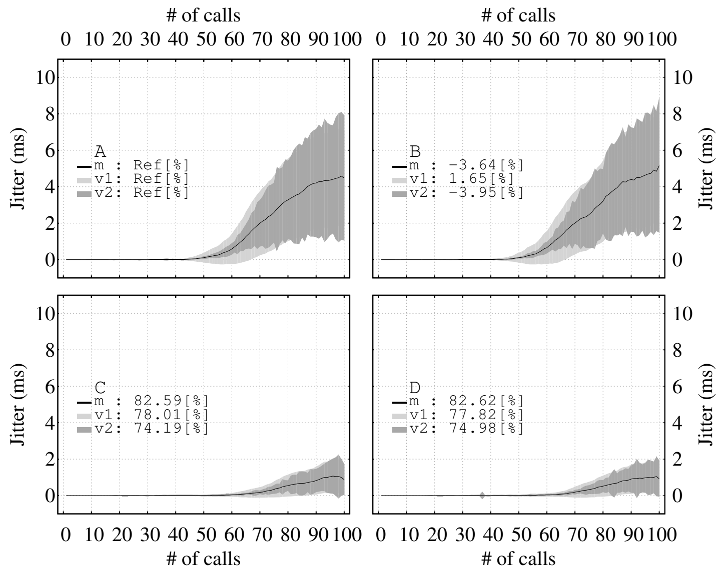

Figure 4.16: Jitter versus number of injected calls on the uplink. *Legend: m represents the* $\mu(\mu(Jitter))$, *v1 represents* $\mu(\mu(Jitter)) \pm \mu(\sigma(Jitter))$ *and v2 represents the* $\mu(\mu(Jitter)) \pm \sigma(\mu(PacketLoss))$.

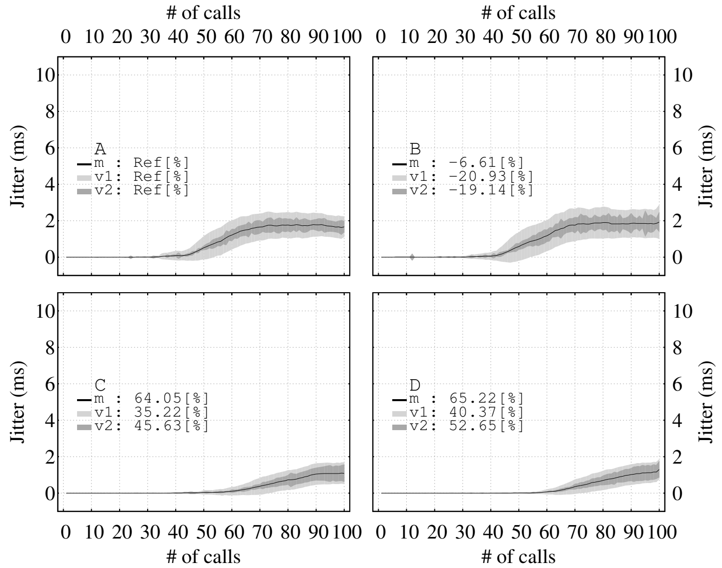

Figure 4.17: Jitter versus number of injected calls on the downlink. *Legend: m represents the $\mu(\mu(Jitter))$, v1 represents $\mu(\mu(Jitter)) \pm \mu(\sigma(Jitter))$ and v2 represents the $\mu(\mu(Jitter)) \pm \sigma(\mu(Jitter))$.*

These results are for a WMN scenario in which VoIP users are uniformly distributed across the network. This is chosen in such a way as to not bias the results, however in a scenario in which users are not uniformly distributed, the improvement achieved by the DPBM would be even more significant.

4.3.5 Summary

This work proposes a novel packet prioritisation and aggregation mechanism for improving the quality and capacity of VoIP in WMNs. The absolute network delay of packets traversing the WMN is accurately measured using the proposed DPBM. The delay thus obtained is used to prioritise more delayed VoIP packets over less delayed VoIP packets. The results show that the proposed DPBM improves the system's capacity and fairness, by providing a more even distribution of packet delay across the WMN.

Extensive simulations were performed in NS3 to validate the proposed mechanism. Results show performance increases in call quality, call quality variation, VoIP capacity and fairness of the call quality distribution over a realistic WMN architecture.

The proposed mechanism performs better on the downlink path of the VoIP packets, as the aggregation mechanism is able to build larger aggregated frames which carry the more delayed VoIP packets that are prioritised by the DPBM. The variations of the call quality, delay, packet loss, and jitter, decreased by about 7% in the case where the proposed DPBM enhanced the performance of the aggregation mechanism. The decrease in variation means the network acts more fairly in distributing the perceived call quality. In addition to the reduction of variation, the achieved VoIP capacity has also increased, when using the proposed DPBM, by about 3 calls, which represents an improvement of about 6% in the number of supported VoIP calls.

A key contribution of this section's work is providing a complete methodology for measuring the overall call quality and capacity of VoIP calls over a WMN. The work highlights some omissions in previous works. Specifically, omitting important statistical concepts such as the standard deviation of the measured call quality scores, can lead to overrated VoIP capacity values.

4.4 Summary

This Chapter introduced the iMOS concept, which is a call quality assessment method different from the existing ones, as the real-time call quality is accurately measured at an intermediate point in the path of VoIP packets.

The iMOS depends on accurate packet delay measurements, which depend on the underlying transport technology. The usage of the iMOS in two use-cases involving the latest wireless access technologies was studied. Specifically, the iMOS was tailored for the specifics of the LTE Femtocell Networks and 802.11-based Wireless Mesh Networks deployments.

The packet delay proves relatively easy to obtain in LTE femtocell deployments as LTE time synchronization requirements reduce the packet delay calculation to a mere time difference calculation. The iMOS proves useful for the MNO by isolating network issues, which could lead to unsatisfied femtocell users. It was shown that in the absence of the iMOS measurements, the MNO would not be able to locate network issues on the path between the caller and the callee. A DSL backhaul connection was considered for the femtocell traffic backhaul, in which the DSLAM was indicated as a possible bottleneck.

In the last part of the Chapter, the 802.11-based Wireless Mesh Networks are considered as the wireless access technology for clients using VoIP applications. The delay calculation in this case proves more difficult to obtain, and a mechanism called the Delay Piggy-Backing Mechanism (DPBM) was proposed to address that. The iMOS can be calculated using the DPBM in the WMN gateway node, thus opening the way for the development of a large variety of call quality assurance solutions.

This Chapter stands as the basis for Chapter 5, where iMOS-based Call Admission Control solutions are developed for the two latest wireless access technologies investigated throughout this thesis.

CHAPTER

FIVE

IMOS-BASED CALL ADMISSION CONTROL

5.1 Call Admission Control for VoIP

Section 3.3 described the role of CAC in VoIP systems, the different types of CAC, and the work to date in this field. Chapter 4 described how the iMOS can be obtained in femtocell networks and WMNs. This Chapter brings the two concepts together and the result is the proposed iMOS-based CAC mechanism.

Conceptually an iMOS-based CAC mechanism, is a passive measurement-based CAC which uses the iMOS measurement as the base for the admission decision. Specifically, the iMOS is permanently monitored, and the decision regarding new call requests are taken based on aggregated values of iMOS samples. Typically, the iMOS is checked against a pre-established threshold, and falling under the threshold indicates the need to activate the CAC procedure.

The VoIP simulation experiments performed so far, indicated that a single extra call accepted in a network can be detrimental to the overall call quality. There are many other works which confirm this issue [63, 67–73]. Since the impact of accepting a new call request can not be determined before the request is actually accepted, two possible solutions exist to alleviate the problems caused by that extra call:

- Increase allocated resources: When the allocated bandwidth for the VoIP calls is a managed resource, then this allocation can be extended to allow timely delivery of the extra traffic, and avoiding buffer overflows leading to packet dropping.
- Dropping existing calls: When allocating extra resources is not possible, then the ultimate solution is to drop a few ongoing calls until the call quality is restored.

The first solution can be used in the case of DSL femtocell deployments, where the bandwidth allocated for VoIP calls in the DSLAM can be adjusted to maximize the number of accepted calls from a large number of call requests. This scenario is described below in Section 5.2.

The second solution is employed in WMN networks, where the entire available bandwidth is used, and thus extending this resource is not possible. Many criteria specifying which call to be dropped can be developed, however in this work the call with the lowest iMOS is the prime candidate for dropping. The VoIP over WMN scenario and the specifics of the developed iMOS-based CAC mechanism are described in Section 5.3.

5.2 iMOS-based CAC in DSL-backhauled Femtocell Deployments

5.2.1 Description

This Section addresses the problem of congestion that can occur when a large number of femtocells utilise DSL connections as a backhaul link. An iMOS-based CAC mechanism is proposed here to enable a CAC mechanism to prevent femtocells from accepting more calls, based on call quality measurements. Since the bottleneck is the DSLAM, a bandwidth negotiation algorithm is proposed between the femtocell operator (the MNO) and the DSL operator (the ISP). The algorithm developed is suitable for Self-Organizing

Networks (SONs) deployments and can be driven by a number of parameters that can be adjusted by operator-defined policies.

5.2.1.1 Targetted Femtocell Architecture

The network architecture shown in Figure 5.1 depicts a typical femtocell over DSL deployment scenario. This architecture is used as reference for the proposed iMOS-based CAC mechanism. It is comprised of multiple households and small offices that have HeNBs provided by the same MNO and connected through the same DSLAM to the MNO's EPC network, through the Broadband Network Gateway (BNG). It is assumed that each HeNB is connected to a DSL broadband router.

The femtocell connects to the HeNBGW through the customer's fixed broadband connection, typically a DSL connection. This critical backhaul segment is often neglected in both research work and in real femtocell deployments; it is assumed to be of secondary importance to the radio link between the mobile unit and the femtocell. Although the femtocells transport both voice and data, due to their real time requirements the voice connections are much more sensitive to constraints in the backhaul network where, it is often not possible to provide guaranteed network resources.

Each voice call transported via a DSL connected femtocell is essentially converted into a VoIP call and encapsulated into an IPSec tunnel (Figure 5.2) for transmission to the HeNBGW located at the edge of the operator's EPC network. The 3GPP have defined AMR and AMR-WB as the mandatory voice codecs for LTE deployments [104]. Each AMR VoIP call has a data rate in the region of 5 to 24 Kbps plus overhead, depending on the codec mode that is being used. The IPSec secured tunnelling, which is used by all femtocell deployments, the overhead becomes comparable to the size of the actual payload.

Upon egress from the femtocell each VoIP packet is tagged as voice using the Differentiated Services Code Point (DSCP) field of the IP header. This allows the DSLAM to identify it as VoIP traffic and provide low-latency prioritization. In this work it is assumed that each MNO operating femtocells utilizing the ISP's network can be individually identified on a per flow basis. This could be done by providing specific VLANs for each MNO using the ISP

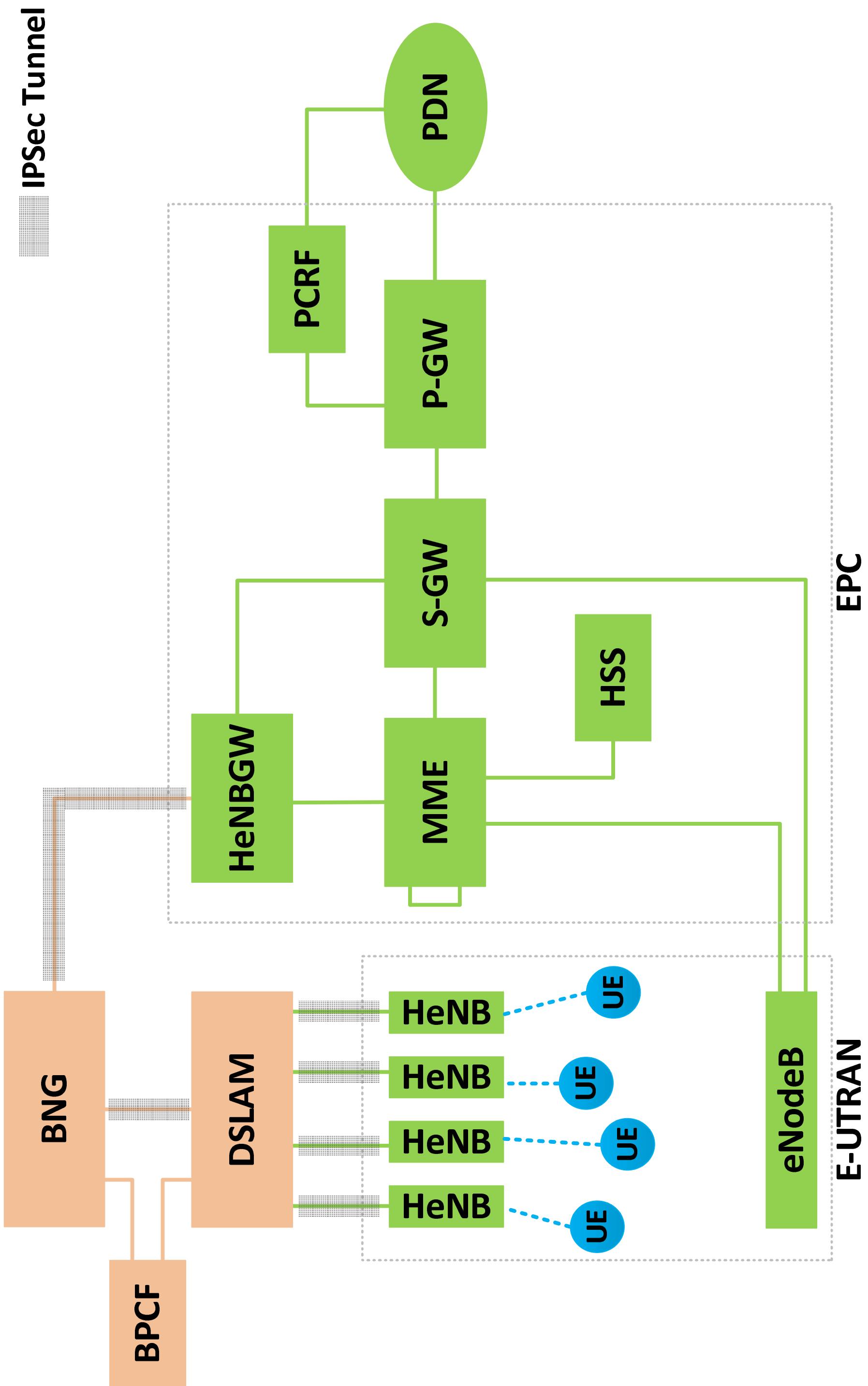

Figure 5.1: Architecture of Femtocell Deployment over DSL backhaul networks

Created in the UE	AMR Header	6	
	AMR Data	AMR SID 4	AMR 12.2 32
	RTP	12	
	UDP	8	
	IP	20	
	GTP-U	8	
	UDP	8	
	IP	20	
	IPSec ESP	8	
	PPPoE	12	
	Ethernet	42	

Figure 5.2: Voice over S1-U

network, therefore allowing the DSLAM to use the VLAN tag of each packet to identify the MNO to which it belongs.

5.2.1.2 The DSLAM is the Bottleneck

In this work it is assumed that each MNO will have SLAs in place with the ISP to provide a maximum bandwidth at each DSLAM dedicated for their VoIP traffic. This is often referred to in the literature as the Committed Information Rate (CIR) [105]. Typically, any additional traffic amount exceeding CIR is tagged as noncompliant and is subject to traffic policing. Various actions can be taken for traffic tagged as noncompliant. The most common action in such case, is packet dropping. Another action is to tag the traffic with a special marker, allowing it to pass through the outgoing queue, but at higher cost. In other words, extra bandwidth is provided for that user's flow. The extra bandwidth is regarded as a bursty pattern, and is commonly referred to as Excess Information Rate (EIR).

At the DSLAM all voice traffic is forwarded through the Expedited Forwarding (EF) queue. The EF queue provides the highest service level and it is utilised for low latency and low bandwidth services such as voice. The EF's bandwidth is shared between users, each receiving a share specified by the CIR [105]. As the number of HeNB deployments increases it is likely that the increased voice traffic volume in the EF queue on DSLAMs will cause congestion and may exceed the CIR. This will lead to high packet loss rates at the DSLAM with a corresponding impact on the voice call quality. Therefore, the quality of the call depends also on the level of congestion on the backhaul link, besides being influenced by issues discovered in the literature regarding the LTE radio link.

Although the MNO could over-provision the dedicated resources (CIR) on each DSLAM, it would lead to increased costs. It is therefore of interest for the MNO to minimise the maximum bandwidth that is reserved on each DSLAM while maintaining high voice call quality for their customers.

5.2.2 iMOS-based Call Admission Control in Femtocell Networks

The solution to the above described DSLAM bottleneck issue can be solved by employing the proposed iMOS-based CAC and dynamic resource allocation mechanism. This mechanism is based upon the quality of on-going voice calls passing through the HeNBGW, which is captured by performing iMOS measurements.

The proposed iMOS-based solution is comprised of multiple steps:

- Permanently monitoring the iMOS in the HeNBGW.
- Filtering between Home Area Network (HAN) or DSLAM issues.
- Determining quality threshold crossing events.
- Managing CAC active periods.
- Negotiating temporarily EIR bandwidth for the existing calls with the ISP.

As described in Section 4.2, it is possible to measure the iMOS in the HeNBGW as the LTE standard imposes strict synchronization requirements, and as such the packet delay can be accurately obtained from timestamp differences.

Keeping track of each MOS value calculated could cause scalability issues as normally around 50 values are obtained per second per voice call. Thus, all iMOS values obtained in a time window are averaged. The mean value of these averages, or the grand mean, represents the overall call quality of all calls. It is assumed that the HeNBGW has knowledge about the DSLAM-to-femtocells pairs. In other words, the HeNBGW is able to calculate the grand iMOS mean per DSLAM. The grand iMOS mean is used as the basis of performing call admission control and resource reservation in the DSLAMs.

The flowchart in Figure 5.3 describes the decision process. The quality of all calls passing through the HeNBGW is continually monitored in real time and the grand iMOS mean value per DSLAM represents the impact of the associated DSLAM's EF queue on the call quality.

The algorithm runs on the HeNBGW for each DSLAM that serves the femtocells managed by the HeNBGW, and takes as input:

- The list of all calls served by the DSLAM.
- The mean iMOS for each call, obtained for the last monitored time interval.
- The default value of the CIR, specified in the SLA.
- The EIR value, specifying the maximum value of CIR that the HeNBGW is able to request from the DSLAM, also specified in the SLA.

As output, the algorithm generates following messages which are sent from the HeNBGW to the:

- **Femtocells:** These messages inform the femtocells whether to accept or not new call requests received from nearby users.
- **DSLAM:** These messages inform the DSLAM to modify the current value of the CIR (CIR_CUR) in the EF queue for the MNO which generated the request.

The algorithm is triggered at intervals of 0.5 second. The reason of choosing this value is based on the findings presented in [94, 106], which show that human subjects are able to determine a quality drop within 5 seconds. Taking that into account, the 0.5 seconds for the CAC loop interval gives the proposed CAC mechanism enough measurement granularity to prevent call quality degradation from being perceived.

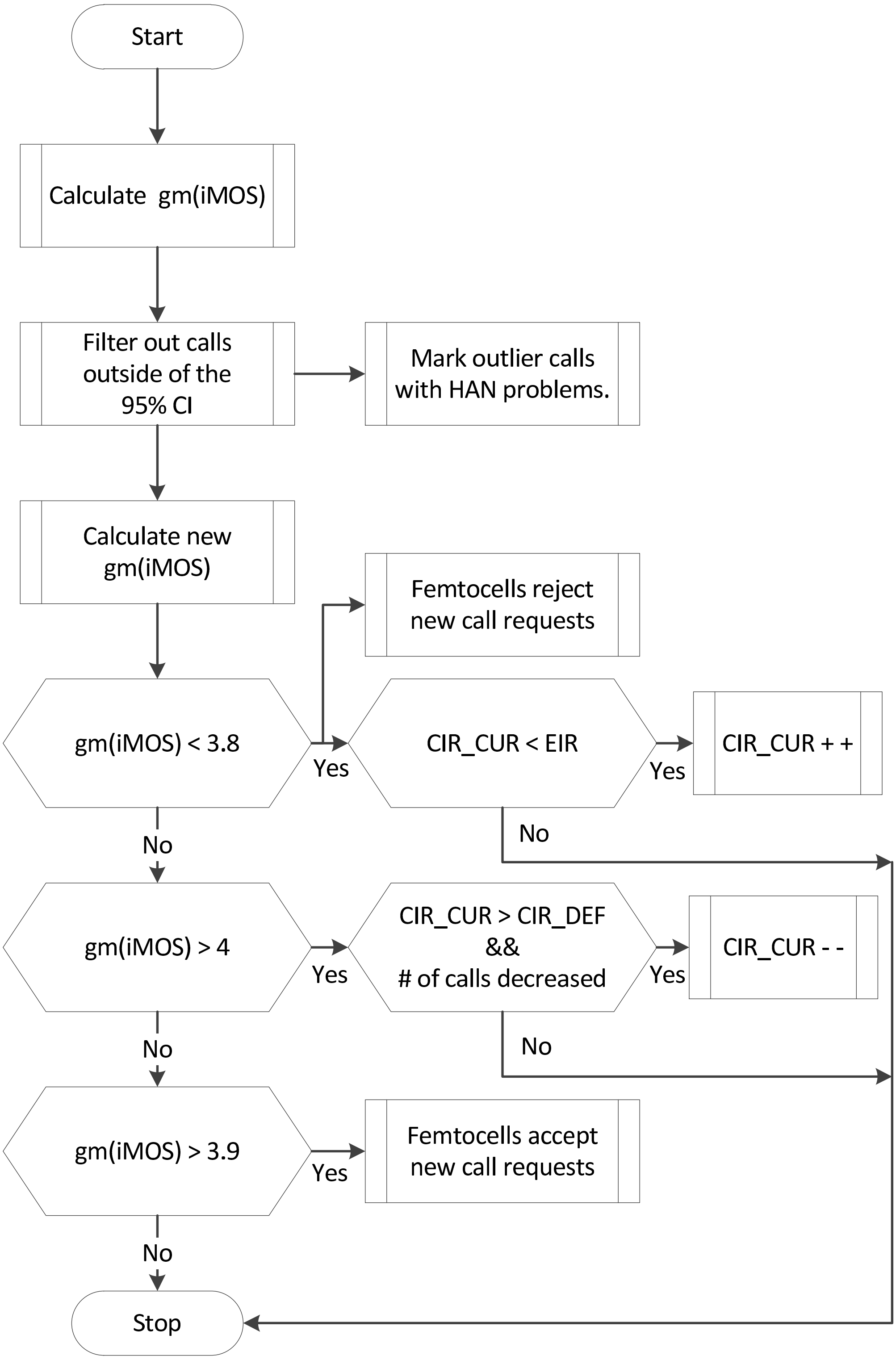

Figure 5.3: Logical diagram of the proposed CAC and Dynamic Resource Allocation Mechanism

The first part of the algorithm checks whether or not call quality degradation is caused due to HAN issues, per call, per DSLAM. This can be achieved statistically by eliminating the outlier calls falling outside a confidence interval around the grand mean of the iMOS, which is expressed in the diagram with *gm(iMOS)*. The 95% confidence interval range is chosen as the threshold for outlier detection. The outlier calls are filtered out and flagged as having HAN problems. A new grand iMOS mean value is computed for the remaining calls, which are those without HAN problems.

The second part of the algorithm verifies the grand iMOS mean against a pre-defined call quality threshold. The threshold chosen in this work is iMOS=3.8, and it represents the middle of the *Some users dissatisfied* call quality categorisation. However, this value can be adjusted by the MNO to any value, which should be higher than the value specified in the SLA. Typically no MOS values are specified in the SLAs yet, however based on the current trends in VoIP deployments, a MOS value or another call quality metric should be used as the basis for SLAs.

If this iMOS value is less than 3.8 then the HeNBGW informs all femtocells served through that specific DSLAM to reject new call requests. If the CIR value can be increased, then the DSLAM is asked to increase the CIR to allow its EF queue to clear the backlog of packets.

If the average gm(iMOS) becomes higher than 4.0, then the extra allocated CIR is decreased only if the number of ongoing calls has decreased. The CIR_-CUR value for the DSLAM's EF queue is decreased until it reaches the value of the default CIR that was specified in the SLA, i.e. CIR_DEF.

If the gm(iMOS) is between 3.9 and 4 then the femtocells are instructed to accept new call requests. It can be observed that the iMOS zone between 3.8 and 3.9 does not cause any actions in the network. This zone is used as a buffer zone, to avoid causing the algorithm to have a ping-pong effect.

It is also important to note that the communication between the MNO and the ISP entities is not as direct as it is between the HeNBGW and the femtocells. Two ways of creating the required interface between the MNO and the ISP are proposed. These are depicted in Figure 5.4:

- *Option 1* would be a proprietary interface developed by the ISP to allow

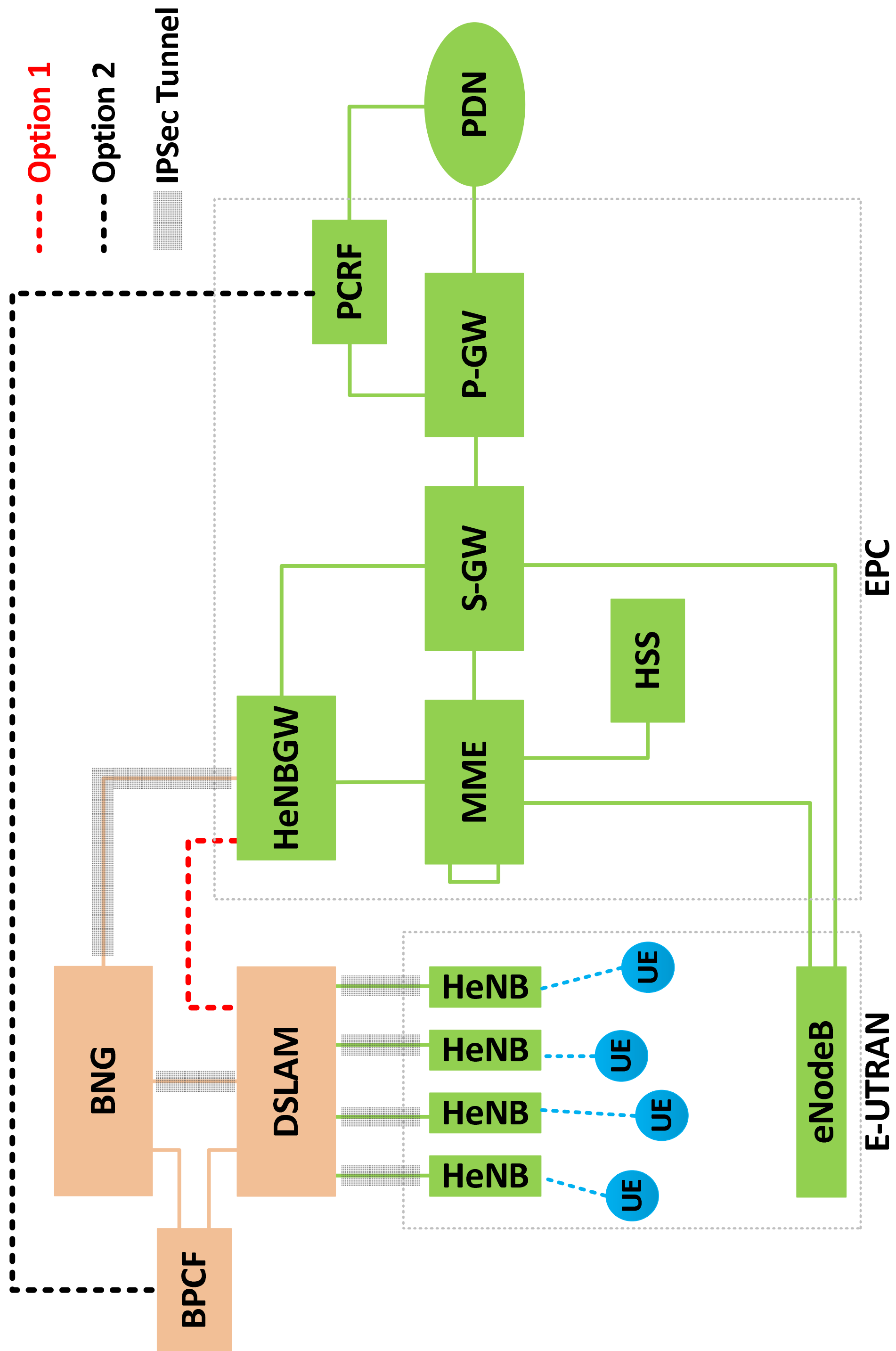

Figure 5.4: Femtocell Deployment Architecture with emphasis on MNO-to-ISP massage interchange

the MNO to have limited dynamic resource allocation capabilities in the DSLAM.

- *Option 2* would utilise an interface being defined by both the Broadband Forum and 3GPP for fixed/mobile convergence [26]. This PCRF to Broadband Policy Control Function (BPCF) interworking would allow the MNO to push resource allocation requests to the fixed access network.

Option 1 is a direct way for the MNO to operate on the bottleneck issue, however proposing a new interface demands a lot of effort from the standardization bodies.

Option 2 is more desirable, as it already uses existing interfaces. It is an indirect way of enabling the HeNBGW and the DSLAM to exchange parameters.

An alternative approach to that described in this work would be to allocate resources the moment call requests are received in the HeNBGW without requiring any call quality monitoring. However, although we assume that there is an EF queue per MNO or subscriber, this queue may be shared by other EF traffic sources in the HAN network. As such, fixed resource allocation could not determine if the allocated resources were sufficient to support high quality voice.

5.2.3 Simulation Parameters

The proposed solution was implemented and validated through simulations using the Network Simulator 3 (NS3) [101]. The simulated scenario is depicted in Figure 5.5. It consists of UEs which generate VoIP traffic and are associated with femtocells. For ease, a wired link between the UEs and femtocells is used, as the radio link is not the scope of this work.

A total of 50 femtocells are simulated, each having 4 UEs associated. The figure of 50 femtocells was derived from the fact that the DLSAM has a subscriber capacity of around 1000, and about 5% of that is reasonably to envisage that it would potentially serve femtocells. All UEs are connected to the femtocell through a 10 Mbps low delay link. The femtocells are connected to the DSLAM using DSL links with a speed of 100 Mbps and a link delay of 1 ms.

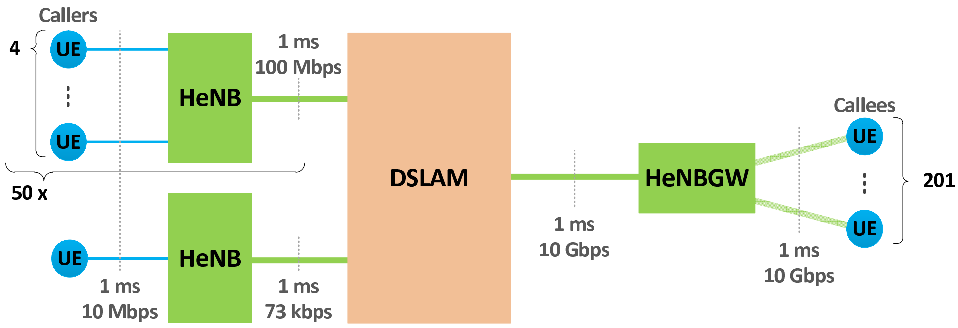

Figure 5.5: Simulation Setup

An additional femtocell serving one UE is used to create a HAN problem scenario, for which the link speed was decreased to 73 kbps.

In order to accurately emulate DSLAM behaviour, DSLAM recommendations from Cisco [105] were used to provide a number of DSLAM parameters including the buffer size used for traffic shaping which is based on link speed. A typical DSLAM egress link speed is 10 Gbps, as per datasheets [105]. The 10 Gbps link forwards the calls further to the HeNBGW where the iMOS is measured. The HeNBGW-DSLAM and HeNBGW-femtocell connections are done out-of-band, for simplicity reasons. A total number of 201 callees are simulated. For simplicity all the callees are connected to the HeNBGW via high speed, low latency connections.

The CIR value for the EF queue in the DSLAM was set to 10 Mbps and the EIR value was set to 15 Mbps. These values are chosen as such to lower the number of calls that have to be simulated. In reality these values may vary, but the concept presented here will be still valid. The mechanism residing in the HeNBGW verifies, out-of-band, the value of the current CIR setting in the DSLAM, in order to assure that the EIR limit is not exceeded.

The VoIP calls in the simulation are consecutively generated with an increment of 0.5 seconds plus a random variable in the range [0.00, 0.10] seconds with a granularity of 10 microseconds. This high time resolution is needed to assure source de-synchronisation. A simulation time of 300 seconds was chosen and 201 full-duplex VoIP calls are simulated with call durations ranging from 90 to 290 seconds. The voice calls are torn down in the same randomised manner as they are created.

For ease of implementation, only two modes of the AMR speech codec, i.e. AMR 12.20 and AMR SID were implemented in the NS3 VoIP applications. A simplified speech model is used in this work. Specifically, the switching between speech activities (i.e. speaking or silence), is specified by a speech activity parameter which is set for each call as a random value between 30% and 80%.

An estimate of the upper bound of the number of calls able to pass through the DSLAM without call degradation, is determined based on the values of 10 Mbps set for the CIR. Assuming all headers and AMR_12.2 payload size specified in Figure 5.2, then a total of 176 bytes are transmitted per VoIP packet. Based on this, the data rate per call is computed as:

$$176\frac{bytes}{packet} * 8\frac{bits}{byte} * 50\frac{packets}{sec} = 70.4\frac{kilobits}{sec}$$

Using a CIR value of 10 Mbps in the EF queue, the maximum number of supported calls can be calculated as:

$$10\frac{megabits}{sec} : 70.4\frac{\frac{kilobits}{sec}}{call} \approx 140\ calls$$

In other words, it is expected that the results will show CAC activity around the value of 140 calls.

The iMOS measurements in the HeNBGW computes the grand iMOS mean of all on-going voice calls every 0.5 seconds. This value was chosen as a reasonable trade-off between maintaining high quality voice calls and reducing the level of overhead.

5.2.4 Results Analysis

This section presents the simulation results obtained using the setup described in the previous section. In order to obtain a baseline for the results, a simulation was performed without the proposed iMOS-based CAC and Dynamic Resource Allocation Mechanism in place. The results of this simulation are depicted in Figure 5.6 and show the average MOS and the number of on-going calls over time.

As can be seen, without the proposed mechanisms in place, all calls are accepted by the femtocells and voice packets are forwarded from the EF queue

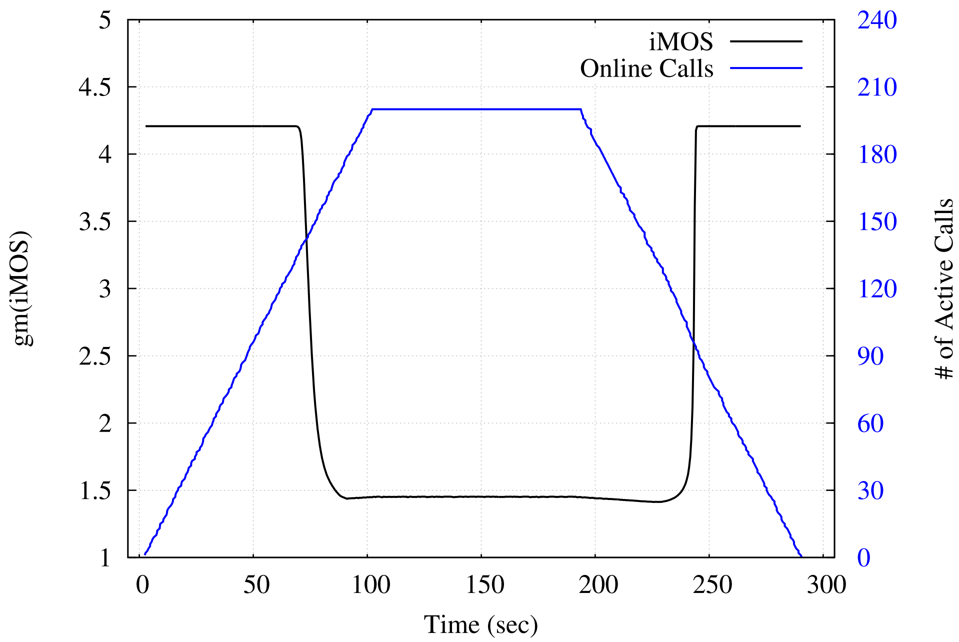

Figure 5.6: iMOS and Number of Active Calls versus Time when the iMOS-based CAC and Dynamic Resource Allocation Mechanism is Disable

as soon as they arrive. As the number of calls increases, the bandwidth limitation on the EF queue will cause packets to accumulate, until at some point the buffer will overflow resulting in dropped packets. At this point the gm(iMOS) degrades rapidly. From Figure 5.6 it can be determined that the maximum number of calls that the DSLAM can accommodate is approximately 140, given our specific simulation setup and assumptions. This confirms the expectations raised after this figure was determined analytically in the section describing the simulation setup (Section 5.2.3).

The fact that all voice calls are degraded draws attention to the difference between circuit switched and packet switched voice traffic. In circuit switched networks dedicated time slots are allocated for voice traffic at call setup time, while in packet switched networks voice packets from different voice calls potentially share the same packet forwarding capacity at each network router. In other words, accepting all call requests results in EF queue overload which further leads to quality degradation. This further highlights the need for a CAC.

Figure 5.7 presents results for the simulation when the iMOS-based CAC

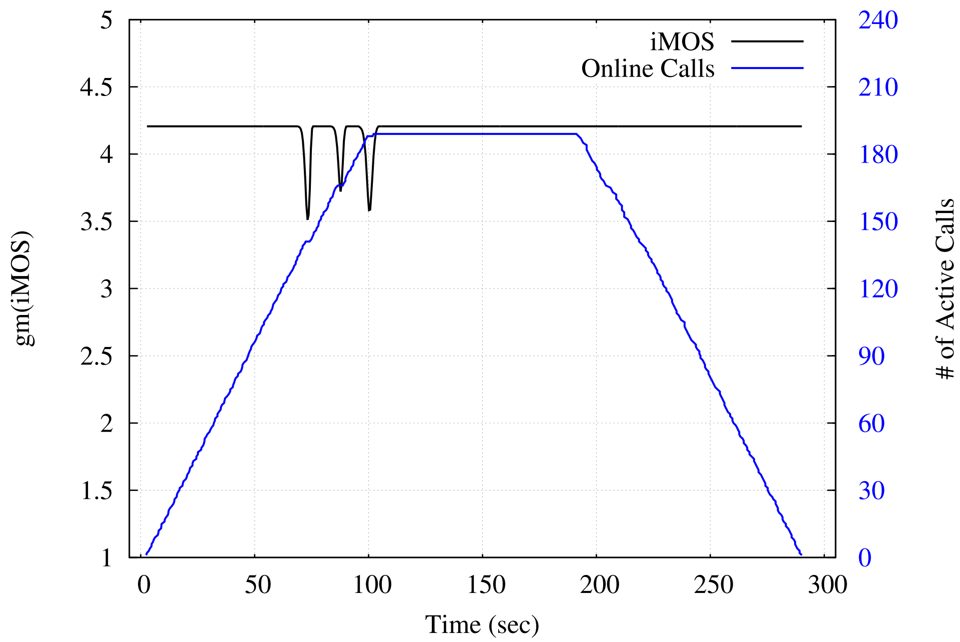

Figure 5.7: iMOS and Number of Active Calls versus Time when the iMOS-based CAC and Dynamic Resource Allocation Mechanism is Enabled

and Dynamic Resource Allocation Mechanism is enabled. It can be seen that when the gm(iMOS) drops, the feedback mechanism temporarily rejects any new call requests and consequently the gm(iMOS) is restored to a high level. It can be observed that the first time the gm(iMOS) drops under the desired quality level, the total number of active VoIP calls is around 140, which again validates the assumption based on the numerical analysis.

Figure 5.8 shows a plot of the CIR modification requests made by the algorithm running in the HeNBGW and granted by the DSLAM. It can be seen that every time the gm(iMOS) falls below 3.8, the HeNBGW requested more bandwidth in the EF queue. The algorithm allows new calls once the gm(iMOS) exceeds 3.9, and new calls are accommodated in the EF queue until the capacity is again reached. This situations repeats a few times, until the current CIR in the DSLAM's EF queue reaches the maximum values it can have, specifically the EIR. Once that threshold is reached, no more calls are allowed to enter the system.

This extra bandwidth is released only when the gm(iMOS) has returned to a high value, specifically higher than 4, and the number of active calls has de-

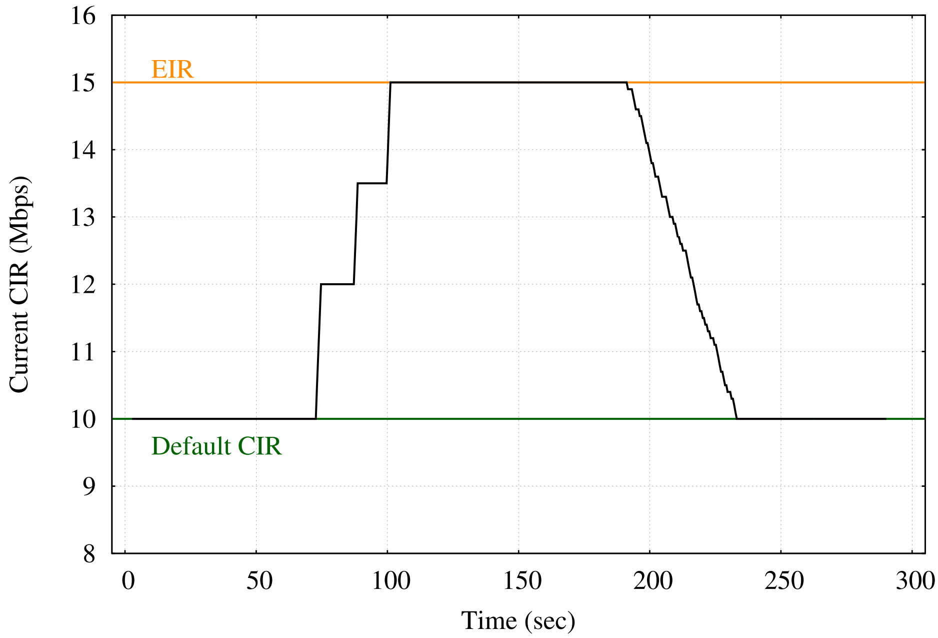

Figure 5.8: The fluctuations of the current CIR value in the DSLAM's EF queue

creased. Since any extra reserved bandwidth exceeding the default CIR would involve a cost to the MNO, increased bandwidth requests are only made when necessary, thereby maintaining a trade-off between the extra bandwidth requested and the number of accepted calls. In order to avoid sudden drops in call quality, the steps in which the bandwidth is released are smaller than then steps used to request the extra bandwidth.

It is noteworthy that the call placed through the femtocell located in a HAN with DSL connectivity issues, was detected as outlier of the 95% confidence interval and filtered out from the calculation of the gm(iMOS) value.

5.2.5 Summary

This section described an iMOS-based CAC and Dynamic Resource Allocation mechanism tailored for femtocell deployments with DSL backhaul. The proposed mechanism used real time call quality measurements in the HeNBGW to dynamically provision DSL resources when the monitored quality degraded.

The integration of an LTE femtocell network with existing fixed broadband DSL networks was also described. Specifically, two possible interface options to allow femtocell networks to provision resources in DSLAMs are provided. The option preferred involves existing interfaces between the MNO and the ISP.

The validation of the proposed mechanism was done using NS3-based simulation analysis. The simulation results demonstrate the capability of the solution to maintain high levels of quality for all voice calls and to dynamically adapt resource allocations based on changes in call quality, while maximizing the number of calls accepted in the network. Thus, this approach provides a working solution to issues of congestion in the backhaul of femtocell deployments.

5.3 iMOS-based CAC in WMN-backhauled Femtocell Deployments

5.3.1 Description

This section aims at bringing together the two wireless access categories investigated in the previous sections. Namely, the LTE wireless access technology is used to grant connectivity to end-users via femtocells, while the WMN access technology is employed as part of the backhaul network.

Wireless access technologies have increased in popularity due to the widespread use of smart-phones. The amount of data traffic has surpassed voice traffic driving both the Mobile Network Operators (MNOs) and wireless access standardization bodies to adapt accordingly. Widespread LTE deployments are being undertaken in an effort to cope with these increased traffic demands, while in parallel, operators are looking to complementary technologies, such as femtocells and WiFi, in an effort to reduce congestion on the cellular radio access networks.

Reducing the radius of cell towers is a possible solution to increase the performance of cellular networks and increase frequency reuse [107]. Femtocells are small access-point-sized devices with much smaller coverage areas when compared to typical cellular base stations. Unlike traditional base stations, femtocells are backhauled over normal IP connected networks such as residential Internet connections.

The focus of this work is to increase the cellular capacity in a defined region where femtocells are deployed. This is particularly beneficial for transient deployments; such scenarios include, but are not limited to, social events where attendees need access to voice and data services at venues which are used for a relatively short period of time. In such scenarios, the networking infrastructure needs to be flexible and quickly deployable.

A backhaul solution where each femtocell is provided with wired infrastructure is not feasible, as wired deployments tend to pose serious logistical problems. This work proposes a new deployment scenario which utilises a WMN backhaul infrastructure for femtocells, in order to overcome the deployment difficulties imposed by wired solutions. WMNs have become increasingly popular due to their capabilities: relatively large coverage areas with minimal cabling requirements, quick deployment, fair price, and ease of maintenance.

This work considers the usage of a 802.11-based WMN infrastructure, where the mesh nodes are hybrid stations featuring multiple 802.11 interfaces and one LTE femtocell embedded or co-located with the mesh node. Figure 5.9 depicts the scenario this work is focused on. It can be seen that clients roam with their UE in an area serviced by a grid of femtocells backhauled over a WMN network into the MNO's EPC network.

The WMN access-medium's capacity saturates when traffic demand is high, resulting in high latency and packet loss causing the performance of services to degrade. A special case of such services are voice calls. In LTE, both data and voice traffic payloads are encapsulated in IP packets, hence the voice calls in an LTE network are VoIP calls.

WMNs can be highly unpredictable in terms of QoS offered to VoIP calls, hence the VoIP call quality on WMNs degrades rapidly when only one call more than the system's capacity is added. In such situations detecting the capacity barrier is critical to ensure ongoing calls are guaranteed with satisfactory QoS levels.

This work in this Section compares the individual and combined effects of three mechanisms designed to improve the call quality and capacity in a femto-over-mesh deployment scenario:

1. A CAC mechanism which uses samples of the calls' quality to determine when to restrict the addition of new calls into the network;
2. The frame aggregation feature of the 802.11e wireless standard, allowing the aggregation of multiple small frames into a single frame, thus reducing the delays induced by the transport protocol's overhead;
3. A proposed delay-piggy-backing mechanism which attaches to every VoIP packet, while inside of the WMN, the cumulative delay experienced as it propagates through the network — this enables the scheduler to choose from a PIFO queue more delayed packets first, and the WMN gateway to compute call quality samples based on which the CAC mechanism takes actions regarding new call requests.

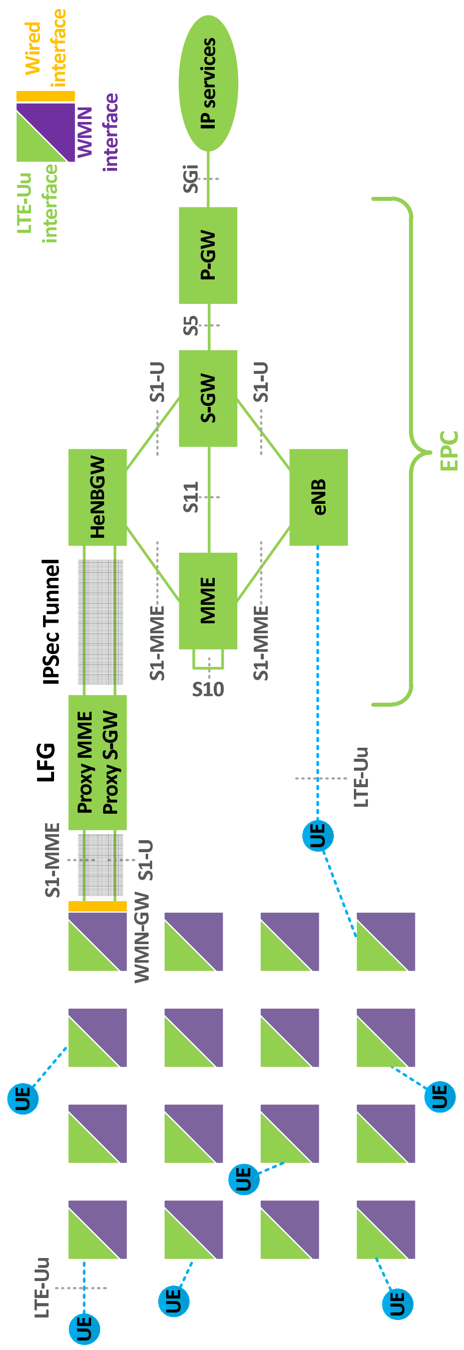

Figure 5.9: Topology of WMN-backhauled LTE femtocells

The CAC mechanism resides in the Local Femto Gateway (LFG) [35] (see also Section 2.2.2) which is an entity placed in the neighbourhood of the

WMN's gateway. Simulation results show that a CAC mechanism is necessary in such scenarios due to the fact that uncontrolled additions of voice calls into the network will negatively impact all existing calls. The overall call quality, measured by the iMOS's grand mean, and VoIP capacity are the two performance indicators analysed in this work.

5.3.1.1 Targeted WMN & LTE Femtocell Architecture

The reference architecture of this work is depicted in Figure 5.9. It is comprised of elements that have already been discussed, however this diagram brings together the LTE architecture and a WMN deployment. It can be seen that each WMN node incorporates a femtocell in addition to the 802.11 interfaces needed for the backhaul connectivity. The linking element in this new architecture is the LFG, which has the role of connecting the two networks in a combined effort to maintain call quality while maximizing the number of calls accepted in the system. The WMN gateway (WMN-GW) is equipped with an additional wired interface, thus creating a bridge between the wireless and wired transport networks.

The LTE security requirements specify that all traffic exchange between a femtocell and the EPC needs to be transported over IPsec tunnels. The tunnel terminates at the entry point of the EPC, which in Figure 5.9 is depicted as the HeNBGW.

Inside the EPC, the MME terminates the control-plane signalling from femtocells and UEs. The S-GW handles the data plane from the UEs. The P-GW is the anchor point of the UEs to the Internet and manages the IP addresses domain.

The LTE user protocol stack is depicted in Figure 5.10. It can be observed that the voice codec considered in this work is AMR, which is one of the voice codecs required to be supported by defaults in LTE devices.

5.3.1.2 iMOS-based CAC Mechanism in WMN-backhauled LTE femtocells

One of the roles of the MME is to manage the bearers and connections with the UEs. Among these functions, the relevant one for this work are the accep-

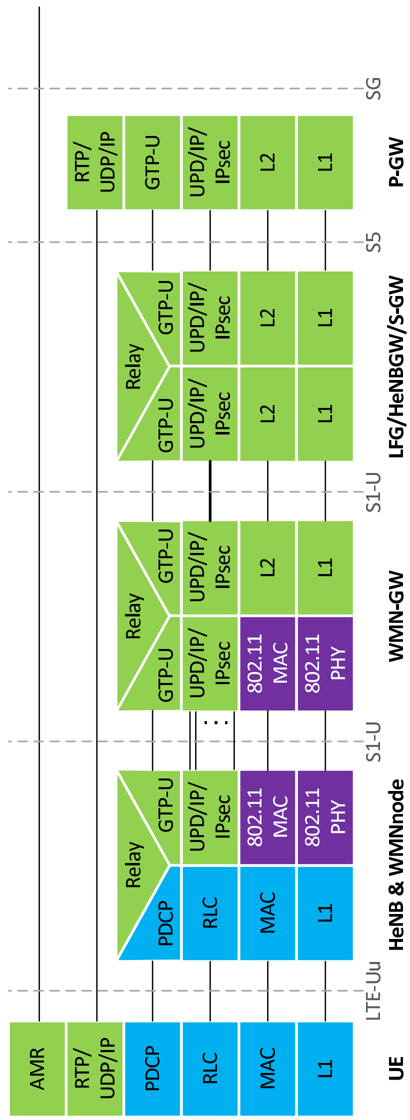

Figure 5.10: Protocol stack of WMN-backhauled LTE femtocells

tance of new calls and tearing down calls when such action is required. The CAC mechanism proposed here is based on using these functions in order to maintain the QoS level for the VoIP calls inside the WMN. However, the MME entity is not an appropriate node to place the CAC mechanism for scalability and overhead reasons. A better place is the LFG, which acts as a proxy for some of the MME's functions.

Zdarsky et al. [35] have shown that the LFG is able to intercept control messages using its Proxy-MME function. This enables the LFG to take actions regarding new or existing calls, based on the iMOS reported by the WMN.

In this work, the DPBM (see Section 4.3.2) is used to enable the WMN-GW to obtain an accurate measurement of the packet delay from the delay-value attached to each VoIP frame coming from the WMN. This accurate delay measurement enables the WMN-GW to measure the call quality. Since the WMN-GW is an intermediate point in the calls' path, the WMN-GW uses the delay measurement to obtain iMOS measurements.

The WMN-GW keeps track of all ongoing VoIP calls and their corresponding iMOS scores. The iMOS scores used by the mechanisms presented in this work are obtained only from the up-link traffic. However, the mechanism can be employed also on the down-link, but would require each WMN node to be iMOS-aware and to have the means to inform the WMN-GW about possible issues. Both of these assumptions would lead to an undesirable increase in the load on the WMN nodes and on the network traffic itself.

The iMOS values obtained from the up-link voice packets are the basis of the CAC mechanism. CAC can be invoked automatically or manually, allowing a network operator to prevent extra traffic from being injected in the network when certain conditions occur. Usually, the conditions revolve around maintaining a pre-established QoS level.

The CAC mechanism is controlled by the LFG based on call quality reports received from the WMN-GW, i.e. the WMN node in Figure 5.9 equipped with a wired interface.

This work assumes the usage of a LFG placed in the architecture as indicated by Figure 5.9. The LFG is able to intercept and interpret signalling messages between the femtocells and the EPC. In this way the CAC mechanism can take actions regarding new and existing calls.

As VoIP applications generate tens of packets per second, it would be infeasible to take a CAC decision based on every VoIP packet traversing the WMN-GW. However, the WMN-GW calculates the iMOS for each traversing VoIP packet and places the value into an accumulator. Periodically the CAC mechanism, residing on the LFG, will request the WMN-GW to provide one aggregated value which represents the overall iMOS of all ongoing calls during the last period. If that value is under a certain pre-established threshold, then CAC is activated and new call requests are rejected.

We define the network's capacity as being identified when the addition of the most recent call causes the grand mean of the iMOS values across all calls to drop below some threshold. In addition to activating the CAC mechanism when the overall iMOS value falls under the threshold, the CAC mechanism in this scenario, drops the call with the worst iMOS. A few calls are dropped in this manner until the network reaches a steady state, after it was destabilized by the extra call accepted above the capacity.

As the CAC mechanism periodically checks the aggregated iMOS value, it is important to determine a proper rate at which this checking is done. According to [106], human subjects are able to determine a quality drop within 5 seconds. Taking that into account, 1 second for the CAC loop interval is used, allowing the mechanism to drop a few calls, hence restoring the actual call quality before the user can detect it.

5.3.2 Simulation Parameters

The simulations performed in this experiment are broadly similar to those used in the earlier WMN work presented in Chapter 4. The simulation parameters are presented in Table 5.1. A WMN grid of 16 nodes is simulated using NS-3.10 [101]. The WMN nodes are equipped with two 802.11a interfaces for backhauling femtocell traffic. For simplicity the UE applications are placed directly on the WMN nodes.

The queues used in the simulation have a maximum capacity of 50 packets, as this is the queue size used by the most widespread wireless drivers, i.e. *MadWiFi* [102] and *ath5k* [103].

Table 5.1: Simulation Setup

Parameter	Value
Simulator	NS-3.10 [101]
Topology	Grid 4x4
Distance between nodes	100 m
Number of interfaces	2
WiFi Mode	802.11a
WiFi Data Rate	6 Mbps
Network Access Method	CSMA-CA
Propagation Model	LogDistancePropagationLossModel
Error Rate Model	YansErrorRateModel
Remote Station Manager	ConstantRateWifiManager
WiFi interfaces queue size	50 packets per AC
Early-drop threshold	250 milliseconds
Routing Algorithm	Fixed routes, pre-discovered by OLSR
No. of injected calls	100
Call Duration	Exp. dist. between 160 and 40 seconds
Call Direction	Full-duplex
Voice codec	AMR_12.20 & AMR_SID modes only
Speech model	ITU-T/P.59 [6]
iMOS threshold	R=70 (MOS=3.6)
CAC loop interval	1 second
Number of simulation epochs	10

The impact of the packet delay on the MOS is significant when its value is higher than 150 ms [31], and values higher than 400 ms render a conversation as non-interactive. Therefore the AC_VO queue in the simulations employs an early-dropping mechanism by removing packets when their queueing delay on a node becomes larger than 250 ms.

The routes are statically assigned so that statistically significant conclusions from the results can be drawn. However, it is worth noting that OLSR [43] was used for initial route discovery.

A total of 100 calls are injected into the network, which is higher than the expected capacity. Calls are injected sequentially and the inter-call arrival rate is exponentially distributed with a mean of 1 second. Figure 5.11 depicts the call duration and start times of all calls injected in the simulation. For consistency of results only the iMOS values obtained during the time span of the shortest accepted call are considered.

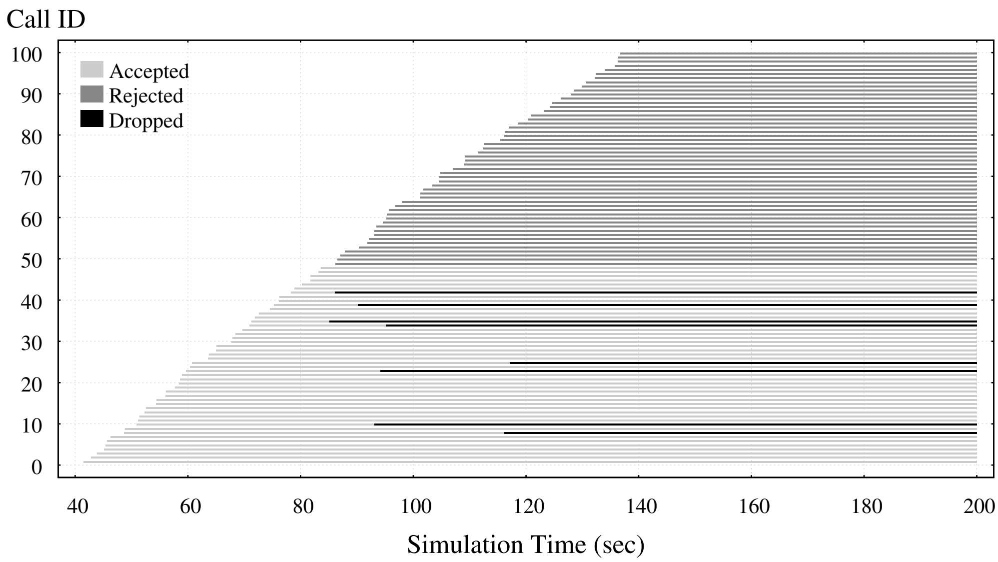

Figure 5.11: Call duration and status; dropped calls are the calls which were initially accepted but the CAC decided later to drop, as their corresponding iMOS was low.

The calls are full duplex and use the AMR codec. For simplicity only the 12.20 kbps and Silence Indicator (SID) modes are implemented. The speech model defined in [6] is implemented and used in the simulations to mimic realistic conversations. During an active period of the speech model, i.e. when someone speaks, AMR_12.20 packets are sent, and during silent periods AMR_SID packets are sent.

A threshold value of 3.6 for the iMOS is used by the CAC mechanism to detect if it is necessary to activate CAC, as on the MOS scale it represents the border between *Some users dissatisfied* and *Many users dissatisfied*.

5.3.3 Results

A comparison between different scenarios is presented in Figure 5.12. Each scenario represents a specific combination of features. There are eight possible combinations with the following cases: CAC enabled or disabled, frame aggregation (AGG) enabled or disabled, and delay-piggy-backing mechanism (DPB) enabled or disabled. Table 5.2 enumerates all the eight cases:

Table 5.2: Possible combinations between the three mechanisms: CAC, Frame Aggregation (AGG), and the Delay Piggy-Backing Mechanism (DPBM)

Case	**CAC**	**AGG**	**DPB**
A	0	0	0
B	0	0	1
C	0	1	0
D	0	1	1
E	1	0	0
F	1	0	1
G	1	1	0
H	1	1	1

The two diagrams in Figure 5.12 compare the cases related to the overall call quality observed by the users (left plot) and the overall system's call capacity (right plot). The middle of the filled box represents the average value, the margins of the box is the standard deviation around the mean, and the whiskers are the max and min values.

When CAC is disabled (cases A to D), the VoIP capacity diagram shows that all injected calls are accepted. However, the overall call quality suffered significant degradation which is beyond acceptable levels. The frame aggregation feature increases the overall call quality by a small amount (cases C and D), however not enough to satisfy the required MOS.

When CAC is enabled (cases E to H), the number of accepted calls is lower than the number of injected calls, but the overall quality is maintained at high levels on the MOS scale. When employing frame aggregation (cases G and H), the VoIP capacity increased by about 10 calls compared to the case when frame aggregation is not used (cases E and F), whereas the decrement on the quality

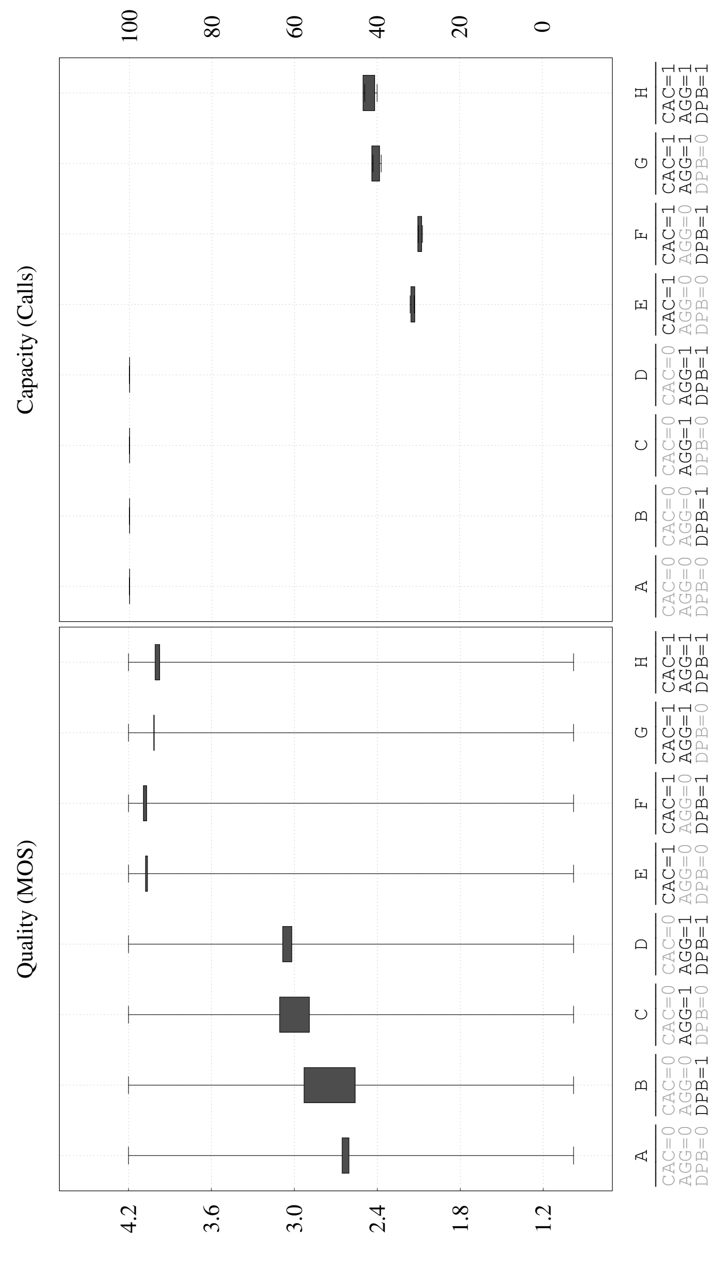

Figure 5.12: Comparison between all combinations resulting from using the CAC, frame aggregation, and delay-piggy-backing mechanisms in enabled or disabled mode.

scale is minimal. It can be seen that the DPBM (case H) allows the network to accept a few more calls compared to the case when it was not employed (case G) with imperceivable reduction in call quality.

The result clearly shows that there is an obvious need for a CAC mechanism to detect and act in situations where the number of call requests is higher than the network's capacity. Regardless of the number of injected calls, the three mechanisms enabled in the WMN assure a maximum VoIP capacity of 43 calls, with the simulation parameters presented above. In order to maintain the quality, some of the calls had to be dropped as shown in Figure 5.11.

The CAC mechanism is triggered by a timer to check the overall call quality and take a decision. The frequency of this periodic check allows the network operator to increase the overall quality or the VoIP capacity. A more frequent check allows the mechanism to detect quality degradations sooner and reject further requests, thus assuring a high overall call quality at the expense of VoIP capacity. The impact of the frequency of these checks was explored by performing a subset of the previous experiments with different loop values, the results of which are shown in Figure 5.13. These results show that there is a trade-off between capacity and quality, as expected. The value of 1 second between checks offers a reasonable operating point.

Another influencing parameter in the behaviour of the combination of mechanisms is the distance between the WMN nodes, or inter-node distance. Figure 5.14 shows that increasing the inter-node distance influences differently the overall call quality when comparing the situations where CAC is enabled (case A to D) or disabled (cases E to H). When CAC is disabled, the overall quality is unacceptable. It is worth noting that starting with 60 meters between the nodes, the wireless interference level drops allowing the call quality to increase, but still it remains under satisfactory levels. Beyond 130 meters the signal quality of the wireless radios cannot sustain proper communication thus causing the overall call quality to decrease.

When the CAC mechanism is enabled (cases E to H), an increase in inter-node distance results in a decrease in the overall call quality. However, the falling slope is small hence the quality is maintained to satisfactory levels for the entire investigated inter-node distance interval. The capacity is higher when frame aggregation is enabled (cases G and H), particularly when the

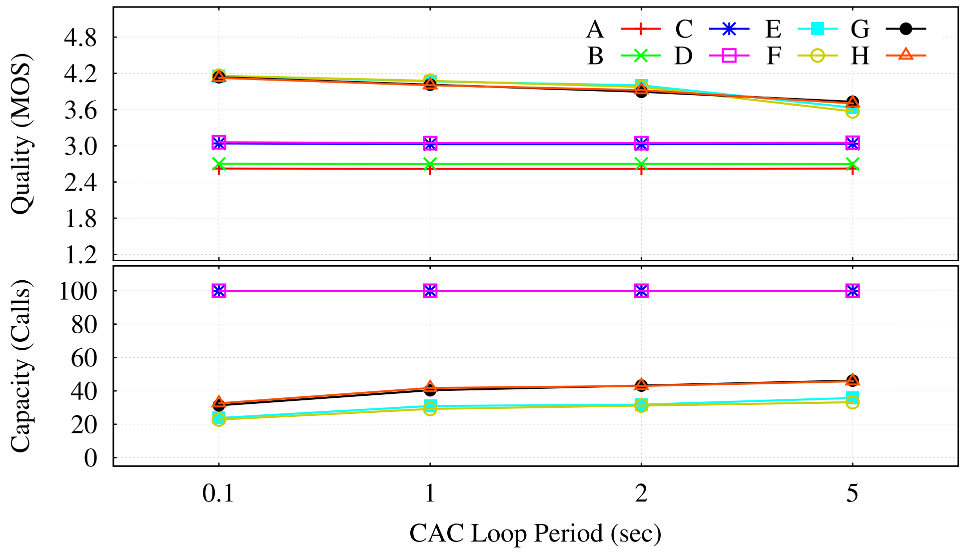

Figure 5.13: Influence of CAC Loop Period
on voice quality and capacity.

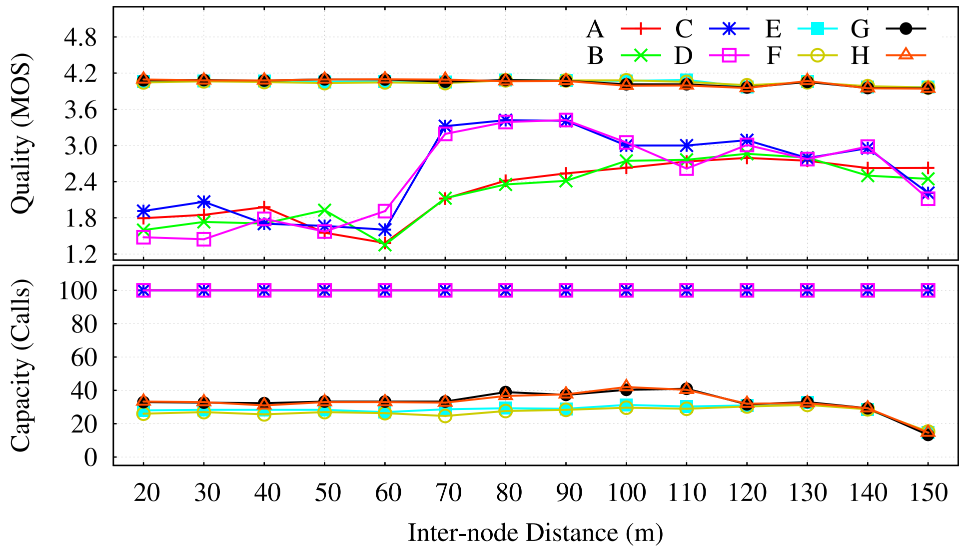

Figure 5.14: Influence of inter-node distance
on voice quality and capacity.

inter-node distance is between 80 to 110 meters. It is interesting to note that the severe interference for inter-node spacing less than 60 meters is remedied by the CAC process.

It can be observed from all results presented above that although the iMOS threshold used for the CAC is 3.6, the actual obtained overall call quality when CAC is enabled, is higher than 3.6. As the relationship between the iMOS CAC threshold and the actual obtained overall call quality is not known, simulations have been carried out to determine this relationship empirically. Figure 5.15 depicts this relationship, alongside with the relationship between the iMOS CAC threshold and the VoIP capacity obtained.

These two plots enable an operator to adjust their system parameters. These plots show the trade-off between quality and capacity but show an unexpected jump in performance when a threshold around 2.6 is used. This is due to the complex interactions in the system and the non-linear function used to calculate the MOS. Judging the results obtained with these network parameters, the system works optimally when the iMOS CAC threshold is chosen in the 2.8 to 3.8 range. In this range for case H, the overall quality varies between 3.7 and 4.2, while the capacity varies between 50 to 40 calls, respectively. This kind of result, would be highly effective to an operator trying to fine-tune the performance of the offered VoIP service.

5.3.4 Summary

The work presented in this Section proposed a) a novel architecture for femtocell deployment utilising WMNs as backhaul and b) mechanisms to assure high QoS for VoIP calls by employing an iMOS-based CAC. The results showed that such a CAC mechanism is needed in deployment scenarios involving a WMN as backhaul infrastructure. Furthermore, the usage of the proposed DPBM combined with the well known frame aggregation mechanism, showed improved capacity in terms of number of accepted calls in the network.

Three important parameters were varied over a range of values in order to test the flexibility of the proposed mechanisms. Namely, the frequency at which the CAC mechanism checks the network status, the WMN inter-node distance, and the iMOS CAC threshold. The results showed firstly that higher polling frequency preserves the call quality at the expense of capacity, however a polling interval larger than 5 seconds results in reduced and un-guaranteed

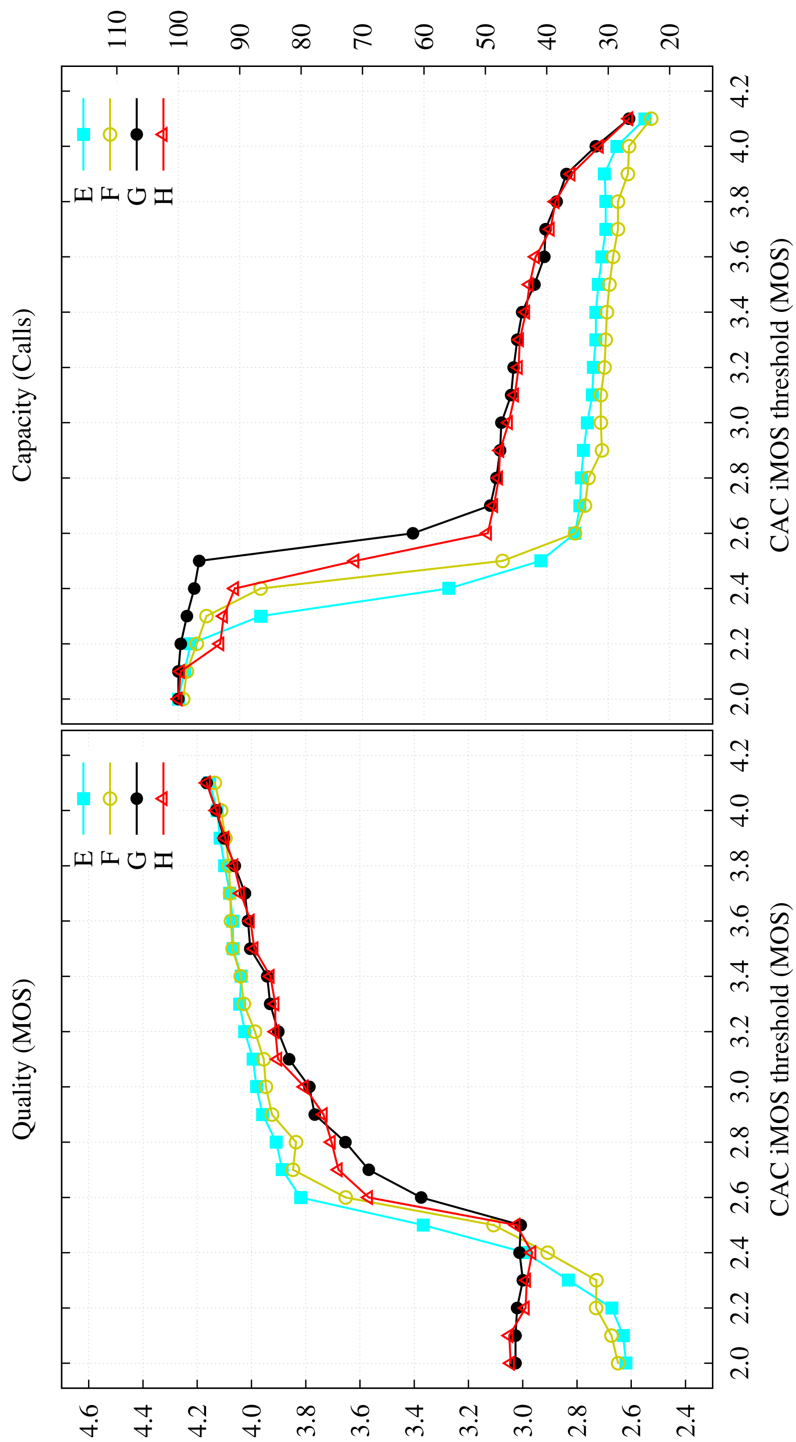

Figure 5.15: Influence of CAC iMOS threshold
on voice quality and capacity.

QoS. Secondly, when the inter-node distance was varied, the mechanisms are able to maintain high levels of VoIP call quality with increased VoIP call capacity around 30%, rising to 40% for distances between 80 to 110 meters. Thirdly, varying the iMOS CAC threshold generates a chart useful to any operator willing to fine-tune the overall call quality achieved against the maximum number of supported calls. Specifically to the parameters in this experiment, the iMOS CAC threshold range between 2.8 and 3.8 is recommended for better optimization.

5.4 Summary

This Chapter focused on using an iMOS-based CAC mechanism to protect networks from oversubscription of VoIP calls in DSL-based femtocell and WMN-based wireless access deployments.

In both cases the need for such CAC is showed by means of NS-3 simulations. When no CAC is used, the overall call quality drops for every call in the network, causing all users to be unsatisfied by the quality of the VoIP service.

When CAC is employed, the complementing QoS mechanisms are able to maintain the overall call quality at high levels, assuring user satisfaction. A second effect of the developed mechanisms is to maximize the number of accepted calls for a given call quality threshold.

CHAPTER

SIX

CONCLUSION

This work focused on identifying the need for a more in-depth Voice Over IP (VoIP) monitoring in deployment scenarios involving the latest wireless access technologies. As a result, the Intermediate Mean Opinion Score (iMOS) is proposed as a novel method of obtaining VoIP call quality measurements at intermediate nodes in the path of VoIP packets.

The iMOS was used to highlight network related issues occurring in two deployments using the latest wireless access technologies. The first investigated scenario is a femtocell deployment where the iMOS was presented as the main solution available for the Mobile Network Operator (MNO) to assess and correlate network status with the VoIP call quality delivered to the end-users.

In a second scenario, a Wireless Mesh Network (WMN) deployment over which VoIP is carried, was described. The results showed that in the absence of any method of protection against bursts of call requests, the network would quickly saturate and negatively affect the VoIP call quality. The iMOS was used as a means of measuring the impact of such bursts. The novelty of the approach consisted in the developing of the Delay Piggy-Backing Mechanism (DPBM) which makes the calculation of iMOS values possible. The DPBM also enabled forwarding WMN nodes to re-prioritise packets based on their global enqueueing delay. In order to increase the overall network's VoIP capacity, the proposed scheme was used in conjunction with the known frame aggregation scheme used in wireless networks to increase capacity. Using these solutions combined, the results showed an overall 12% network performance increase.

After showing the benefits of using the iMOS as a powerful monitoring tool, the work investigated a congestion scenario in which the wired Digital Subscriber Line (DSL) backhaul used for femtocells can cause call quality degradation. The results of the investigation showed that there is the need for a Call Admission Control (CAC) mechanism to prevent the addition of extra calls to the system when congestion was detected using the iMOS. It was observed that detecting the system reaching its VoIP capacity and activating CAC is not sufficient to restore the quality of the ongoing calls. A solution able to address these shortcomings was the development of a resource allocation mechanism used in conjunction with the iMOS-based CAC.

The same behaviour where employing CAC was not sufficient, was also observed in another scenario investigating a deployment in which femtocell VoIP traffic was backhauled into the core-network over a WMN. In this case, allocation of more resources was not possible and hence a few VoIP calls had to be dropped for the benefit of the other ongoing calls.

6.1 Future Work

Possible future improvements and extensions to the work presented in this work are:

- **Validation of iMOS measurements via test-bed** - This work uses mainly simulations for concept validation but, in order to verify that the models used are realistic, a NS3 emulation test-bed similar to the one in [3] can be deployed. Real VoIP calls can be established, and the iMOS values can be compared against other VoIP call quality assessment solution, such as PESQ.
- **Increase DPBM delay measurement accuracy** - The DPBM mechanism can be extended to include the time needed to transfer and to retransmit a packet. This will bring the measured delay value in the delay field closer to the actual packet delay. The implementation of the DPBM will

thus extend to the lower part of the MAC layer to be able to capture these times.

- **iMOS-based codec-rate adaptation** - When the iMOS detects call quality degradation in the VoIP calls, the VoIP packets which are already intercepted for quality check can be modified to instruct the end-devices to change the codec rate to a lower value, hence generating less traffic which further leads to network de-congestion.

- **Applicability to other time-sensitive traffic** - The algorithms and mechanism developed in this work are related but not limited to VoIP traffic. As an example, IPTV and other types of live video streaming can benefit from the enhancements brought by these solutions.

- **Using the iMOS as the basis for user's Service Level Agreement (SLA)** - Along with other network performance indicators such as throughput, delay, packet loss and jitter, the iMOS can be added to the list of elements negotiated between the user and the service provider. The iMOS can be specified as the value under which the quality of the calls should not decrease.

- **Real WMN deployment** - The iMOS and the DPBM have only been implemented in the NS3 simulation environment. In a real WMN deployment, actual Linux wireless nodes can be used to act as WMN forwarding nodes where the DPBM mechanism would be supported by implementing the Push-In-First-Out (PIFO) queue. The wireless mesh routing protocol needs to be modified to support the extra field where the cumulative queueing delay is stored and in the WMN gateway the iMOS calculation would be implemented at the application layer.

- **iMOS in a real femtocell deployment** - This would involve collaboration with a femtocell service provider, which usually are the same entities offering cellular services. With the provider's agreement, the iMOS calculation can be implemented in the Home evolved NodeB Gateway (HeNBGW) and used to monitor the actual call quality in addition to what the operator is already monitoring.

REFERENCES

[1] Cristian Olariu, John Fitzpatrick, Yacine Ghamri-Doudane, and Liam Murphy. Provisioning Call Quality and Capacity for Femtocells over Wireless Mesh Backhaul. In *2013 IEEE 24th International Symposium on Personal, Indoor and Mobile Radio Communications: Mobile and Wireless Networks (PIMRC'13 - Mobile and Wireless Networks)*, London, United Kingdom, September 2013. xv, 53

[2] Cristian Olariu, John Fitzpatrick, Philip Perry, and Liam Murphy. A QoS based call admission control and resource allocation mechanism for LTE femtocell deployment. In *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, pages 884 –888, Las Vegas, USA, jan. 2012. doi: 10.1109/CCNC.2012.6181184. xv, 53

[3] Cristian Olariu, Adriana Hava, Piotr Szczechowiak, and Yacine Ghamri-Doudane. Prototyping Telematic Services in a Wireless Vehicular Mesh Network Environment. In *WCNC 2012 Workshop on Wireless Vehicular Communications and Networks (W5 - Wireless Vehicular Communications and Networks)*, pages 371–376, Paris, France, April 2012. xv, 124

[4] Cristian Olariu, Mícheál Ó Foghlú, Philip Perry, and Liam Murphy. VoIP Quality Monitoring in LTE Femtocells. In *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pages 501–508, Dublin, Ireland, 2011. IEEE. xv

[5] Cristian Olariu, John Fitzpatrick, Yacine Ghamri-Doudane, and Liam Murphy. A Delay-Piggybacking–based Packet Prioritisation Mechanism for Voice over IP in Wireless Mesh Networks. *Elsevier Computer Networks*, 2013. xvi

[6] ITU-T P.59. Artificial Conversational Speech. 1993. 6, 23, 76, 114, 115, 148

[7] D. Cohen. Specifications for the Network Voice Protocol (NVP). RFC 741, November 1977. URL `http://www.ietf.org/rfc/rfc741.txt`. 9

[8] J. Postel. User Datagram Protocol. RFC 768 (Standard), August 1980. URL http://www.ietf.org/rfc/rfc768.txt. 9, 12

[9] J. Postel. Internet Protocol. RFC 791 (INTERNET STANDARD), September 1981. URL http://www.ietf.org/rfc/rfc791.txt. Updated by RFCs 1349, 2474, 6864. 9

[10] ITU-T H.323. Packet-Based Multimedia Communications Systems. 1999. 9, 48

[11] Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 1889 (Proposed Standard), January 1996. URL http://www.ietf.org/rfc/rfc1889.txt. Obsoleted by RFC 3550. 9

[12] Henning Schulzrinne and Mark Handley. Session Initation Protocol, 1999. 9, 48

[13] Skype. URL www.skype.com. 9

[14] H. Nyquist. Certain Topics in Telegraph Transmission Theory. *American Institute of Electrical Engineers, Transactions of the*, 47(2):617–644, 1928. ISSN 0096-3860. doi: 10.1109/T-AIEE.1928.5055024. 10

[15] ITU-T recommendation G.711. Pulse code modulation (PCM) of voice frequencies, November 1988. 11

[16] ITU-T Recommendation G.729. Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP), 1996. 11

[17] K. Srinivasan and A. Gersho. voice Activity Detection For Cellular Networks. *Speech Coding for Telecommunications, 1993. Proceedings., IEEE Workshop on*, pages 85–86, 1993. 11

[18] J. Postel. Transmission Control Protocol. RFC 793 (Standard), September 1981. URL http://www.ietf.org/rfc/rfc793.txt. Updated by RFC 3168. 12

[19] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard), July 2003. URL http://www.ietf.org/rfc/rfc3550.txt. 12, 60

[20] G. Almes, S. Kalidindi, and M. Zekauskas. A One-way Delay Metric for IPPM. RFC 2679 (Proposed Standard), September 1999. URL http://www.ietf.org/rfc/rfc2679.txt. 13

[21] C. Demichelis and P. Chimento. IP Packet Delay Variation Metric for IP Performance Metrics (IPPM). RFC 3393 (Proposed Standard), November 2002. URL http://www.ietf.org/rfc/rfc3393.txt. 13

[22] G. Almes, S. Kalidindi, and M. Zekauskas. A One-way Packet Loss Metric for IPPM. RFC 2680 (Proposed Standard), September 1999. URL http://www.ietf.org/rfc/rfc2680.txt. 14

[23] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205 (Proposed Standard), September 1997. URL http://www.ietf.org/rfc/rfc2205.txt. Updated by RFCs 2750, 3936, 4495. 15

[24] J. Wroclawski. The Use of RSVP with IETF Integrated Services. RFC 2210 (Proposed Standard), September 1997. URL http://www.ietf.org/rfc/rfc2210.txt. 15

[25] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC 1633 (Informational), June 1994. URL http://www.ietf.org/rfc/rfc1633.txt. 15

[26] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. RFC 2475 (Informational), December 1998. URL http://www.ietf.org/rfc/rfc2475.txt. Updated by RFC 3260. 15, 100

[27] Y. T'Joens, P. Crivellari, and B. Sales. Layer Two Tunnelling Protocol (L2TP): ATM access network extensions. RFC 3301 (Proposed Standard), June 2002. URL http://www.ietf.org/rfc/rfc3301.txt. 15

[28] V. Sharma and F. Hellstrand. Framework for Multi-Protocol Label Switching (MPLS)-based Recovery. RFC 3469 (Informational), February 2003. URL http://www.ietf.org/rfc/rfc3469.txt. Updated by RFC 5462. 15

[29] ITU-T P.800. Methods for Objective and Subjective Assessment of Quality. 1996. 16

[30] ITU-T P.862. Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. 2001. 17

[31] ITU-T G.107. The E-Model, A Computational Model For Use In Transmission Planning. 2005. 17, 18, 19, 20, 47, 56, 75, 76, 114, 138

[32] Alan D Clark, Ph D Fellow Iee, et al. Modeling the effects of burst packet loss and recency on subjective voice quality. 2001. 18, 49, 50

[33] ITU-T G.113. Transmission impairments due to speech processing. 2007. 18

[34] 3GPP Technical Specification 24.301. Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3 (Release 8). Technical report, 2008. URL www.3gpp.org. 29, 59, 60

[35] Frank A Zdarsky, Andreas Maeder, Sabah Al-Sabea, and Stefan Schmid. Localization of Data and Control Plane Traffic in Enterprise Femtocell Networks. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–5. IEEE, 2011. 30, 31, 109, 112

[36] Joshua Robinson and Edward W Knightly. A performance study of deployment factors in wireless mesh networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2054–2062. IEEE, 2007. 34, 36

[37] Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput. *IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009)*, pages 1 –565, 29 2009. doi: 10.1109/IEEESTD.2009.5307322. 35, 40

[38] Jangeun Jun and Mihail L Sichitiu. The nominal capacity of wireless mesh networks. *Wireless Communications, IEEE*, 10(5):8–14, 2003. 36

[39] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003. URL `http://www.ietf.org/rfc/rfc3561.txt`. 38

[40] D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728 (Experimental), February 2007. URL `http://www.ietf.org/rfc/rfc4728.txt`. 38

[41] Lei Wang, Lianfang Zhang, Yantai Shu, and Miao Dong. Multipath source routing in wireless ad hoc networks. In *Electrical and Computer Engineering, 2000 Canadian Conference on*, volume 1, pages 479–483. IEEE, 2000. 38

[42] Daniel Aguayo, John Bicket, and Robert Morris. SrcRR: A high throughput routing protocol for 802.11 mesh networks (DRAFT), 2005. 38

[43] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), October 2003. URL `http://www.ietf.org/rfc/rfc3626.txt`. 38, 114

[44] J. Chroboczek. The Babel Routing Protocol. RFC 6126 (Experimental), April 2011. URL `http://www.ietf.org/rfc/rfc6126.txt`. 38

[45] IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific

requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, 2012. doi: 10.1109/IEEESTD.2012.6178212. 38

[46] Medium Access Control (MAC) Quality of Service (QoS) Enhancements (Replaced by approved draft 802.11e/D13.0). *IEEE Std P802.11e/D13.0*, 2005. 39, 40, 73

[47] Junfeng Wang, Mingtian Zhou, and Yuxia Li. Survey on the End-to-End Internet Delay Measurements. pages 155–166, 2004. 44

[48] L. De Vito, S. Rapuano, and L. Tomaciello. One-Way Delay Measurement: State of the Art. *IEEE Transactions on Instrumentation and Measurement*, 57(12):2742–2750, December 2008. ISSN 0018-9456. doi: 10.1109/TIM.2008.926052. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4542794`. 44, 45

[49] D. Mills, J. Martin, J. Burbank, and W. Kasch. Network Time Protocol Version 4: Protocol and Algorithms Specification. RFC 5905 (Proposed Standard), June 2010. URL `http://www.ietf.org/rfc/rfc5905.txt`. 45, 61

[50] Kang Lee and John Eidson. IEEE-1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. In *In 34 th Annual Precise Time and Time Interval (PTTI) Meeting*, pages 98–105, 2002. 45

[51] O. Gurewitz, I. Cidon, and M. Sidi. One-way delay estimation using network-wide measurements. *IEEE Transactions on Information Theory*, 52(6):2710–2724, June 2006. ISSN 0018-9448. doi: 10.1109/TIT.2006.874414. URL `http://dl.acm.org/citation.cfm?id=1148663.1148692`. 45

[52] Gu Wei-Xuan and Yu Shun-Zheng. Inference of One-Way Queuing Delay Distribution Using Packet-Pair Probes without Clock Synchronization. In *2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007)*, pages 169–175. IEEE, September 2007. ISBN 0-7695-2943-7. doi: 10.1109/NPC.2007.137. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4351480`. 46

[53] B. Ngamwongwattana and R. Thompson. Sync & Sense: VoIP Measurement Methodology for Assessing One-Way Delay Without Clock Synchronization. *IEEE Transactions on Instrumentation and Measurement*, 59(5):1318–1326, May 2010. ISSN 0018-9456. doi: 10.1109/TIM.2010.2043978. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5438761`. 46

[54] A.P. Markopoulou, F.A. Tobagi, and M.J. Karam. Assessment of VoIP quality over Internet backbones. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 150–159 vol.1, 2002. doi: 10.1109/INFCOM.2002.1019256. 47

[55] Wenyu Jiang and Henning Schulzrinne. assessment Of VoIP Service Availability In The Current Internet. In *Proceedings of the 4th International Workshop on Passive and Active Network Measurement (PAM 2003)*, 2003. 48

[56] Mark Sylor, Nagarjuna Venna, and Harrison Ripps. Voice Quality on the Internet in 2005 as Measured by www. TestYourVoIP. com. In *Large Scale Management of Distributed Systems*, pages 112–123. Springer, 2006. 48

[57] Atsushi Kobayashi and Keisuke Ishibashi. Voip measurement architecture using data mediation. In Giorgio Nunzi, Caterina Scoglio, and Xing Li, editors, *IP Operations and Management*, volume 5843 of *Lecture Notes in Computer Science*, pages 15–26. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-04967-5. doi: 10.1007/978-3-642-04968-2_2. URL `http://dx.doi.org/10.1007/978-3-642-04968-2_2`. 48, 49

[58] R. Birke, M. Mellia, M. Petracca, and D. Rossi. Understanding VoIP from Backbone Measurements. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2027–2035, 2007. doi: 10.1109/INFCOM.2007.235. 49, 78

[59] Robert Birke, Marco Mellia, Michele Petracca, and Dario Rossi. Experiences of VoIP traffic monitoring in a commercial ISP. *International Journal of Network Management*, 20(5):339–359, 2010. ISSN 1099-1190. doi: 10.1002/nem.758. URL `http://dx.doi.org/10.1002/nem.758`. 49

[60] France Telecom. Study of the relationship between instantaneous and overall subjective speech quality for time-varying quality speech sequences: influence of a recency effect. Technical report, ITU Study Group 12ÂăContribution D.139, 2000. 50

[61] Adrian E Conway. A passive method for monitoring voice-over-IP call quality with ITU-T objective speech quality measurement methods. In *Communications, 2002. ICC 2002. IEEE International Conference on*, volume 4, pages 2583–2586. IEEE, 2002. 50

[62] M.H. Ahmed. Call admission control in wireless networks: a comprehensive survey. *IEEE Communications Surveys & Tutorials*, 7(1):49–68, 2005. ISSN 1553-877X. doi: 10.1109/COMST.2005.1423334. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1423334`. 51

[63] Lin Cai, Yang Xiao, and X. Shen. VoIP over WLAN: Voice capacity, admission control, QoS, and MAC. *International Journal of Communication Systems*, 19(4): 491–508, May 2006. ISSN 1099-1131. doi: 10.1002/dac.801. URL http://onlinelibrary.wiley.com/doi/10.1002/dac.801/abstract. 51, 90

[64] Majid Ghaderi and Raouf Boutaba. Call admission control in mobile cellular networks: a comprehensive survey. *Wireless Communications and Mobile Computing*, 6(1):69–93, February 2006. ISSN 1530-8669. doi: 10.1002/wcm.246. URL http://doi.wiley.com/10.1002/wcm.246.

[65] M.A. Hoque and F. Afroz. Call admission control: QoS issue for VoIP. In *Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on*, pages 757–761. IEEE, 2007. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4554512. 51

[66] Sofiene Jelassi, Gerardo Rubino, Hugh Melvin, Habib Youssef, and Guy Pujolle. Quality of Experience of VoIP Service: A Survey of Assessment Approaches and Open Issues. *IEEE Communications Surveys & Tutorials*, 14(2): 491–513, January 2012. ISSN 1553-877X. doi: 10.1109/SURV.2011.120811. 00063. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6129367.

[67] Xiuzhong Chen, Chunfeng Wang, Dong Xuan, Zhongcheng Li, Yinghua Min, and Wei Zhao. Survey on QoS management of VoIP. In *2003 International Conference on Computer Networks and Mobile Computing, 2003. ICCNMC 2003.*, pages 69–77. IEEE Comput. Soc, 2003. ISBN 0-7695-2033-2. doi: 10.1109/ICCNMC.2003.1243029. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1243029. 51, 90

[68] David Houck and Gopal Meempat. Call admission control and load balancing for voice over IP. *Performance Evaluation*, 47(4):243–253, March 2002. ISSN 01665316. doi: 10.1016/S0166-5316(01)00066-9. URL http://dx.doi.org/10.1016/S0166-5316(01)00066-9. 51

[69] S. Garg and M. Kappes. Can I add a VoIP call? In *IEEE International Conference on Communications, 2003. ICC '03.*, volume 2, pages 779–783. IEEE, 2003. ISBN 0-7803-7802-4. doi: 10.1109/ICC.2003.1204432. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1204432. 53, 54

[70] Hung-yu Wei, Kyungtae Kim, A. Kashyap, and S. Ganguly. On Admission of VoIP Calls Over Wireless Mesh Network. volume 5, pages 1990–1995. IEEE, June 2006. ISBN 1-4244-0355-3. doi: 10.1109/ICC.2006.255062. URL http://ieeexplore.ieee.org/ielx5/4024074/4024439/04024457.pdf?tp=&arnumber=4024457&isnumber=4024439. 53, 54

[71] P. McGovern, S. Chung, S. Murphy, and L. Murphy. Endpoint Admission Control for VoIPoWLAN. In *13th International Conference of Telecommunications*. Citeseer, 2006. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.75.2206&rep=rep1&type=pdf`. 52, 53

[72] Alex P Silva, Julio F Pimentel, Francisco R P Cavalcanti, and Vicente A Jr Sousa. A New Call Admission Control Algorithm for VoIP in IEEE 802.11 Wireless LANs. 21(3):176–187, 2006. 53, 54

[73] S. Shin and H. Schulzrinne. Call Admission Control in IEEE 802.11 WLANs Using QP-CAT. In *2008 IEEE INFOCOM - The 27th Conference on Computer Communications*, pages 726–734. IEEE, April 2008. ISBN 978-1-4244-2026-1. doi: 10.1109/INFOCOM.2008.123. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4509718`. 51, 53, 54, 90

[74] K. Mase, Y. Toyama, A.A. Bilhaj, and Y. Suda. *QoS management for VoIP networks with edge-to-edge admission control*. IEEE, 2001. ISBN 0-7803-7206-9. doi: 10.1109/GLOCOM.2001.966237. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=966237`. 52, 53

[75] K. Mase and Y. Toyama. *End-to-end measurement based admission control for VoIP networks*. IEEE, 2002. ISBN 0-7803-7400-2. doi: 10.1109/ICC.2002.997039. URL `http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=997039`.

[76] T. Ezaki, H. Kawakami, and K. Asatani. A new VoIP call admission control scheme with use of alternate routing for low call loss probability. In *2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577)*, volume 4, pages 2209–2213 Vol.4. IEEE, 2004. ISBN 0-7803-8533-0. doi: 10.1109/ICC.2004.1312910. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1312910`. 52, 53

[77] G. Bianchi, A. Capone, and C. Petrioli. Throughput analysis of end-to-end measurement-based admission control in IP. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, volume 3, pages 1461–1470. IEEE, 2000. ISBN 0-7803-5880-5. doi: 10.1109/INFCOM.2000.832544. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=832544`. 52, 53

[78] Nicola Baldo, Paolo Dini, and Jaume Nin-Guerrero. User-driven Call Admission Control for VoIP over WLAN with a Neural Network based cognitive engine. In *2010 2nd International Workshop on Cognitive Information Processing*, pages 52–56. IEEE, June 2010. ISBN 978-1-4244-6459-3. doi: 10.1109/

CIP.2010.5604128. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5604128. 53, 54

[79] Kenta Yasukawa, Andrea G. Forte, and Henning Schulzrinne. Distributed Delay Estimation and Call Admission Control in IEEE 802.11 WLANs. In *2007 IEEE International Conference on Network Protocols*, pages 334–335. IEEE, October 2007. ISBN 978-1-4244-1587-8. doi: 10.1109/ICNP.2007.4375868. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4375868. 53, 54

[80] P. Dini, N. Baldo, J. Nin-Guerrero, J. Mangues-Bafalluy, S. Addepalli, and L. L. Dai. Distributed Call Admission Control for VoIP over 802.11 WLANs Based on Channel Load Estimation. In *2010 IEEE International Conference on Communications*, pages 1–6. IEEE, May 2010. ISBN 978-1-4244-6402-9. doi: 10.1109/ICC.2010.5502825. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5502825. 53, 54

[81] S Garg and M Kappes. Admission control for VoIP traffic in IEEE 802.11 networks. In *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, volume 6, pages 3514 – 3518 vol.6, 2003. doi: 10.1109/GLOCOM.2003.1258888. 53

[82] I.D. Chakeres and E.M. Belding-Royer. PAC: perceptive admission control for mobile wireless networks. In *First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks*, pages 18–26. IEEE Comput. Soc, 2004. ISBN 0-7695-2233-5. doi: 10.1109/QSHINE.2004.37. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1366297.

[83] Hongqiang Zhai, Xiang Chen, and Yuguang Fang. A call admission and rate control scheme for multimedia support over IEEE 802.11 wireless LANs. *Wireless Networks*, 12(4):451–463, May 2006. ISSN 1022-0038. doi: 10.1007/s11276-006-6545-y. URL http://dl.acm.org/citation.cfm?id=1147541.1147546.

[84] Yi Qian, Qingyang Hu, and Hsiao-Hwa Chen. A call admission control framework for voice over WLANs, 2006. ISSN 1536-1284.

[85] Hongqiang Zhai, Jianfeng Wang, and Yuguang Fang. Providing statistical QoS guarantee for voice over IP in the IEEE 802.11 wireless LANs. *IEEE Wireless Communications*, 13(1):36–43, February 2006. ISSN 1536-1284. doi: 10.1109/MWC.2006.1593523. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1593523. 53

[86] A. Kashyap, S. Ganguly, S.R. Das, and S. Banerjee. VoIP on wireless meshes: Models, algorithms and evaluation. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2036–2044. IEEE, May 2007. ISBN 1-4244-1047-9. doi: 10.1109/INFCOM.2007.236. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4215818`. 53, 54

[87] Bih-Hwang Lee and Jhih-Ming Chen. VoIP services with embedded CAC over cable modem systems. *IEEE Transactions on Consumer Electronics*, 51(3):815–823, August 2005. ISSN 0098-3063. doi: 10.1109/TCE.2005.1510489. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1510489`. 53, 54

[88] Yakim Mihov, Kiril Kassev, Adriana Kalaydzhieva, and Boris Tsankov. CAC for VoIP Traffic over Wireless Systems with Voice Activity Detection and Background Noise Transmission. In *2009 3rd International Conference on New Technologies, Mobility and Security*, pages 1–5. IEEE, December 2009. ISBN 978-1-4244-4765-7. doi: 10.1109/NTMS.2009.5384733. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5384733`.

[89] Kiril Kassev, Yakim Mihov, Adriana Kalaydzhieva, and Boris Tsankov. A New Paradigm of CAC Dimensioning for VoIP Traffic over Wireless Access Networks. In *2010 Fourth International Conference on Digital Society*, pages 54–59. IEEE, February 2010. ISBN 978-1-4244-5805-9. doi: 10.1109/ICDS.2010.18. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5432822`. 53, 54

[90] Hao Jiang and Constantinos Dovrolis. Passive estimation of TCP round-trip times. *ACM SIGCOMM Computer Communication Review*, 32(3):75–88, July 2002. ISSN 01464833. doi: 10.1145/571697.571725. URL `http://dl.acm.org/citation.cfm?id=571697.571725`. 57

[91] V. Chandrasekhar, J. Andrews, and A. Gatherer. Femtocell Networks: a Survey. *IEEE Communications Magazine*, 46(9):59–67, september 2008. ISSN 0163-6804. doi: 10.1109/MCOM.2008.4623708. 59

[92] Forum Femto. Femtocell Synchronization and Location. (June), 2010. 60

[93] Smallcellforum. Femtocell Synchronization and Location. Technical Report 6295097, 2012. 61, 82

[94] ITU Study Group 12. Study of the relationship between instantaneous and overall subjective speech quality for time-varying quality speech sequences: influence of a recency effect. *Contribution D.139*, 2000. 62, 96

[95] ZL Lu, SJ Williamson, and L Kaufman. Behavioral lifetime of human auditory sensory memory predicted by physiological measures. *SCIENCE-NEW YORK THEN ...*, 1992. URL `http://lobes.osu.edu/Journals/science92.pdf`. 63

[96] Scalable Network Technologies. Qualnet 4.5. Website. `http://www.scalable-networks.com/`. 64

[97] D. Niculescu, S. Ganguly, K. Kim, and R. Izmailov. Performance of VoIP in a 802.11 Wireless Mesh Network. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1 –11, april 2006. doi: 10.1109/INFOCOM.2006.243. 72

[98] Kyungtae Kim, S. Ganguly, R. Izmailov, and Sangjin Hong. On Packet Aggregation Mechanisms for Improving VoIP Quality in Mesh Networks. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 2, pages 891 –895, may 2006. doi: 10.1109/VETECS.2006.1682953.

[99] Kyungsoo Lee, Sangki Yun, Inhye Kang, and Hyogon Kim. Hop-by-Hop Frame Aggregation for VoIP on Multi-Hop Wireless Networks. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 2454 –2459, may 2008. doi: 10.1109/ICC.2008.466. 72

[100] Feng Wang and M. Hamdi. iPIFO: A Network Memory Architecture for QoS Routers. In *High Performance Switching and Routing, 2007. HPSR '07. Workshop on*, pages 1 –5, 30 2007-june 1 2007. doi: 10.1109/HPSR.2007.4281241. 75

[101] NS-3 v.10, 2011. URL `www.nsnam.org`. 76, 100, 113, 114

[102] MadWifi. Multiband Atheros driver for WiFi, 2006. URL `http://www.madwifi-project.org`. 78, 113

[103] FOSS Drivers. ath5k, 2010. URL `http://linuxwireless.org/en/users/Drivers/ath5k`. 78, 113

[104] 3GPP Technical Specification 26.090 version 8.1.0 Release 8. Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Mandatory Speech Codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec; Transcoding functions. Technical report, 2008. URL `www.3gpp.org`. 92

[105] Joseph Vinod and Brett Chapman. *Deploying QoS for Cisco IP and Next Generation Networks*. 2009. 94, 101

[106] R. G. Cole and J. H. Rosenbluth. Voice over IP Performance Monitoring. *SIGCOMM Comput. Commun. Rev.*, 31(2):9–24, April 2001. ISSN 0146-4833. doi: 10.1145/505666.505669. 96, 113

[107] Vikram Chandrasekhar, Jeffrey Andrews, and Alan Gatherer. Femtocell Networks: a Survey. *Communications Magazine, IEEE*, 46(9):59–67, 2008. 107

APPENDIX A

NS-3 CODE

A.1 NS-3 Implementation of the E-Model - ITU-T G.107

The C++ code in Listing A.1 is the actual implementation of the ITU-T E-Model G.107 [31]. The MOS calculation function uses a class termed *VoipMonitorConversationDetails* developed to store statistics about the measured metrics, as well as for the packet loss calculation, and the actual C++ code of this class is presented in Listing A.2.

Listing A.1: ITU-T G.107 - The E-Model - A NS-3 Implementation

```
double
VoipServer::CalculateMos (Ptr<Packet> packet)
{
  // local variables
  double m_loss, m_mos;
  uint32_t m_ssrc;

  if(m_isFirstPacket)
  {
    m_isFirstPacket = false;
    Ptr<Packet> packetCopyForCodecDetection = packet->Copy();
    RtpHeader rtpHeaderForCodecDetection;
    packetCopyForCodecDetection->PeekHeader(
        rtpHeaderForCodecDetection);
```

```
    uint8_t payloadType = rtpHeaderForCodecDetection.
        GetPayloadType();

    switch (payloadType) {
      case 0: // G.711
        Bpl = 4.3;
        Ie = 0;
        break;
      case 18: // G729
        Bpl = 19;
        Ie = 11;
        break;
      case 3: // GSM - AMR
        Bpl = 10;
        Ie = 5;
        break;
      default:
        NS_FATAL_ERROR (this << "VoipServer -- No voice codec
            mathed!!!");
    }
  }

  Ptr<Packet> packetCopyForDje = packet->Copy();
  packet->PeekHeader(m_rtpHeader);

  VoipTag voipTag;
  bool found = false;

  found = packet->FindFirstMatchingByteTag(voipTag);
  if (!found) {
    NS_LOG_INFO ("  Server: VoipTag NOT found; Maybe not a voip
        packet!!!  ");
  }
  else {
    NS_LOG_INFO ("  Server: VoipTag found!");
  }

  Time m_delay;
  Time m_jitter;
  /*
  * check if the packet belongs to an ongoing conversation:
  * if yes, update the corresponding object with current seqNumber
  * if no, create a new entry in m_conversationsMap and update it
      with current seqNumber
  */
```

```
m_ssrc = m_rtpHeader.GetSsrc();
if(m_serverConversationsMap.count(m_ssrc)) // existing call in
    the network
{
  m_loss = m_serverConversationsMap[m_ssrc]->CalculateLoss(
      m_rtpHeader.GetSequenceNumber());
  m_serverConversationsMap[m_ssrc]->DjeRecordRx (
      packetCopyForDje);
  m_delay = m_serverConversationsMap[m_ssrc]->DjeGetLastDelay();
  m_jitter = m_serverConversationsMap[m_ssrc]->DjeGetLastJitter
      ();
}
else // new call in the network
{
  Ptr <VoipMonitorConversationDetails>
      voipServerConversationDetails;
  voipServerConversationDetails =
      m_voipServerConversationDetails.Create <
      VoipMonitorConversationDetails> ();
  m_serverConversationsMap[m_ssrc] =
      voipServerConversationDetails;
  m_serverConversationsMap[m_ssrc]->SetSsrc(m_ssrc);
  m_serverConversationsMap[m_ssrc]->SetCurrentNodeId(GetNode()->
      GetId());
  m_serverConversationsMap[m_ssrc]->SetOriginatingNodeId(voipTag
      .GetOrigNodeId()) ;
  m_loss = m_serverConversationsMap[m_ssrc]->CalculateLoss(
      m_rtpHeader.GetSequenceNumber());
  m_serverConversationsMap[m_ssrc]->DjeRecordRx (
      packetCopyForDje);
  m_delay = m_serverConversationsMap[m_ssrc]->DjeGetLastDelay();
  m_jitter = m_serverConversationsMap[m_ssrc]->DjeGetLastJitter
      ();
}

double burstr = 1;
Ie_eff = Ie + ((95-Ie) *  m_loss / ( ( m_loss / burstr ) + Bpl)
    );

// Id Parameter calculation
double jitter = m_jitter.GetMilliSeconds();
double Drtcp = m_delay.GetMilliSeconds();

double  Dj = 125 + 0.9 *jitter; //0.125ms is codec frame size in
     G7.11
```

```
if (Dj>300)
{
  Dj=300;
}

// Estimation of De
double De = 1.2 * 0.000125; //Encoding delay is +20% of frame
    size

// Da and Dr is set to zero
double Dr = 0;

// Calculation of T
double T = Drtcp + Dj + De + Dr;

// Id Calculation
double TERV = 65 + (6 * exp(-0.3 * pow(T,2))) - 40 * log10( (1+T
     / 10) / (1 + T / 150) );

// Idte
double Re = 80 + 2.5 * (TERV - 14);
double Roe = -1.5 * (-61.18 - 2);
double Xdt = (Roe - Re) / 2;
double Idte = Xdt + sqrt(pow(Xdt,2) + 100);
Idte = (Idte - 1) * (1 - exp(-T));

// Idle
double Tr = 2*Drtcp+Dj+De;
double Rle = 1228.5 * pow((Tr + 1), -0.25);
double Xdl = (94.77 - Rle) / 2;
double Idle = Xdl + sqrt(pow(Xdl,2) + 169);

// Idd
double Ta = Drtcp+Dj+De;
double Idd = 0;
if (Ta > 100)
{
  double X = (log(Ta / 100)) / log(2);
  Idd = 25.0 * (pow((1.0 + pow(X,6.0)), (1.0 /6.0)) - 3.0 * pow
      ((1.0 + pow((X / 3.0), 6.0)), (1.0 / 6.0)) + 2.0);
}
else
{
  Idd = 0;
}
```

```
  double Id = Idte + Idle + Idd;

  // R value calculation
  double R = 93.34 - Id - Ie_eff;

  m_mos = RtoMOS(R);

  // m_serverConversationsMap is used for loss and jitter
      calculation
  // but also for periodically printing the metric stats
  m_serverConversationsMap[m_ssrc]->UpdateMetricAverage("delay",
      Drtcp);
  m_serverConversationsMap[m_ssrc]->UpdateMetricAverage("jitter",
      jitter);
  m_serverConversationsMap[m_ssrc]->UpdateMetricAverage("loss",
      m_loss);
  m_serverConversationsMap[m_ssrc]->UpdateMetricAverage("mos",
      m_mos);

  return m_mos;
}

double
VoipServer::RtoMOS(double R)
{
  double MOS;
  if(R<0) {
    MOS=1;
  }
  else if(R>0 && R<100) {
    MOS = 1 + 0.035 * R + R * (R -60) * (100 - R) * 7 * 0.000001;
  }
  else if(R>100) {
    MOS = 4.5;
  }
  else {
    NS_FATAL_ERROR ("ERROR: Calculated R value not in correct
        range for MOS conversion");
  }
  return MOS;
}
```

Listing A.2: Implementation of the *VoipMonitorConversationDetails* class

```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */

#include "voip-tag.h"
#include "voip-monitor-conversation-details.h"
#include "ns3/log.h"
#include "ns3/nstime.h"
#include "ns3/simulator.h"
#include "ns3/uinteger.h"
#include "ns3/boolean.h"

namespace ns3 {

NS_LOG_COMPONENT_DEFINE ("VoipMonitorConversationDetails");
NS_OBJECT_ENSURE_REGISTERED (VoipMonitorConversationDetails);

VoipMonitorConversationDetails::VoipMonitorConversationDetails ()
{
  m_voipPacketsReceived   = 0;
  m_lossWindowSize        = 50;
  m_lossArrayPosition     = 0;
  m_loss                  = 0;
  m_mos                   = 0;
  m_delay                 = 0;
  m_isFirstPacket         = true;
  m_isFirstAverageLoop    = true;
  m_mosAverageRefreshRate = Seconds (0.5);
  m_currentNodeId                = -1;
  m_originatingNodeId            = -1;
//  m_mosAverage.Update(m_mos);
}

VoipMonitorConversationDetails::~VoipMonitorConversationDetails ()
{
}

TypeId
VoipMonitorConversationDetails::GetTypeId (void)
{
  static TypeId tid = TypeId ("ns3::VoipMonitorConversationDetails
      ")
    .SetParent<Object> ()
    .AddConstructor<VoipMonitorConversationDetails> ()
    ;
```

```
  return tid;
}

void
VoipMonitorConversationDetails::SetSsrc (uint32_t ssrc)
{
  m_ssrc = ssrc;
}

uint32_t
VoipMonitorConversationDetails::GetSsrc () const
{
  return m_ssrc;
}

void
VoipMonitorConversationDetails::SetCurrentNodeId (uint16_t nodeId)
{
  m_currentNodeId = nodeId;
}

void
VoipMonitorConversationDetails::SetOriginatingNodeId (uint16_t
    nodeId)
{
  m_originatingNodeId = nodeId;
}

uint16_t
VoipMonitorConversationDetails::GetCurrentNodeId () const
{
  return m_currentNodeId;
}

uint16_t
VoipMonitorConversationDetails::GetOriginatingNodeId () const
{
  return m_originatingNodeId;
}

double
VoipMonitorConversationDetails::GetLoss(void)
{
  return m_loss;
}
```

```
double
VoipMonitorConversationDetails::CalculateLoss (SequenceNumber16
    seqNumber)
{
  // NS_LOG_INFO (Simulator::Now () << ":Node:MonConvDet:seqNumber"
       << seqNumber << "isfirst:" << m_isFirstPacket);
  if (m_isFirstPacket)
  {
    m_lowestSeqNumber  = seqNumber;
    m_highestSeqNumber = seqNumber;
    NS_LOG_INFO (Simulator::Now () << ":HERE");
    m_lossArray.assign(m_lossWindowSize, m_lowestSeqNumber);
    m_isFirstPacket = false;
    // Simulator::Schedule (m_mosAverageRefreshRate, &
        VoipMonitorConversationDetails::RefreshAverage, this);
  }

  m_lossArray.at(m_lossArrayPosition % m_lossWindowSize) =
      seqNumber;

  if (seqNumber.operator > (m_highestSeqNumber))
    m_highestSeqNumber = seqNumber;

  m_lowestSeqNumber = seqNumber;
  for(uint16_t i=0; i < m_lossWindowSize; i++)
  {
    if (m_lowestSeqNumber.operator > (m_lossArray.at(i)) )
    {
      m_lowestSeqNumber = m_lossArray.at(i);
    }
  }

  ++m_voipPacketsReceived;
  ++m_lossArrayPosition;

  if (m_voipPacketsReceived >= m_lossWindowSize)
  {
    m_loss = double(100) * (double(1) - double(m_lossWindowSize) /
         double(m_highestSeqNumber.operator -(m_lowestSeqNumber.
        operator - (1) )));
    NS_LOG_INFO (Simulator::Now () << ":m_lossWindowSize:" <<
        m_lossWindowSize << ":m_highestSeqNumber:" <<
        m_highestSeqNumber << ":m_lowestSeqNumber:" <<
        m_lowestSeqNumber);
```

```
    NS_LOG_INFO (Simulator::Now() << ":m_loss:" << m_loss << ":
        m_voipPacketsReceived:" << m_voipPacketsReceived << "
        m_lossArrayPosition % m_lossWindowSize" <<
        m_lossArrayPosition % m_lossWindowSize);
  }
  else
  {
    m_loss = 0;
  }

  NS_LOG_INFO (Simulator::Now() << ":" << m_ssrc << ":" <<
      m_lowestSeqNumber << ":" << m_highestSeqNumber << ":" <<
      m_lossArrayPosition);
  return m_loss;
}

void
VoipMonitorConversationDetails::DjeRecordRx (Ptr<Packet> packet)
{
  m_dje.RecordRx(packet);
}

Time
VoipMonitorConversationDetails::DjeGetLastDelay (void)
{
  return m_dje.GetLastDelay();
}

Time
VoipMonitorConversationDetails::DjeGetLastJitter (void)
{
  return m_dje.GetLastJitter ();
}

void
VoipMonitorConversationDetails::UpdateMetricAverage (std::string
    metric, double value)
{
  if     (metric == "delay") m_delayAverageObject.Update(value);
  else if (metric == "jitter") m_jitterAverageObject.Update(value)
      ;
  else if (metric == "loss") m_lossAverageObject.Update(value);
  else if (metric == "mos") m_mosAverageObject.Update(value);
  else NS_FATAL_ERROR ("VoipMonitorConversationDetails::
      UpdateAverage - Invalid metric name!");
```

```
}

double
VoipMonitorConversationDetails::GetMetricStat (std::string metric,
     std::string statType)
{
  if (metric == "delay")
    {
      if     (statType == "avg") return m_delayAverageObject.Avg();
      else if (statType == "min") return m_delayAverageObject.Min
          ();
      else if (statType == "max") return m_delayAverageObject.Max
          ();
      else if (statType == "stddev") return m_delayAverageObject.
          Stddev();
      else if (statType == "count") return m_delayAverageObject.
          Count();
      else NS_FATAL_ERROR ("VoipMonitorConversationDetails::
          GetMetricStat - Invalid metric type!");
    }
  else if (metric == "jitter")
    {
      if     (statType == "avg") return m_jitterAverageObject.Avg()
          ;
      else if (statType == "min") return m_jitterAverageObject.Min
          ();
      else if (statType == "max") return m_jitterAverageObject.Max
          ();
      else if (statType == "stddev") return m_jitterAverageObject.
          Stddev();
      else if (statType == "count") return m_jitterAverageObject.
          Count();
      else NS_FATAL_ERROR ("VoipMonitorConversationDetails::
          GetMetricStat - Invalid metric type!");
    }
  else if (metric == "loss")
    {
      if     (statType == "avg") return m_lossAverageObject.Avg();
      else if (statType == "min") return m_lossAverageObject.Min()
          ;
      else if (statType == "max") return m_lossAverageObject.Max()
          ;
      else if (statType == "stddev") return m_lossAverageObject.
          Stddev();
```

```
      else if (statType == "count") return m_lossAverageObject.
          Count();
      else NS_FATAL_ERROR ("VoipMonitorConversationDetails::
          GetMetricStat - Invalid metric type!");
    }
  else if (metric == "mos")
    {
      if     (statType == "avg") return m_mosAverageObject.Avg();
      else if (statType == "min") return m_mosAverageObject.Min();
      else if (statType == "max") return m_mosAverageObject.Max();
      else if (statType == "stddev") return m_mosAverageObject.
          Stddev();
      else if (statType == "count") return m_mosAverageObject.
          Count();
      else NS_FATAL_ERROR ("VoipMonitorConversationDetails::
          GetMetricStat - Invalid metric type!");
    }
  else NS_FATAL_ERROR ("VoipMonitorConversationDetails::
      UpdateAverage - Invalid metric name!");
}

void
VoipMonitorConversationDetails::ResetMetricAverage (std::string
    metric)
{
  if     (metric == "delay") m_delayAverageObject.Reset();
  else if (metric == "jitter") m_jitterAverageObject.Reset();
  else if (metric == "loss") m_lossAverageObject.Reset();
  else if (metric == "mos") m_mosAverageObject.Reset();
  else NS_FATAL_ERROR ("VoipMonitorConversationDetails::
      ResetMetricAverage - Invalid metric name!");
}

} // Namespace ns3
```

A.2 NS-3 Implementation of the Speech Model - ITU-T P.59

Listing A.3 shows the actual NS-3 C++ code used to generate speech activity events according to the four state Markov chain model, developed by the ITU-T in their P.59 report [6].

Listing A.3: ITU-T P.59 Speech Model - A NS-3 Implementation

```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */

#include <math.h>

// this function returns a random value between 0 and 1 as double
double GetRandVar (void)  {
  return double(std::rand()) / double(RAND_MAX);
}

// Get Single Talk Duration
double GetTst (void)  {
  return double((-0.854)*log(1 - GetRandVar()) + 0.03); // the
      0.03 seconds = 30 milliseconds are there to avoid events
      shorter that the packetization interval
}

// Get Dual Talk Duration
double GetTdt (void)  {
  return double((-0.226)*log(1 - GetRandVar()) + 0.03);
}

// Get Mutual Silence Duration
double GetTms (void)  {
  return double((-0.456)*log(1 - GetRandVar()) + 0.03);
}

// Get next conversation event, based on current event and on p1,
    p2, and p3 probabilities
uint GetNextConversationEvent (uint currentEvent) {
  double p1 = 0.4;
  double p2 = 0.5;
  double p3 = 0.5;
  switch (currentEvent) {
    case 0:
      return (GetRandVar() < p2) ? 1:2 ;
      break;
    case 1:
      return (GetRandVar() < p1) ? 0:3 ;
      break;
    case 2:
      return (GetRandVar() < p1) ? 0:3 ;
      break;
    case 3:
```

```
      return (GetRandVar() < p3) ? 1:2 ;
      break;
  }
  NS_FATAL_ERROR("GetNextConversationEvent: No such conversation
      event!");
}

// Get current event's duration
double GetEventsDuration (uint currentEvent)  {
  switch (currentEvent) {
    case 0:
      return GetTms() ;
      break;
    case 1:
      return GetTst() ;
      break;
    case 2:
      return GetTst() ;
      break;
    case 3:
      return GetTdt() ;
      break;
  }
  NS_FATAL_ERROR("GetEventsDuration: No such conversation event!")
      ;
}

// Writes two lists, one for each conversation node; take
    startTime and endTime as time parameters
// Model based on the P.59 report from ITU-T
void ComputeCallActivity (std::list<double>& callA, std::list<
    double>& callB, double startTime, double endTime, uint16_t
    call_ssrc) {
  startTime += 3;
  double callDuration = endTime - startTime - 1;  // -1 second is
      to avoid scheduling a send event after the socket is closed
  double talkTime = double(0.0);
  uint currentEvent = 0;
  uint previousEvent = 0;
  while (talkTime < callDuration) {
    double currentEventsDuration = GetEventsDuration(currentEvent)
        ;
    if ( currentEvent == 0 )  {
      if ( previousEvent == 1 ) {
        callA.push_back(talkTime + startTime);
```

```
      }
      if ( previousEvent == 2 ) {
        callB.push_back(talkTime + startTime);
      }
    }
    if ( currentEvent == 1 )  {
      if ( previousEvent == 0)  {
        callA.push_back(talkTime + startTime);
      }
      if ( previousEvent == 3)  {
        callB.push_back(talkTime + startTime);
      }
    }
    if ( currentEvent == 2 )  {
      if (previousEvent == 0) {
        callB.push_back(talkTime + startTime);
      }
      if ( previousEvent == 3)  {
        callA.push_back(talkTime + startTime);
      }
    }
    if ( currentEvent == 3 )  {
      if ( previousEvent == 1)  {
        callB.push_back(talkTime + startTime);
      }
      if ( previousEvent == 2 ) {
        callA.push_back(talkTime + startTime);
      }
    }
    talkTime += currentEventsDuration;
    previousEvent = currentEvent;
    currentEvent = GetNextConversationEvent (currentEvent);
  }

  // these two if's are there to make sure each conversation ends
     with a silent period
  if (!(callA.size()%2))  {
    callA.pop_back();
  }
  if (!(callB.size()%2))  {
    callB.pop_back();
  }
}
```

A.3 NS-3 Implementation of the PIFO Queue

Listing A.4 shows the actual NS-3 C++ code used to push VOIP packets in a PIFO queue. The criterion of insertion is the cumulative queueing delay value attached to each VoIP packet.

Listing A.4: Push-In Method in a PIFO Queue - A NS-3 Implementation

```
void WifiMacQueue::PushInQueue (Ptr<const Packet> packet, const
    WifiMacHeader &hdr)
{
  Cleanup ();

  if (m_queue.empty())
  {
    Time timestamp = Simulator::Now();
    m_queue.push_front (Item (packet, hdr, timestamp));
    m_size++;
    return;
  }

  VoipTag currentPacketsVoipTag;
  Ptr <Packet> cpCurrentPacket = packet->Copy();

  {
    bool found = false;
    found = cpCurrentPacket->FindFirstMatchingByteTag(
        currentPacketsVoipTag);
    if (!found)
    {
      NS_LOG_UNCOND(Simulator::Now() << ":WifiMacQueue:PushInQueue
          :currentPacket:No VoipTag. Pushing packet back if there
          is space");
      Time timestamp = Simulator::Now();
      m_queue.push_back (Item (cpCurrentPacket, hdr, timestamp));
      return;
      // NS_LOG_UNCOND ("This packet is not a VoIP packet and
          needs to go last");
    }
    else
    {
      NS_LOG_INFO(Simulator::Now() << ":WifiMacQueue:PushInQueue:
          currentPacket:VoipTag found!");
      DelayJitterEstimationTimestampTag currentByteDjeTag;
```

```
      found = cpCurrentPacket->FindFirstMatchingByteTag (
          currentByteDjeTag);
      if (!found)
      {
        NS_LOG_UNCOND ("WifiMacQueue::PushInQueue
            DelayJitterEstimation bytetag not found!");
      }
      else
      {
      // NS_LOG_UNCOND ("WifiMacQueue::PushInQueue
          DelayJitterEstimation bytetag found");
        QosTag qosTag;
        NS_ASSERT(cpCurrentPacket->FindFirstMatchingByteTag (
            qosTag));
        cpCurrentPacket->RemoveAllByteTags();
        cpCurrentPacket->AddByteTag (currentPacketsVoipTag);
        cpCurrentPacket->AddByteTag (currentByteDjeTag);
        cpCurrentPacket->AddByteTag (qosTag);
      }
    }
  }

  PacketQueueI i;
  uint16_t positionOfInsertion = 1;
  for (i = m_queue.begin (); i != m_queue.end ();)
  {
    VoipTag voipTag;
    bool found = false;
    Ptr <Packet> cp_packet = i->packet->Copy();
    found = cp_packet->FindFirstMatchingByteTag(voipTag);
    if (!found) {
      NS_LOG_INFO(Simulator::Now() << ":WifiMacQueue:SortQueue:
          VoipTag NOT found, Maybe not a voip packet!!!");
      i++;
      positionOfInsertion++;
    }
    else {
      NS_LOG_INFO(Simulator::Now() << ":WifiMacQueue:SortQueue:
          VoipTag found!");

      uint16_t newTimeSpentInQueues = uint16_t((Simulator::Now() -
           i->tstamp).GetMilliSeconds()) + voipTag.
          GetTimeSpentInQueues();
      // NS_LOG_UNCOND ("now - timestamp + voiptag= " << uint16_t
          ((Simulator::Now() - i->tstamp).GetMilliSeconds()) +
```

```
        voipTag.GetTimeSpentInQueues() << " current packets delay
        = " << currentPacketsVoipTag.GetTimeSpentInQueues());

    if (currentPacketsVoipTag.GetTimeSpentInQueues() >
        newTimeSpentInQueues) // if the current packet is older
        than one of the packets in the queue, it will get
        inserted there
    {
      Time timestamp = Simulator::Now();
      i = m_queue.insert(i, Item (cpCurrentPacket, hdr,
          timestamp));
      // NS_LOG_UNCOND (Simulator::Now().GetSeconds() << ":" <<
          m_nodeId << ":" << m_queue.size () << ":VoIP packet
          pushed at position:" << positionOfInsertion);
      m_size++;

      std::ostringstream ss;
      std::string  sss;

      // NS_LOG_UNCOND(Simulator::Now().GetSeconds() << ":
          QueueSizeAfterPush:" << m_queue.size ());
      // if the queue's size got bigger than the m_maxSize, than
           the last packet in the queue needs to be dropped
      UintegerValue maxSize;
      g_wifiQueueMaxSize.GetValue(maxSize);
      m_maxSize = maxSize.Get();
      // NS_LOG_UNCOND("MaxQueueSize is: " << m_maxSize);
      if (m_size > m_maxSize) {
        i = m_queue.end ();
        i--;
        // this->PrintVoipPacketTagContents(i->packet->Copy(),
            ":PUSH:OlderPacket:FullQueue:DroppingYoungestPacket")
            ;
        // this->PrintVoipByteTagContents(packet->Copy(), ":P:
            opqfdyp");

        ss << ":P:opqfdyp:";
        ss << positionOfInsertion;
        sss = ss.str();

        // PIFO_EVENT_FOR_LOG
        // this->PrintVoipByteTagContents(packet->Copy(), sss);
        m_queue.erase (i);
        m_size--;
      }
```

```
          else {
          // this->PrintVoipPacketTagContents(packet->Copy(), ":PUSH
              :OlderPacket");
            // this->PrintVoipByteTagContents(packet->Copy(), ":P:op
                ");
            ss << ":P:op:";
            ss << positionOfInsertion;
            sss = ss.str();

            // PIFO_EVENT_FOR_LOG
            // this->PrintVoipByteTagContents(packet->Copy(), sss);
          }
          return;
        }
        else {
          i++;
          positionOfInsertion++;
        }
      }
    }

    // if the packet is the youngest, we try to insert it last in
        the queue
    UintegerValue maxSize;
    g_wifiQueueMaxSize.GetValue(maxSize);
    m_maxSize = maxSize.Get();
    if (m_size == m_maxSize) // if the queue is full this young
        packet is dropped
    {
      // PIFO_EVENT_FOR_LOG
      // this->PrintVoipByteTagContents(packet->Copy(), ":P:qfdyp");
      NS_LOG_UNCOND ("NEWEST PACKET DROPPED!!!");
    }
    else // if the queue is not full, the packet is enqueued last
    {
      Time timestamp = Simulator::Now();
      i = m_queue.insert(i, Item (cpCurrentPacket, hdr, timestamp));
      m_size++;
      // PIFO_EVENT_FOR_LOG
      // this->PrintVoipByteTagContents(packet->Copy(), ":P:ypel");
    }
  }
```