<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Optimizing the QoS of VoIP applications over WiFi through use of synchronized time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>O Flaithearta, Padraig</td>
</tr>
<tr>
<td><strong>Publication Date</strong></td>
<td>2015-01-31</td>
</tr>
<tr>
<td><strong>Item record</strong></td>
<td><a href="http://hdl.handle.net/10379/5311">http://hdl.handle.net/10379/5311</a></td>
</tr>
</tbody>
</table>

Some rights reserved. For more information, please see the item record link above.
DECLARATION OF ORIGINALITY

I hereby declare that the work contained in this thesis which I submit for assessment leading to the award of PhD is entirely my own work, and that I have exercised reasonable care to ensure that the work is original and does not, to the best of my knowledge, breach any law of copyright or has not been taken from the work of others. Appropriate credit has been given where reference is made to others work.

________________________________________
Padraig Ó Flaithearta

30th January 2015
Abstract

With a rapidly increasing volume of Internet content and connected devices, along with the development of cloud services and server virtualization, it’s becoming clear that traditional network architectures are becoming increasingly ill-equipped to cater for today’s enterprises, and end-users. The Internet has now evolved to a stage where the number of Internet enabled devices exceeds the global population and it is estimated that there will be 50 billion devices connected to the Internet by 2020. This growth is being driven by concepts such as the Internet of Things (IoT) and Cyber Physical Systems (CPS), facilitated by new paradigms such as Software Defined Networking (SDN) and is expected to be characterized by machine-to-machine communication. With such a growth in devices, the need to protect the Quality of Service (QoS) of certain traffic classes such as Real Time Communications (RTC) has never been more necessary. This is particularly so in the wireless domain where bandwidth provision is typically more problematic and where much of this connectivity will be centred.

The ongoing development of communications and industrial applications over computer networks increasingly relies on accurate time synchronization. Precise and verifiable timing is a key requirement that current systems do not support, particularly in wireless networks. The Network Time Protocol (NTP) is widely used to synchronize computer clocks on the Internet where it can provide sub-millisecond accuracies on LANs; however it is typically much less accurate on WLANs due to
their inherent asymmetry. The challenge of improving time synchronization in wireless networks is part of other ongoing research at NUI Galway.

This thesis focuses on evaluating the extent to which synchronized time can be used to improve the QoS of Real Time Communications (RTC) such as Voice over Internet Protocol (VoIP) applications over wireless networks. Although the IEEE 802.11 (WiFi) standard has a QoS extension, it has significant limitations that this thesis addresses. A dual approach of simulation and real-world experimentation is taken. The former is used to validate the core idea whereas the latter is used principally to assess the technical feasibility of the approach. Regarding the latter, an intelligent Access Point (iAP) is presented which integrates a number of key features that operate in real-time. It firstly dynamically calculates accurate one-way delays based on synchronized time. These values (as well as the loss rate) inform a QoS estimator to produce QoS ratings for individual VoIP calls and these values in turn are used by a VoIP traffic prioritization mechanism which prioritizes certain VoIP calls, if required, based on their QoS score. Results show that the approach is thus valid and feasible. The approach is also very much aligned with the emerging SDN paradigm and these linkages are discussed to put the thesis contribution into context. The core research questions addressed by this thesis are thus twofold. Firstly the use of synchronized time in wireless networks is shown to facilitate a significant improvement in QoS management of RTC over WiFi under certain conditions. Secondly, the thesis proves the feasibility of the approach through development of a proof-of-concept (PoC) implementation, based on the WLAN QoS protocol 802.11e.
Acknowledgements

The undertaking of this PhD has been a time of many ups and downs in my academic, social, and personal life. I would not have been able to reach this point without the help of the following people:

My supervisor, Dr. Hugh Melvin has endured my sometimes; alternative, thinking process while always encouraging me to move forward. His ability to see the big picture as well as the smallest detail has provided me with excellent guidance and support, without which I would not have reached this point. He and I also share a love for music and we have played music sessions before and I’m sure we will do so many times again in future.

Tá mé fior buíochas do Gearóid Ó Cadhain, Dr. Seathrún Ó Tuairisg, agus Séamas Ó Concheanainn san Acadamh na hOllscolaíochta Gaeilge leis an cúnamh agus comhairle a thug siad dhom i rith mo chuid taighde.

I would like to thank all of my colleagues in the Discipline of Information Technology at NUI Galway, especially Dr. Michael Schukat, who has provided me with invaluable feedback and assistance throughout my time here, and also Dr. Jonathan Shannon, Frank Glavin, Mark White, and all in NUIG Computer DISC. Thanks also to Ercan Öztürk for his very useful contribution, Hakan Oskok in Ankara, Turkey and to Dr. Peter Pocta in Zilina, Slovakia for all of their help and during my research.

I would like to thank my parents PJ and Grace not only for their unconditional love and support throughout my life, but also for demonstrating to me how to be ambitious and hard-working while maintaining a sense of integrity and grace. My brother Cathal, and sister Aisling have always been there for me and they continue to show me how one can enjoy life doing what they love. The support of Ann, Seamus, Fionnuala, John, and Bridgie O Malley and all of my extended family has also been invaluable to me, as has the constant support of my two friends Adam and Mairtin whether they were in Inis Mór, Canada or Australia, and Ciara has also helped over the last hurdle.

I would like to dedicate this work to three people; firstly to my parents PJ and Grace for being the people they are. And secondly, to the first scientist I ever knew; my cousin Mark Joyce, whose enthusiastic conversations about science and engineering with my older brother Cathal when we were kids opened my eyes to the fascinating possibilities of technology.
TABLE OF CONTENTS

Declaration of Originality ........................................................................................................2
Table of Contents .....................................................................................................................6
List of Figures ..........................................................................................................................11
List of Acronyms .....................................................................................................................14
Publications .............................................................................................................................19

Section I

Chapter 1  Introduction.........................................................................................................23
  1.1  Research Motivation.....................................................................................................25
  1.2  Problem Statement & Research Questions ....................................................................27
  1.3  Solution Approach .......................................................................................................28
  1.4  Contributions ...............................................................................................................29
  1.5  Thesis Outline .............................................................................................................30

Section II

Chapter 2  Background.........................................................................................................33
  2.1  Voice over Internet Protocol (VoIP)...............................................................................34
  2.1.1  VoIP Protocols ........................................................................................................34
        2.1.1.1  VoIP Data Protocols ......................................................................................36
        2.1.1.2  VoIP Control Protocols ................................................................................38
  2.1.2  VoIP Codecs ............................................................................................................44
2.2 Voice Call Quality Performance Evaluation .......................................................... 45
  2.2.1 Quality of Service vs. Quality of Experience ....................................................... 45
    2.2.1.1 QUALINET ......................................................................................................... 46
  2.2.2 Speech Quality Evaluation ..................................................................................... 47
    2.2.2.1 The ITU-T G.107 E-Model ........................................................................... 47
    2.2.2.2 Mean Opinion Score ....................................................................................... 55
  2.2.3 Potential of Synchronized Time to Optimize QoS ..................................................... 57
2.3 WLAN Networks ....................................................................................................... 58
  2.3.1 IEEE 802.11 WLAN Architecture ....................................................................... 58
    2.3.1.1 Wireless Medium Access ............................................................................... 60
    2.3.1.2 WLAN Synchronization .................................................................................. 62
  2.3.2 IEEE 802.11e: Quality of Service Extensions ....................................................... 63
    2.3.2.1 Enhanced Distributed Channel Access (EDCA) .............................................. 63
  2.3.3 WLAN Software Architecture .............................................................................. 69
  2.3.4 802.11aa .............................................................................................................. 71
  2.3.5 VoIP Support over 802.11e .................................................................................... 74
    2.3.5.1 Optimal EDCA Parameter Configuration ....................................................... 75
    2.3.5.2 Game Theoretical Approach towards EDCA Parameter Optimization .............. 78
    2.3.5.3 802.11 VoIP Capacity ..................................................................................... 78
2.4 Time Synchronization ................................................................................................. 79
  2.4.1 Future Timing Requirements ................................................................................. 80
  2.4.2 Network Timing .................................................................................................... 80
    2.4.2.1 Delay Measurement & Synchronization ......................................................... 81
    2.4.2.2 The Network Time Protocol ......................................................................... 84
2.5 Software Defined Networking (SDN) ...................................................................... 87
  2.5.1 OpenFlow ............................................................................................................. 89
  2.5.2 Trends Driving Network Evolution ...................................................................... 89
  2.5.3 SDN and WLAN Networks, QoS/QoE Benefits .................................................. 91
2.6 Linux Traffic Control ............................................................................................... 92
Section III

Chapter 3  System Design ................................................................. 97

3.1  Network Simulation................................................................. 98

3.2  Experimental Test-bed............................................................ 99

3.2.1  Intelligent Access Point – iAP ............................................... 100

3.2.2  Multiple SDN APs ............................................................... 102

3.3  QoS Estimation Methodology ................................................... 104

3.3.1  One-Way Delay over Wi-Fi ................................................... 104

3.3.2  Delay Calculation Mechanism ............................................... 108

3.3.2.1  Time Synchronization Challenges ..................................... 113

3.4  Traffic Prioritization................................................................. 114

3.4.1  Intra-AC Multimedia Categorization ...................................... 114

3.4.2  Traffic Prioritization ............................................................. 118

3.4.3  Traffic Re-Classification ....................................................... 120

3.5  Summary.................................................................................. 120

Chapter 4  Implementation .............................................................. 122

4.1  Simulation Setup....................................................................... 123

4.1.1  Preliminary Simulations - NS-2 ............................................. 123

4.1.1.1  NS-2: IEEE 802.11e VoIP Capacity Evaluation (Simulation Test 1) 123

4.1.1.2  NS-2: Effects of Differing One-Way Delay on VoIP (Simulation Test 2) 124

4.1.1.3  NS-2: Preliminary EDCA Parameter Optimization (Simulation Test 3) 126

4.1.2  iAP Implementation - NS-3 .................................................... 127

4.1.2.1  NS-3: Multimedia Category Evaluation (Simulation Test 4) .......... 127

4.2  Experimental Test-bed Setup .................................................... 129

4.2.1  iAP Setup ............................................................................ 131
4.2.1.1 TCPDump........................................................................................................131

4.2.1.2 Prioritization Implementation ........................................................................132

4.2.2 iAP Operation Routine .......................................................................................133

4.3 Time Synchronization Implementation ...................................................................134

4.3.1 One-Way Delay Calculation without Synchronized AP ....................................136

Section IV

Chapter 5 Simulation Results .......................................................................................140

5.1 Test 1 – NS-2: IEEE 802.11e VoIP Call Capacity .................................................140

5.2 Test 2 – NS-2: The Effects of Differing One-Way Delay on VoIP .........................142

5.3 Test 3 – NS-2: Preliminary EDCA Parameter Optimization ..................................146

5.4 Test 4 – NS-3: Multimedia Category Evaluation ..................................................150

5.4.1 Test 4.1 – NS-3: Default EDCA .........................................................................150

5.4.2 Test 4.2 – NS-3: Multimedia Categorization (MC) Implementation ................156

5.4.3 Wireless MAC delay Improvement ....................................................................161

5.5 Summary ..............................................................................................................163

Chapter 6 Experimental Results ..................................................................................164

6.1 Synchronization Validation ....................................................................................165

6.1.1 Wired client 1 ....................................................................................................165

6.2 Experimental Test-bed Delay & R-factor Validation .............................................166

6.3 Validation of Proof-of-Concept .............................................................................169

6.3.1 MC Prioritization ...............................................................................................169

6.3.1.1 Preliminary Test ............................................................................................169

6.3.1.2 Multimedia Category Prioritization ...............................................................171

6.4 Summary ..............................................................................................................175

Section V

Chapter 7 Conclusions ...............................................................................................177
7.1 Core Research Questions & Contributions .......................................................... 177

7.2 Future Work ........................................................................................................... 181

7.2.1 Traffic Prioritization .......................................................................................... 182

7.2.1.1 EDCA Parameter Optimization ........................................................................ 182

7.2.1.2 Evolution of the VoIP Prioritization Algorithm ................................................ 182

7.2.1.3 QoE Subjective Testing .................................................................................. 182

7.2.2 TAACCS – A new era in QoS/QoE Management .................................................. 183

7.2.3 SDN Development ............................................................................................. 183

7.2.4 Extension to other types of Traffic ..................................................................... 183

References .................................................................................................................... 184

Section VI

Appendix 1 ..................................................................................................................... 193

Appendix 2 ..................................................................................................................... 196

Appendix 3 ..................................................................................................................... 198

Appendix 4 ..................................................................................................................... 201
LIST OF FIGURES

FIGURE 1.1  THE INTERNET OF THINGS (TRADITIONAL IT & NETWORKING DEVICES)[5]  .................................................. 24
FIGURE 2.1  VOIP NETWORK ................................................................................................................................. 34
FIGURE 2.2  OSI MODEL [16] ................................................................................................................................. 35
FIGURE 2.3  DSCP VALUE IN IP HEADER ........................................................................................................... 36
FIGURE 2.4  RTP HEADER ....................................................................................................................................... 37
FIGURE 2.5  RTCP SR/RR PACKET HEADER ....................................................................................................... 39
FIGURE 2.6  SIP SESSION SETUP EXAMPLE ..................................................................................................... 43
FIGURE 2.7  REFERENCE CONNECTION OF THE E-MODEL [12] ..................................................................... 48
FIGURE 2.8  DETERMINATION OF THE EFFECTS OF ABSOLUTE DELAY BY THE E-MODEL [30] ..................... 53
FIGURE 2.9  MOS AS A FUNCTION OF R[12] ......................................................................................................... 56
FIGURE 2.10  INFRASTRUCTURE WLAN ............................................................................................................. 59
FIGURE 2.11  REQUEST - RESPONSE PROCESS .................................................................................................. 60
FIGURE 2.12  DCF CSMA/CA ................................................................................................................................. 61
FIGURE 2.13  EDCA ACs .......................................................................................................................................... 64
FIGURE 2.14  THE WLAN SOFTWARE PACKAGE ARCHITECTURE .................................................................... 71
FIGURE 2.15  802.11AA INTRA-AC TRAFFIC PRIORITIZATION ............................................................................. 73
FIGURE 2.16  UNI-DIRECTIONAL SYNCHRONIZATION ....................................................................................... 82
FIGURE 2.17  ROUND-TRIP SYNCHRONIZATION ............................................................................................... 83
FIGURE 2.18  (A) NTP SUBNET 1  (B) SUBNET 2 ..................................................................................................... 84
FIGURE 2.19  NTP DELAY & OFFSET MEASUREMENT ....................................................................................... 85
FIGURE 2.20  NTP PACKET HEADER .................................................................................................................. 86
FIGURE 2.21  SOFTWARE DEFINED NETWORK ARCHITECTURE .................................................................... 88
FIGURE 2.22  IPTABLES PACKET FLOW ............................................................................................................. 94
FIGURE 3.1  TEST-BED ARCHITECTURE ............................................................................................................. 100
FIGURE 3.2  INTELLIGENT ACCESS POINT ARCHITECTURE ............................................................................. 102
FIGURE 3.3  SDN MULTIPLE AP MESH ARCHITECTURE .................................................................................... 103
FIGURE 3.4  WIRED / WIRELESS NETWORK DELAY (WND/WLND) ............................................................. 104
FIGURE 3.5  DIFFERING ONE-WAY DELAYS .................................................................................................... 105
FIGURE 6.6 WND on Downlink for S1................................................................. 170
FIGURE 6.7 One-way delay for S1 & S2 .......................................................... 171
FIGURE 6.8 R-factors for S1 & S2 (Downlink)............................................. 171
FIGURE 6.9 NetEM WND Delays added to S1 Downlink................................. 172
FIGURE 6.10 Calculated delay for S1 & S2 ...................................................... 173
FIGURE 6.11 Downlink R-factors for S1 & S2 .............................................. 174
FIGURE 6.12 Uplink R-factors for S1 & S2 ..................................................... 174
Netfilter (iptables) Packet Flow Graph: J. Engelhardt, Creative Commons, Linux 2.6.36+, Feb 2014 .... 196
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>Access Category</td>
</tr>
<tr>
<td>AC_BE</td>
<td>Access Category Best Effort</td>
</tr>
<tr>
<td>AC_BK</td>
<td>Access Category Background</td>
</tr>
<tr>
<td>AC_VI</td>
<td>Access Category Video</td>
</tr>
<tr>
<td>AC_VO</td>
<td>Access Category Voice</td>
</tr>
<tr>
<td>ACK</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>AIFS(N)</td>
<td>Arbitrary Inter Frame Space (Number)</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>BSS</td>
<td>Basic Service Set</td>
</tr>
<tr>
<td>CBR</td>
<td>Constant Bit Rate</td>
</tr>
<tr>
<td>CFB</td>
<td>Contention Free Bursting</td>
</tr>
<tr>
<td>CFP</td>
<td>Contention Free Period</td>
</tr>
<tr>
<td>Codec</td>
<td>Code/Decode</td>
</tr>
<tr>
<td>CP</td>
<td>Contention Period</td>
</tr>
<tr>
<td>CPS</td>
<td>Cyber Physical Systems</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access with Collision Avoidance</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>CWmin/max</td>
<td>Contention Window min/max</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>DCF</td>
<td>Distributed Coordination Function</td>
</tr>
<tr>
<td>DIFS</td>
<td>Distributed Inter Frame Space</td>
</tr>
<tr>
<td>DLSR</td>
<td>Delay since Last Sender Report</td>
</tr>
<tr>
<td>DSCP</td>
<td>Differentiated Services Code Point</td>
</tr>
<tr>
<td>EDCA</td>
<td>Enhanced Distributed Channel Access</td>
</tr>
<tr>
<td>EIFS</td>
<td>Extended Inter Frame Space</td>
</tr>
<tr>
<td>ESS</td>
<td>Extended Service Set</td>
</tr>
<tr>
<td>HCCA</td>
<td>HCF Controlled Channel Access</td>
</tr>
<tr>
<td>HCF</td>
<td>Hybrid Coordination Function</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>iAP</td>
<td>Intelligent Access Point</td>
</tr>
<tr>
<td>ICMP</td>
<td>Internet Control Messaging Protocol</td>
</tr>
<tr>
<td>IoE</td>
<td>Internet of Everything</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>iM2E</td>
<td>Intra Mouth to Ear</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IPTV</td>
<td>Internet Protocol TV</td>
</tr>
<tr>
<td>IPv6</td>
<td>Internet Protocol version 6</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial, Scientific, Medical</td>
</tr>
<tr>
<td>ITU-T</td>
<td>International Telecommunications Union–Telecomm’s Standards</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LSW</td>
<td>Least Significant Word</td>
</tr>
<tr>
<td>M2E</td>
<td>Mouth to Ear</td>
</tr>
<tr>
<td>M2M</td>
<td>Machine to Machine</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MC</td>
<td>Multimedia Category</td>
</tr>
<tr>
<td>MMOG</td>
<td>Massively Multiplayer Online Gaming</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>MOS</td>
<td>Mean Opinion Score</td>
</tr>
<tr>
<td>MSW</td>
<td>Most Significant Word</td>
</tr>
<tr>
<td>NAV</td>
<td>Network Allocation Vector</td>
</tr>
<tr>
<td>NIC</td>
<td>Network Interface Card</td>
</tr>
<tr>
<td>NTP</td>
<td>Network Time Protocol</td>
</tr>
<tr>
<td>NS-2/3</td>
<td>Network Simulator-2/3</td>
</tr>
<tr>
<td>ONF</td>
<td>Open Network Foundation</td>
</tr>
<tr>
<td>PBX</td>
<td>Private Branch Exchange</td>
</tr>
<tr>
<td>PCM</td>
<td>Pulse Code Modulation</td>
</tr>
<tr>
<td>PEL</td>
<td>Performance Engineering Laboratory</td>
</tr>
<tr>
<td>PESQ</td>
<td>Perceptual Evaluation of Speech Quality</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical Layer</td>
</tr>
<tr>
<td>PLC</td>
<td>Packet Loss Concealment</td>
</tr>
<tr>
<td>POLQA</td>
<td>Perceptual Objective Listening Quality Assessment</td>
</tr>
<tr>
<td>PSQM</td>
<td>Perceptual Speech Quality Measure</td>
</tr>
<tr>
<td>PSTN</td>
<td>Packet Switched Telephone Network</td>
</tr>
<tr>
<td>PTP</td>
<td>Precision Time Protocol</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Experience</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RTS</td>
<td>Request To Send</td>
</tr>
<tr>
<td>RR</td>
<td>Receiver Report</td>
</tr>
<tr>
<td>RTC</td>
<td>Real Time Communications</td>
</tr>
<tr>
<td>RTCP</td>
<td>Real Time Control Protocol</td>
</tr>
<tr>
<td>RTP</td>
<td>Real-time Transport Protocol</td>
</tr>
<tr>
<td>RTT</td>
<td>Round Trip Time</td>
</tr>
<tr>
<td>R-factor</td>
<td>Transmission Rating Factor - R</td>
</tr>
<tr>
<td>SDN</td>
<td>Software Defined Networks</td>
</tr>
<tr>
<td>SIFS</td>
<td>Short Inter Frame Space</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SIP</td>
<td>Session Initiation Protocol</td>
</tr>
<tr>
<td>SR</td>
<td>Sender Report</td>
</tr>
<tr>
<td>STA</td>
<td>Station</td>
</tr>
<tr>
<td>TAACCS</td>
<td>Time Aware Applications, Computer and Communications Systems</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TOS</td>
<td>Type of Service</td>
</tr>
<tr>
<td>TXOP</td>
<td>Transmission Opportunity</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UP</td>
<td>User Priority</td>
</tr>
<tr>
<td>UTC</td>
<td>Coordinated Universal Time</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over Internet Protocol</td>
</tr>
<tr>
<td>WB</td>
<td>Wideband</td>
</tr>
<tr>
<td>WiFi</td>
<td>Wireless Fidelity</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WLND</td>
<td>Wireless Network Delay</td>
</tr>
<tr>
<td>WMM</td>
<td>Wireless Multimedia</td>
</tr>
<tr>
<td>WND</td>
<td>Wired Network Delay</td>
</tr>
</tbody>
</table>
Publications

International Journals


International Conferences

[1] Padraig O Flaithearta, Hugh Melvin, Peter Pocta. *Time Synchronization for QoE in SDN Networks*, ETSI TC STQ Workshop on Telecommunications Quality beyond 2015, Vienna, Austria, October 2015 (Accepted)


Other Publications


Section I

INTRODUCTION
Chapter 1

INTRODUCTION

The Internet has evolved in recent years to encompass concepts such as the Internet of Everything (IoE) which will bring together people, process, data and will, according to some sources, create economic value of $14.4 trillion between now and 2022 [1]. It is estimated that there will be 25 billion devices connected to the Internet in 2015 (with 50 billion by 2020) [2]. The next phase of Internet growth will be driven by related concepts such as IoE, Cyber Physical Systems (CPS), Industrial Internet (IIC) and a key component will be characterized by machine-to-machine communication. Another enabling technology for such growth is Software Defined Networking (SDN), defined as an emerging network architecture where network control is decoupled from forwarding and is directly programmable which enables underlying infrastructure to be abstracted for applications and network services, which can treat the network as a logical or virtual entity [3]. Within this context, precise and verifiable timing is a key requirement - however current systems do not support this functionality. This broad question is being addressed by the Time Aware Applications, Computer, and Communications Systems (TAACCS) group set up in 2014 [4], which aims to address accurate timing opportunities and needs via a cross-disciplinary research approach. In addition to traditional Internet devices, Figure 1.1 illustrates a summary of the cross-sector scale that the IoT will encompass [5].
With such a growth in devices and traffic, the need to protect the Quality of Service (QoS) of certain traffic classes such as Real Time Communications (RTC) has never been more necessary. This is particularly so in the wireless domain where bandwidth provision is typically more problematic and where much of this connectivity will be centred. Aligned with this, the ongoing development of communications and industrial applications over computer networks increasingly relies on accurate time synchronization. The Network Time Protocol (NTP) is widely used to synchronize computer clocks on the Internet [6] and synchronizes computers to Coordinated Universal Time (UTC) via Internet time servers. It can provide sub-millisecond accuracies on LANs although it is typically much less accurate on WLANs due to their inherent asymmetry. The challenge of improving time synchronization over legacy wireless networks is part of other ongoing research [7] at NUI Galway. The IEEE have also done significant standards work in this area, although the amount of legacy equipment installed will make transition a slow process. Experimental results on legacy wireless networks from this research reduce the error in time related data by up to 90% delivering results similar to that achievable over wired networks.

Figure 1.1 The Internet of Things (Traditional IT & Networking devices)[5]
In terms of QoS, Real-Time Communications (RTC) applications such as Voice over Internet Protocol (VoIP), Video Conferencing, Internet Protocol Television (IPTV), Hybrid Broadband-Broadcast TV (HbbTV), and Multiplayer Online Gaming (MOG) can benefit from accurate time synchronization. The IEEE 802.11e (WMM or Wireless Multi-Media) protocol [8], was developed as an amendment to the 802.11 WiFi standard in 2005 to help address Quality of Service (QoS) concerns for time sensitive data. With modifications at the Media Access Control (MAC) layer, 802.11e provides a set of enhancements for wireless LAN applications which implement a degree of traffic categorization that can improve QoS. 802.11e provides a default set of static parameters which are configured statically. The leading subjective measurement of voice quality is the Mean Opinion Score (MOS) as defined in the ITU (International Telecommunications Union) recommendation P.800 [9]. However, it can be time consuming and expensive to have people sit down and listen to, and evaluate the quality of phone calls. As a result, a number of objective methods have been developed, which include Perceptual Evaluation of Speech Quality (PESQ) [10], Perceptual Objective Listening Quality Assessment (POLQA) [11], and the E-Model (ITU-T G.107) [12], all of which can be mapped to an MOS score.

1.1 Research Motivation

In what can be viewed as a move in the direction of SDN, the research in this thesis evaluates the extent to which synchronized time deployed over wireless networks can aid in improving the QoS of multimedia applications, specifically voice. Voice telecommunications have been gradually migrating from traditional connection orientated systems to the packet based Internet Protocol since the late 1990’s, which has brought about huge savings for the industry and facilitated the development of much more sophisticated communications systems. In order to address resulting QoS issues, arising from the lack of admission control, Diffserv was developed to provide data categorization, and improve QoS for multimedia over private wired networks. However the IEEE 802.11 MAC protocol did not provide any similar QoS facility for wireless networks, which increasingly represent the last hop. The IEEE 802.11e protocol was then developed by the IEEE working group as an amendment to 802.11. The 802.11e standard defines a coordination function called the Hybrid Coordination Function (HCF). HCF includes both a contention-based channel access method,
called the Enhanced Distributed Channel Access (EDCA) mechanism, for contention-based data transmission, and a controlled channel access, referred to as the HCF Controlled Channel Access (HCCA) mechanism, for contention-free data transmission. This amendment which is described in detail in section 2.3.2 provides four traffic access categories (ACs) whereby data traffic is categorized into voice, video, best effort or background queues. 802.11e was designed to improve the QoS of delay-sensitive traffic when operating in contention with other types of data traffic.

We are now at a point in time where QoS-enabled WiFi networks are ubiquitous. Whether using 802.11a/b/g/n, or the recently ratified 802.11ac [13] physical layer, the current standard commercially available QoS-based 802.11e routers on the market can improve real-time traffic performance by assigning priority over best-effort, and background traffic. This reinforces users increasing QoS/QoE expectations of applications to the point that users are beginning to expect the same level of quality over wireless networks as they do over wired networks [14]. As multimedia technologies evolve and become more integrated into everyday life, the level of acceptable QoS needs continuous improvement in line with users’ tolerance and expectations.

The issue of precise time synchronization in WLAN networks has been researched at NUI Galway in parallel with the work in this thesis. When synchronized time is implemented across all nodes in a wireless network, this thesis shows that it is possible to implement a novel delay calculation technique using the Real Time Control Protocol (RTCP) Sender Reports (SR) and Receiver Reports (RR) which can then inform a QoS module producing QoS ratings for individual VoIP sessions based on these delays and in some cases packet loss rate. A further aspect of this is that the delay calculation and subsequent prioritization are done dynamically and in real-time by a central coordinating device, which is very much aligned with the SDN concept.

The prioritization mechanism developed utilises precise delay information and operates at the wireless MAC layer whereby 802.11e QoS implementation doesn’t go far enough with respect to competing delay-sensitive traffic of the same type. In the current 802.11e, all traffic within the 802.11e EDCA voice Access Category (AC) will have similar delays at the wireless MAC layer as it is all treated with an equal level of priority. The literature provides extensive work on the pursuit of optimal parameter configuration using both static and dynamic configurations, some of which use game theoretical selection methods. By contrast, my PhD provides a hybrid static-dynamic 802.11e parameter selection approach which is better suited to a multiple VoIP call scenario. In particular, it proposes that if the precise Mouth-to-Ear (M2E)
one-way delay information for each VoIP session is known in both directions, EDCA parameters can be configured differently between VoIP sessions to optimise the QoS, so that the sessions with the higher one-way M2E network delays receive higher priority treatment at the wireless MAC level relative to other VoIP sessions with lower delays. Essentially, this implements informed prioritization within the voice access category. For a session that has a high E-Model R-factor (or a small one-way delay), packets can afford to wait for a longer time at the wireless MAC layer, on the condition that they are delivered within a period that will not degrade their QoS significantly (150ms according to ITU-T G.114 recommendation), whereas a VoIP session that has a long M2E network delay will benefit from a shorter contention delay at the wireless MAC layer.

1.2 Problem Statement & Research Questions

While the IEEE 802.11 standard proves sufficient for web traffic users, there is an increasing demand for delay sensitive applications among both personal and business users. While the 802.11e QoS amendment goes some way towards servicing this demand, there still exists a limitation in 802.11e in that it does not provide any prioritization amongst traffic that lies within the same AC. When this problem is aligned with improvements in the precision of time synchronization in wireless networks, there is an opportunity to exploit a new one-way delay calculation technique for VoIP that can then be used to build on 802.11e. This thesis seeks to address the following two core questions:

1. Can the implementation of time synchronization over WiFi improve the QoS of VoIP sessions that have relatively high one-way M2E delays?

2. What are the key engineering challenges required to prioritize certain VoIP calls over others using the 802.11e EDCA traffic access mechanism in a real-world RTC environment? This challenge can be subdivided as follows:
a. Is it possible to determine in real-time, the accurate one-way M2E delays for multiple RTC streams?

b. Is it possible to translate dynamic network conditions into QoS scores based on delay/loss for RTC streams in real-time?

c. How feasible is it to manage/configure multiple RTC streams via a single management device, by building on the EDCA Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) mechanism in real-time?

1.3 Solution Approach

A dual-approach methodology is employed in this research whereby network simulations are initially carried out to assess the viability of the proposal and then to inform the development of a real-world experimental test-bed. Simulations are used to investigate the thesis concept and evaluate it particularly over large scale network scenarios that would require extensive resources. However, a scalable, proof-of-concept experimental test-bed is also developed which firstly demonstrates that simulation results achieved are possible in the real-world, and more so that the engineering challenges can be met. The delay calculation mechanism showcases how the move from simulation to the experimental test-bed raises significant engineering challenges that must be overcome. This technique is tested and verified on the proof-of-concept test-bed. Once calculated, delay values are used by a real-time QoS estimator that is based on the standard QoS measurement tool, the E-Model, to provide quality scores for each VoIP call. These quality scores (R-factors) are then used by a prioritization algorithm which alters (mangles) the IP DSCP QoS field in real-time to improve the QoS of relatively poorly performing VoIP calls. While packet loss rate is factored into the R-factor calculations in simulations, it is not used as a parameter in the experimental test-bed as focus was on the one-way delay.

By combining these features in a single device, an intelligent Access Point (iAP) based on SDN concepts is proposed as the core component of the proof-of-concept test-bed. In summary, the prioritization idea is evaluated primarily via simulation (factoring in delay and loss) while the practical feasibility of the approach is proven
via a combination of the delay calculation mechanism (based on synchronized time using RTCP SR/RR packets), the QoS E-Model estimator (factoring in just delay), and Prioritization module within the test-bed.

1.4 Contributions

In addressing the question of whether the implementation of synchronized time in WLAN networks can significantly aid in improving the QoS of real-time multimedia, the concept of an intelligent Access Point (iAP) that oversees all traffic, evaluates QoS of all individual RTC streams, and intervenes as necessary in real-time emerges. The iAP is composed of modules that collectively implement the following:

- Development of a VoIP delay calculation module based on RTCP SR/RR packets and AP packet timestamps. This mechanism requires that endpoints and the AP are time synchronized for successful operation and takes place on an 802.11 AP.
- Using this technique, intra-Mouth-to-Ear (iM2E) delay can be calculated along with Mouth-to-Ear (M2E) delay and Round Trip Delay (RTD).
- An E-Model based QoS Estimator module is proposed. R-factor scores are calculated using the aforementioned delay calculation mechanism.
  - Both delay and loss are factored into QoS estimations in simulations, however, the QoS Estimator module in the experimental test-bed just factors in delay.
- A Traffic-Reclassification module implements the Multimedia Categorization (MC) hierarchical system which facilitates the prioritization algorithm. This carries out the actual prioritization in real-time by mangling the DSCP value in IP headers which is then mapped to the appropriate MC category. This allows VoIP sessions to move up and down between three tiers of prioritization categories.

Finally, a synchronization test is designed and implemented to verify that nodes in the experimental test-bed are synchronized. This test is run before all experiments, and it provides clarity on the precision of synchronization of multiple wired and wireless nodes. At a more conceptual level, the concept of SDN management in the context of the thesis contribution is also considered. Considering that configuration
and management of a wireless BSS via an intelligent AP is shown to be possible from a remote machine, where network-wide delay calculations and prioritization decisions can be carried out remotely, this aligns well with the core SDN concept.

1.5 Thesis Outline

In the first section (Section I), the current chapter has served as an introduction and provided a context for the remaining work contained in this thesis. It has briefly introduced relevant concepts such as the Internet of Things (IoT) and Software Defined Networking (SDN) as well as introducing the research group on Time Aware Applications, Computer and Communications Systems (TAACCS). Research questions were identified and the methodology employed towards conducting the research was provided also along with a summary of the contributions made.

In Section II, Chapter 2 provides the reader with relevant background information on a range of research areas relating to this thesis. VoIP protocols and codecs are discussed along with voice call quality estimation techniques. Technical details on IEEE 802.11 wireless networks are provided including a review of the literature relating to RTC support over WLAN networks. Time synchronization is also discussed with a summary of future timing requirements and some technical details on network timing. Background information is provided on Software Defined Networks (SDN) and information on the SDN interface OpenFlow. Finally, some background information is provided on the Linux Traffic Control suite, part of which is used to implement prioritization of RTC streams in the experimental test-bed.

In Section III (Design & Implementation), there are two chapters, the first of which, Chapter 3, commences with an overview of the rationale for, and design of, the simulation phase of the thesis. It then moves on to test-bed design issues, namely details of the iAP system design. The experimental test-bed design is discussed here along with details on the two core aspects of the functionality of the iAP; the Delay and QoS Estimation Methodology, and Traffic Prioritization. Section III, Chapter 4 then details the implementation of both the simulation test setup and of the experimental test-bed. Simulation tests are introduced here which evaluate different aspects of VoIP over wireless networks.

Section IV of the thesis presents and analyses results for the simulations and experiments described in Chapter 4. The first chapter in Section IV (Chapter 5) provides the results for preliminary simulations carried out on NS-2, as well as more
detailed simulation results, using NS-3 that tested the core thesis research question. Chapter 6 provides experimental results for the iAP that validate the proof-of-concept whereby a VoIP session is prioritized in real-time over another based on its R-factor score which is calculated via delay values provided by the RTCP based delay calculation algorithm.

Section V and Chapter 7 conclude the thesis by summarizing the core contributions of this thesis as well as describing some potential future work that will help build on the initial idea and its experimental proof-of-concept.
Chapter 2

LITERATURE REVIEW

This chapter provides background information for the reader on different aspects of the areas involved in this research, and sets the context for the challenges that this research addresses. The primary background areas in question are Voice over Internet Protocol (VoIP) communications including voice protocols and codecs, as well as information on voice call quality estimation which details models for QoS and QoE (Quality of Experience) estimation. Wireless LANs are also detailed including information on the 802.11 WiFi protocol and its QoS extension 802.11e along with a review of how voice communications behave on WLAN networks. This chapter also covers background information on computer clock and network time synchronization using the Network Time Protocol (NTP) where future timing requirements are discussed as is the Time Aware Applications, Computer and Communications Systems (TAACCS) research project. The Software Defined Networking (SDN) concept is also discussed in this chapter, as is background information on the Linux Traffic Control suite which provides the tools used in the traffic prioritization module as detailed in Chapter 3 and Chapter 4.
2.1 Voice over Internet Protocol (VoIP)

Voice over Internet Protocol (VoIP) refers to the transmission of voice communications over packet switched networks. This technology has succeeded older technologies that utilized connection-oriented analogue transmission such as Public Switched Telephone Networks (PSTN) that use circuit switching. VoIP involves capturing a human voice signal with a microphone on a computer device such as a Personal Computer (PC), laptop, dedicated VoIP phone handset, or mobile phone or smartphone. Once an analogue speech signal is captured it is then encoded at the application layer of the protocol stack by a codec (coder/decoder) into a digital signal. This can then be broken up and packaged in IP packets to be sent over packet switched computer networks to a receiver where the process is then reversed, and the digital signal is decoded and played out to the receiver through a speaker in a VoIP device. Circuit switching involves reserving an entire communication path for the duration of a session whereas VoIP packets might use different paths to get to their ultimate destination (receiver).

2.1.1 VoIP Protocols

VoIP protocols can be categorized into two sub categories; data and control protocols. Data protocols such as Real Time Protocol (RTP) and User Datagram Protocol (UDP) transport the actual voice data, whereas control protocols such as Session Initiation Protocol (SIP) and Real Time Control Protocol (RTCP) set up and maintain VoIP sessions, and handle call quality information.

Figure 2.1 VoIP Network
The Internet Protocol (IP) [15] is used to deliver data packets between host computers based on an IP address that is contained in the IP header. IP is a connectionless protocol where packets can traverse varying paths to reach their destination. IP doesn’t provide guarantees concerning reliability, flow control, error detection or error correction, therefore packets can arrive at the destination in the wrong sequence, with errors, or may be dropped along the way. Some of these characteristics of IP do not make it well suited for voice transmission however these issues are addressed by higher level protocols.

<table>
<thead>
<tr>
<th>Application Layer</th>
<th>HTTP</th>
<th>SIP</th>
<th>NTP</th>
<th>RTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presentation Layer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Session Layer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transport Layer</td>
<td>TCP</td>
<td></td>
<td>UDP</td>
<td></td>
</tr>
<tr>
<td>Network Layer</td>
<td>IP</td>
<td></td>
<td>ICMP</td>
<td></td>
</tr>
<tr>
<td>Data Link Layer(LLC/MAC)</td>
<td>Ethernet</td>
<td></td>
<td></td>
<td>802.11 WLAN</td>
</tr>
<tr>
<td>Physical Layer (PHY)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.2 OSI Model [16]

Within the IP header, the Type of Service (TOS) field provides an indication of the quality of service desired (Traffic Class in IPv6). These parameters are to be used to guide the selection of the actual service parameters when transmitting a datagram through a network, however in reality, these parameters are ignored unless they are being used in a diffserv network. Diffserv came along after TOS but it is backwards compatible. The IP RFC [15] defines traffic service as a three-way trade-off between low-delay, high-reliability, and high-throughput traffic.
The DSCP value in the IP header is used in the test-bed prioritization mechanism to assign packets to appropriate priority queues.

2.1.1.1 **VoIP Data Protocols**

The Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are the two common protocols used to transport information over an IP network. TCP is a connection-oriented protocol that enables reliable communication between two network clients. TCP allows hosts to establish a connection for exchange of data streams and it provides guarantees that packets will be delivered in the order that they were sent. TCP was designed to dynamically adapt to changing network conditions, and to resend data that is lost or dropped in the network. It is typically used in cases where data integrity is important such as web browsing, email and file transfer. TCP breaks application data into segments that it encapsulates within a header. The TCP header contains 10 mandatory fields including source and destination ports, sequence number field and an acknowledgement field which contains the next sequence number that the receiver is expecting.

UDP is a connectionless, unreliable protocol with minimal overhead. UDP datagrams can be sent at any time without prior handshaking or negotiation. UDP routes data to a destination port within an endpoint but does not provide any sequencing or ensure data integrity. It is preferred as a protocol for real-time applications that are more sensitive to delayed packets than dropped packets such as VoIP. Applications can implement mechanisms to tackle dropped packets such as Packet Loss Concealment (PLC) algorithms. Real-time applications require mechanisms to ensure that a stream of data can be reconstructed accurately and in the correct order. The Real-
time Transport Protocol (RTP) and its associated Real-time Transport Control Protocol (RTCP) operate on top of UDP to provide these services.

RTP [17] is the main transport protocol used for IP Telephony media streams and defines a standardized packet format for delivering media over the Internet. RTP provides end-to-end network transport functionality to applications transmitting real-time data. The network services include payload type identification and sequence numbering.

RTP does not have a standard UDP port on which it communicates. It maintains a session for each media stream where RTP packets use an even port while RTCP uses the next higher odd port. RTP carries voice or video data while call setup and teardown is generally performed by call signalling protocols such as SIP.

The fields in the RTP header are described as follows:

- **Version** – Identifies the version of RTP
- **Padding** – If the padding bit is set, the packet contains one or more additional padding octets at the end which are not part of the payload
- **Extension** – If set, the fixed header must be followed by one header extension
- **CSRC Count** – Contains the number of CSRC identifiers that follow the fixed header
- **Marker** – Interpretation to be defined by a profile
- **Payload Type** – Identifies the format of the RTP payload and determines its interpretation by the application. A receiver must discard packets with a Payload Type that it doesn’t understand
- **Sequence Number** – This number increments by one each time an RTP packet is sent, it can be used by a receiver to detect packet loss or to re-
order packets. The initial value should be set to a random number to combat hacking

- *Timestamp* – Reflects the sampling instant of the first octet in the RTP packet. The initial value must be random. Timestamps from different media streams may advance at different rates and usually have independent varying offsets. Therefore these timestamps are sufficient to reconstruct the timing of a single stream, directly comparing RTP timestamps from different media is not effective for synchronization

- *SSRC* – Identifies the Synchronization Source, this should be chosen randomly

- *CSRC* – Identifies the Contributing Sources for the payload contained in this packet. The number of identifiers is contained in the CC field

### 2.1.1.2 VoIP Control Protocols

#### 2.1.1.2.1 Real-time Transport Control Protocol – RTCP

RTCP provides periodic transmission of control packets to all participants in a session, and uses the same distribution mechanism as the data packets (RTP). The primary function of RTCP is to provide feedback on Quality of Service. This feedback is provided by Sender and Receiver Reports (SR/RR). All participants must send RTCP packets at a controlled rate in order for RTP to scale up to a large number of participants. The RTCP specification [17] defines several packet types to carry control information:

- SR – Sender Reports for transmission and reception statistics from active senders

- RR – Receiver Reports for reception statistics from participants that are not active senders and along with SR, for active senders

- SDES – Source description items including CNAME

- BYE – Indicates end of participation

- APP – Application specific functions
Sender reports (SR) consist of three sections, the header (8 octets long), the sender information section (20 octets long), and the report block (RR). The values in the SR packet are as follows:

- **Padding** – If this value is set, it indicates that the packet contains additional padding at the end of the RTCP packet that is not control information, but is included in the length field

- **Reception Report Count (RC)** - The number of reception report blocks that are contained in this packet

- **Packet Type** – Value of 200 indicates SR, 201 indicates RR (Figure 2.5)

- **Length** – The length of this packet including header and padding

- **SSRC** – The Synchronization source identifier for where this SR packet originates
- NTP Timestamp (Most/Least Significant Word) – This indicates the wall-clock time for when this packet was generated\(^1\)

- RTP Timestamp – This is taken at the same sampling instant as the NTP timestamp above, but it’s in the same units as the RTP timestamp in RTP data packets.

- Senders Packet Count – Total number of RTP data packets that have been sent by this sender since the beginning of transmission

- Senders Octet Count – the total number of payload octets sent by this sender since the beginning of transmission

- SSRC_1 – this is the SSRC identifier of the source to where the information in this report block originates

- Fractions Lost – The fraction of RTP data packets lost from source SSRC_1 since the previous SR packet was sent. Calculated as the number of packets lost divided by the number of packets expected as described in Appendix A.3 in [17]

- Cumulative Number Of Packets Lost – This is the total number of packets lost from Source SSRC_1 since the beginning of reception, defined as the sum of packets expected minus the number of packet received

- Extended Highest Sequence Number – (32 bits) The low 16 bits contain the highest sequence number received in an RTP data packet, the high 16 bits extend that sequence number with the corresponding count of sequence number cycles as described in Appendix A.1 in [17]

- Interarrival Jitter – The estimate of the statistical variance of the RTP data packet arrival times as described in [17]

- Last SR Timestamp – The middle 32 bits of the NTP timestamp received in the last SR packet received

- Delay Since Last SR – The delay since receiving the last SR packet and the sending of this current report block. This value is expressed in units of 1/65536 seconds\(^2\)

\(^1\) Used in delay calculation algorithm for time when packet leaves endpoint

- - 40 -
The RTCP transmission interval decides how often SR and RR packets are sent between session endpoints and its calculation is described in RFC 3550 [17]. The interval is calculated based on allocated session bandwidth, the mean size of all RTCP packets, the number of participants in a session and the role that they hold (Sender/Receiver). This interval allows applications to provide fast response for sessions with few participants where, for example, identification of all participants is important, while at the same time it must be able to automatically adapt to large sessions. The recommended default transmission interval is 5 seconds.

This research utilizes the values contained in the NTP timestamp and the DLSR fields along with AP timestamps in order to calculate one-way delays in the experimental test-bed. This calculation mechanism is described in Chapter 3.

2.1.1.2.2 Call Signalling – Session Initiation Protocol (SIP)

When a user wants to connect a voice call to another user it must use some form of call signalling such as SIP or H323 which are peer-to-peer control signalling protocols. The Session Initiation Protocol (SIP) operates on top of many transport protocols at the application layer and is used for creating, modifying and terminating (tear-down) of VoIP, multimedia distribution, and multimedia conference sessions with one or more participants [18].

SIP enables Internet endpoints (senders) to send invite messages (containing session descriptions) to other endpoints (receivers) to partake in VoIP sessions. Associated participants can use this information to agree on a set of compatible media types [18]. The routing of requests is aided by the use of ‘network hosts’ called proxy servers which also aid in authentication, authorization, call routing policy and the provision of features to users. SIP defines five aspects of session establishment/termination:

- User Location – Determines the endpoint that will be used in session
- User Availability/Presence
- User Capabilities – Media and parameters that a user can use
- Session Setup - Establishment of session parameters at all partaking participants

2 Used in delay calculation algorithm to deduce time previous packet was received.
- Session Management – Modification of sessions during sessions, session termination

RFC 3261 [18] recommends that SIP will be used in conjunction with other IETF protocols such as RTP two way communications (VoIP), Real-Time Streaming Protocol (RTSP) for streaming media, Media Gateway Control Protocol (MEGACO) for controlling gateways to the PSTN, and the Session Description Protocol (SDP) for multimedia session description. Although SIP is used along with these protocols, it doesn’t rely on them to provide its own basic functionality. SIP users are defined and categorised as two types of User Agents (UA), User Agent Clients (UAC) which generate requests and send them to servers, and User Agent Servers (UAS) which handle requests, and generate responses. A single UA may function as both.

Figure 2.6 illustrates an example of a message exchange between User 1 and User 2 that uses Proxy Servers $P_{Server1}$ and $P_{Server2}$ to help set up the session. The ‘SIP Trapezoid’ is depicted by the red dotted lines between the users and servers. User 1 sends an initial invite (F1) to User 2 via $P_{Server1}$ as it doesn’t know the location of User 2. $P_{Server1}$ proceeds to forward the invite message to $P_{Server2}$ (it knows to seek $P_{Server2}$ due to the Registration process) while also sending a “Trying” response to User 1. Upon receiving the invite (F2) from $P_{Server1}$, $P_{Server2}$ performs a similar operation by forwarding the invite to User 2 (F4) as it knows the location of User 2. If $P_{Server2}$ did not know the location of User 2, it would have forwarded the message on to another PServer. At this point $P_{Server2}$ sends a “Trying” message to $P_{Server1}$. Once User 2 receives the invite message, it starts ringing and then sends a “Ringing” response back to User 1 via PServers 1 and 2 (F6, F7, F8 in Figure 2.6).
At this point in the user at endpoint User 2 can decide to accept or decline the call. Assuming User 2 accepts, a “200 OK” response is sent to User 1 via PServers 1 and 2 (F9, F10, F11) informing User 1 that it is accepting the call. User 1 then completes the (invite – ok – ack) three way handshake process by sending a direct “ACK” acknowledgement message to User 2 to confirm the call setup. Users 1 and 2 can now directly communicate as they now know each other’s locations. Once the connection is set up, media information can flow directly between the two users using protocols such as RTP. When one participant in the session decides to disconnect
(User 2 in this case), they send a “BYE” message to the other participant, who then responds with an “OK” acknowledgement (F14).

2.1.2 VoIP Codecs

A codec (coder/decoder) converts a sampled analog signal at a senders end to a compressed digital format. The same codec at the other end of the communication network converts the digital data back into an analog signal to be played out at the receivers end. Codecs can provide data compression in order to save network bandwidth, and also support VoIP facilities such as silence suppression, where samples below a certain threshold are discarded and therefore not encoded or transmitted. Some of the most popular codecs are G.711 [19], SPEEX [20], and OPUS [21]. The use of different codecs can affect the QoS of a VoIP call due to the trade-off between processing time (delay) and compression (reduced sound quality). A comparison of codecs is provided in terms of their QoS scores in TABLE I

<table>
<thead>
<tr>
<th>Codec</th>
<th>Bitrate</th>
<th>MOS</th>
<th>Processing Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.711</td>
<td>64kbps</td>
<td>4.1</td>
<td>0.75ms</td>
</tr>
<tr>
<td>G.726</td>
<td>32kbps</td>
<td>3.85</td>
<td>1ms</td>
</tr>
<tr>
<td>G.729</td>
<td>8kbps</td>
<td>3.92</td>
<td>10ms</td>
</tr>
</tbody>
</table>

TABLE I Codec Comparison

G.711 uses Pulse-Code Modulation (PCM) to sample voice signals at a rate of 8000 samples per second using eight binary digits per sample. Therefore the G.711 encoder creates a 64Kbps bitstream. G.711 supports frame lengths of 40, 80, 160, 240 and 320 samples per frame and has a maximum algorithmic delay equal to 5, 10, 20, 30, and 40 ms which corresponds to the frame length sample sizes at 8000hz. G.711 is a variable bit rate and the minimum size of a frame is 8 bits (1 octet). G.711 is widely supported and it doesn’t implement compression therefore it has low processing rate.
2.2 Voice Call Quality Performance Evaluation

The QoS testing of phone calls has traditionally involved picking up the phone and listening to the quality of the voice. The leading subjective measurement of voice quality is the Mean Opinion Score (MOS) as defined in the ITU (International Telecommunications Union) recommendation P.800 [9]. However, it can be expensive to actually have people sit down, listen to and evaluate the quality of phone calls. Due to a need for objective call quality measurement, a number of standards have been developed which include Perceptual Speech Quality Measure (PSQM) [22], Perceptual Evaluation of Speech Quality (PESQ) [10], and the E-Model (ITU-T G.107) [12].

The MOS provides a widely used 1-5 scoring model which grades VoIP quality from human responses. The MOS value is an indication of the perceived quality of a call and is a widely accepted standard for call quality. Other scoring algorithms are often mapped to the MOS.

2.2.1 Quality of Service vs. Quality of Experience

The terms Quality of Service (QoS) and Quality of Experience (QoE) are sometimes used interchangeably. They represent two distinct methods of performance quality evaluation. Quality of Service is defined by the ITU-T in [23] as the totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service. QoS is a measure of performance from a network point of view, measuring objective parameters such as packet loss, delay, and jitter. Whereas QoE is defined by the ITU-T in [24] as the overall acceptability of an application or service, as perceived subjectively by the user, where complete end-to-end system effects are taken into account. A more elaborate definition of QoE is provided in [25] as “The degree of delight or annoyance of the user of an application or service. It results from the fulfilment of his or her expectations with respect to the utility and/or enjoyment of the application of service in light of the users’ personality and current state”. This definition also notes that overall acceptability may be influenced by the expectations of the users and context. It is a measurement of end-to-end service performance from the user’s perspective and indicates how well the network meets the user’s needs. QoS metrics such as delay, loss, and jitter can precisely measure network characteristics. However the quality experienced by a user cannot be accurately evaluated using these metrics alone alt-
hough they directly and indirectly contribute to QoE issues such as glitches and waiting times [26]. Subjective determination of quality requires a number of people to listen to, and evaluate call quality, which can be expensive and time consuming to carry out.

2.2.1.1 QUALINET

In 2010 a European Cooperation in Science and Technology (COST) action was set up on European Network on Quality of Experience in Multimedia Systems and Services (QUALINET). The main objective of QUALINET is to develop and promote methodologies and metrics that subjectively and objectively measure QoE issues on existing and future multimedia products and services [27] when taking user expectations, context, and interactions between users and content into account. This COST action aims to bring together and coordinate fragmented efforts carried out in this field by European experts from both academia and industry under one umbrella with the aim of imposing a substantial scientific impact on the field.

QUALINET aims to extend the idea of the network-centric QoS in multimedia systems by relying on the concept of QoE. The QUALINET White Paper on Definitions of Quality of Experience (QoE) and Related Concepts [25] states that while the term ‘quality’ has been associated with QoS in the area of communications since the 1990s, the QoE concept has emerged in this field because QoS “does not fully express” all that is involved in a communications service. With the aim of defining QoE, [25] proposes three requirements that the general QoE definition should fulfil, which are as follows;

- Being simple and intuitive while also challenging, powerful and complete
- Not confusing the concept with a given model or representation
- Making clear the relationship and distinction with other related concepts such as QoS

The general definition may be tuned however to specific application scenarios if required.
2.2.2 Speech Quality Evaluation

The ITU algorithm Perceptual Evaluation of Speech Quality (PESQ) aims to determine the quality of a narrowband voice stream by comparing a voice recording from the sender side with the same recording arriving at the receiver. PESQ supersedes the Perceptual Speech Quality Measure (PSQM) with an improved time alignment algorithm and a different perceptual model [28]. An extension to the PESQ model incorporating wideband quality estimation was standardized in 2005 called WB-PESQ [29], however this was superseded with the Perceptual Objective Listening Quality Assessment (POLQA) [11] which covers both narrowband, wideband, and super-wideband speech signals and predicts subjective speech quality in terms of MOS. POLQA provides quality estimation for many codecs including G.711, G.729, and SILK³.

PESQ and POLQA are examples of signal based models where an input speech signal is compared with an output signal after it is transmitted over a communications network. This process requires a speech signal as an input which is not always available. As an alternative, parametric models characterize the network itself by analysing factors such as its probability to drop packets, or the time delay taken for packets to traverse the network. These factors along with information on codecs or Packet Loss Concealment (PLC) algorithms used can provide speech quality estimation. One such estimation mechanism is the E-Model that was developed by the ITU-T.

2.2.2.1 The ITU-T G.107 E-Model

The E-Model is an analytic model that predicts VoIP quality. The ITU-T G.107 E-Model Recommendation [12] defines five categories of end-to-end speech transmission quality that act as a guide in establishing different speech transmission quality levels in telecommunications networks. The five categories are defined in terms of "user satisfaction", which have ratings given by the transmission planning tool [12]. The ratings take the combined effects of various transmission impairments into account. The E-Model is independent of any specific technology that may be used in different types of network scenarios under consideration.

³ The SILK codec was developed by Skype ltd and is incorporated into the IETF OPUS codec [21] which is supported in WebRTC.
The E-model is a widely used subjective measure of end-to-end service performance from a network point of view. QoS is measured using objective parameters such as packet loss, delay, and jitter. It defines five categories of end-to-end speech transmission quality that act as a guide in establishing different speech transmission quality levels in telecommunications networks. The E-model is a useful tool for assessing the combined effect of all parameters and hence differentiating between categories of speech transmission quality. The primary output of the E-model is a transmission rating R-Factor.

The E-model is based on the equipment impairment factor method, following previous transmission rating models. It was developed by a European Telecommunications Standards Institute (ETSI) ad hoc group called "Voice Transmission Quality from Mouth to Ear". The reference connection, as shown in Figure 2.7 is split into a send side and a receive side. The model estimates the conversational quality from mouth to ear as perceived by the user at the receive side, both as listener and talker.

![Figure 2.7 Reference Connection of the E-Model](image)

The model takes loss, delay, echo, codec type and noise, caused by the signals properties, and network characteristics into consideration to produce a single R-rating. The calculation of the transmission factor R is defined in [12]. The minimum
MOS value of 3.1 is equivalent to an R-value of 60 which is the minimum call quality recommended by the ITU-T. The R-factor calculation is defined as follows:

\[ R = R_o - I_s - I_d - I_e - A \]  

(2.1)

where \( R_o \) represents the basic signal to noise ratio and includes circuit noise, the sum of which is referred to the 0 dB Br point and room noise. \( I_s \) is a combination of all impairments due to the voice signal. Factor \( I_d \) represents the impairments caused by delay and \( I_e \) represents impairments due to low bit rate codecs. The advantage factor \( A \) allows for compensation of impairment factors when there are other advantages to the user. Each of the elements in equation (2.1) depends on several configuration parameters. A further description of the R-factor algorithm is as follows:

2.2.2.1.1 Signal to noise Ratio \( R_o \)

The basic signal-to-noise ratio \( R_o \) is defined by:

\[ R_o = 15 - 1.5(SLR + No) \]  

(2.2)

The term \( No \) [in dBm0p] is the power addition of different noise sources

\[ No = 10 \log \left[ 10^{10} N_c + 10^{10} Nos + 10^{10} Nor + 10^{10} N_o \right] \]  

(2.3)

where \( N_c \) [in dBm0p] is the sum of all circuit noise powers, which are all referred to the 0 dB Br point. The \( Nos \) [in dBm0p] value is the equivalent circuit noise at the 0 dB Br point, caused by the room noise \( Ps \) at the send side;

\[ Nos = Ps - SLR - Ds - 100 + 0.004(PS - OLR - Ds - 14)^2 \]  

(2.4)

where Overall Loudness Rating (\( OLR \)) = \( SLR(Sender) + RLR(Receiver) \). In the same way, the room noise \( Pr \) at the receive side is transferred into an equivalent circuit noise \( Nor \) [in dBm0p] at the 0 dB Br point

\[ Nor = RLR - 121 + Pre + 0.008(Pre - 35)^2 \]  

(2.5)

The term \( Pre \) [in dBm0p] is the "effective room noise" caused by the enhancement of \( Pr \) by the listener's sidetone path:
\[
Pre = Pr + 10 \log \left[ 1 + 10^{(10 - LSTR)} \right] 
\] (2.6)

\( Nfo \) [in dBm0p] represents the noise floor at the receive side where;

\[
Nfo = Nfor + RLR 
\] (2.7)

with \( Nfor \) usually set to \(-64\) dBm0p.

### 2.2.2.1.2 Simultaneous Impairment Factor \( Is \)

The \( Is \) value considers non-optimum sidetone, quantizing distortion, overall loudness and other impairments that occur simultaneously with voice transmission. \( Is \) is calculated by;

\[
Is = Iolr + Ist + Iq 
\] (2.8)

where \( Iolr \) represents the decrease in quality due to low loudness values (OLR), this is calculated by;

\[
Iolr = 20 \left\{ \frac{Xolr}{8} \right\}^{\frac{1}{8}} - \frac{Xolr}{8} \right\} 
\] (2.9)

and

\[
Xolr = OLR + 0.2(64 + No - RLR) 
\] (2.10)

The value for \( Ist \) factors in the impairment due to non-optimum sidetone and is calculated by;

\[
Ist = 12 \left[ 1 + \left( \frac{STMRo - 13}{6} \right)^{8} \right]^{\frac{1}{8}} - 28 \left[ 1 + \left( \frac{STMRo + 1}{19.4} \right)^{35} \right]^{\frac{1}{35}} - 13 \left[ 1 + \left( \frac{STMRo - 3}{33} \right)^{13} \right]^{\frac{1}{13}} + 29 
\] (2.11)

where;

\[
STMRo = -10 \log \left[ 10^{-\frac{STMR}{10}} + e^{\frac{T}{4}} \frac{TELR}{10} \right] 
\] (2.12)

STMR is the Sidetone Masking Rating and TELR is the Talker Echo Loudness Rating.
The value for impairment factor $Iq$ represents the impairment caused by quantizing distortion;

$$Iq = 15 \log \left[ 1 + 10^Y + 10^Z \right]$$  \hspace{1cm} (2.13)

where

$$Y = \frac{R_o - 100}{15} + \frac{46}{8.4} - \frac{G}{9}$$  \hspace{1cm} (2.14)

and

$$Z = \frac{46}{30} - \frac{G}{40}$$  \hspace{1cm} (2.15)

and

$$G = 1.07 + 0.258Q + 0.0602Q^2$$  \hspace{1cm} (2.16)

$$Q = 37 - 15 \log(qdu)$$  \hspace{1cm} (2.17)

The value for $qdu$ (Quantizing Distortion Units) is for the whole connection between the send side and the receive side.

2.2.2.1.3 Delay Impairment Factor $Id$

The equation for $Id$ is broken down into three factors which represent impairments due to delay as follows;

$$Id = Idte + Idle + Idd$$  \hspace{1cm} (2.18)

The factor $Idte$ provides an estimate for the impairments due to talker echo;

$$Idte = \left[ \frac{Roe - Re}{2} + \sqrt{\frac{(Roe - Re)^2}{4} + 100 - 1} \right] \left( 1 - e^{-T} \right)$$  \hspace{1cm} (2.19)

where $Roe=-1.5(No-RLR)$ and $Re=80+2.5(TERV-14)$ with


\[ TERV = TELR - 40 \log \left( \frac{1 + \frac{T}{10}}{1 + \frac{1}{150}} \right) + 6e^{-0.3T^2} \]  

(2.20)

TERV is Talker Echo Loudness Rating. If \( T < 1 \)ms, talker echo should be considered as sidetone, therefore \( Idle \) should be set to 0. The factor \( Idle \) factor in listener echo and is described by;

\[ Idle = \frac{Ro - Rle}{2} + \sqrt{\frac{(Ro - Rle)^2}{4} + 169} \]  

(2.21)

with \( Rle=10.5(WEPL+7)(Tr+1)^{0.25} \) where \( WEPL \) stands for Weighted Echo Path Loss. The factor \( Idd \) represents the impairment caused by ‘too-long’ absolute delay \( Ta \). Where \( Ta<=100 \)ms;

\[ Idd = 0 \]  

(2.22)

However when \( Ta >= 100 \)ms, \( Idd \) is calculated as follows;

\[ Idd = 25 \left( 1 + X^6 \right)^{\frac{1}{6}} - 3 \left( 1 + \left( \frac{X}{3} \right)^6 \right)^{\frac{1}{6}} + 2 \]  

(2.23)

Where

\[ X = \frac{\log \left( \frac{Ta}{100} \right)}{\log 2} \]  

(2.24)
2.2.2.1.4 Equipment Impairment Factor $I_e$

The value for $I_e$ typically represents the effects of VoIP impairments that are introduced by different low bit-rate codecs. This factor aims to cover all perceptively diverse effects such as distortion, sound/voice degradation that are associated with a codec used in a connection, except for those that are already covered in the E-Model by another factor (e.g. absolute delay covered by $I_d$). Appendix I of [31] maintains an up to date listing of values $I_e$. The packet-loss dependent effective equipment impairment factor $I_{e-eff}$ is derived using a codec specific value for the equipment impairment factor at zero packet-loss $I_e$ and the packet loss robustness factor $Bpl$. These default values are also listed in Appendix I of [31], and along with the packet-loss probability $P_{pl}$, $I_{e-eff}$ is calculated as:

$$I_{e-eff} = I_e + (95 - I_e) \cdot \frac{P_{pl}}{P_{pl} + Bpl} + Bpl$$  \hspace{1cm} (2.25)
The Burst Ratio \( (\text{BurstR}) \) describes the burstiness of a loss distribution and is defined as:

\[
\text{BurstR} = \frac{\text{Average length of observed bursts in an arrival sequence}}{\text{Average length of bursts expected for the network under "random" loss}}
\]

\( \text{BurstR} = 1 \) when packet loss is random and \( \text{BurstR} > 1 \) when packet loss is bursty.

It is recommended that the BurstR approach in the E-Model should only be implemented for codecs that have efficient codec-state based Packet Loss Concealment (PLC). Some provisional planning values for equipment impairment factor are listed in TABLE II

<table>
<thead>
<tr>
<th>Codec</th>
<th>Sample Interval</th>
<th>PLC</th>
<th>Ie value</th>
<th>Bpl Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.711</td>
<td>10ms</td>
<td>no</td>
<td>0</td>
<td>4.3</td>
</tr>
<tr>
<td>G.711</td>
<td>10ms</td>
<td>yes</td>
<td>0</td>
<td>25.1</td>
</tr>
<tr>
<td>G.711</td>
<td>20ms</td>
<td>yes</td>
<td>0</td>
<td>4.8</td>
</tr>
<tr>
<td>G.729</td>
<td>20ms</td>
<td>yes</td>
<td>11</td>
<td>19.0</td>
</tr>
<tr>
<td>GSM</td>
<td>20ms</td>
<td>yes</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

TABLE II  CODEC PLANNING VALUES FOR IMPAIRMENT FACTOR Ie

These are recommended values for the PCM, GSM and G.729 codecs when using PLC algorithms to counter the effects of random packet loss.

2.2.2.1.5  Advantage Factor A

The advantage factor \( A \) attempts to account for any expectations the user has of using a particular technology. Not dealing with measurable effects such as codec used or

---

4 The Ie and Bpl values are calculated from a very specific sample of packet loss, they might not represent the impairment due to packet loss in general
signal distortion, the factor $A$ deals with the “ponderation of functionality and transmission quality in users’ expectations of services according to the type of user and the time”. The factor essentially allows transmission planners to calculate the rate at which some customers (users) may accept a decrease in quality in exchange for access, (e.g. satellite communications in remote areas). The E-Model recommends maximum values for $A$ in different scenarios as listed in TABLE III

<table>
<thead>
<tr>
<th>Communication system example</th>
<th>Maximum value of $A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional (Wirebound)</td>
<td>0</td>
</tr>
<tr>
<td>Mobility by cellular networks in a building</td>
<td>5</td>
</tr>
<tr>
<td>Mobility in a geographical area or in a vehicle</td>
<td>10</td>
</tr>
<tr>
<td>Access to remote locations (multi-hop satellite)</td>
<td>20</td>
</tr>
</tbody>
</table>

TABLE III  EXAMPLE FOR ADVANTAGE FACTOR $A$[12]

2.2.2.1.6 Default Values

The E-Model provides default input parameter values for parameters which are not varied during calculation. In a case where all parameters in the calculation are set to these default values, the QoS rating is $R=93.2$. These default values are listed in Appendix I.

2.2.2.2 Mean Opinion Score

In cases where planners might not be familiar with the E-Model, the $R$-factor can be equated to an equivalent Mean Opinion Score (MOS) 1-5 scale. The MOS can be obtained from the $R$-factor using the formula:

$$
MOS = \begin{cases} 
1 & R < 0 \\
1 + 0.035R + R(R - 60)(100 - R)7 \cdot 10^{-6} & 0 < R < 100 \\
4.5 & R > 100
\end{cases}
$$

(2.27)
A further comparison is illustrated in a table of categories in TABLE IV and a MOS graph as a function of R in Figure 2.9.

<table>
<thead>
<tr>
<th>User Satisfaction Level</th>
<th>MOS</th>
<th>R-Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>4.4</td>
<td>93</td>
</tr>
<tr>
<td>Very Satisfied</td>
<td>4.3 – 5.0</td>
<td>90 – 100</td>
</tr>
<tr>
<td>Satisfied</td>
<td>4.0 – 4.3</td>
<td>80 – 90</td>
</tr>
<tr>
<td>Some Dissatisfied</td>
<td>3.6 – 4.0</td>
<td>70 – 80</td>
</tr>
<tr>
<td>Many Dissatisfied</td>
<td>3.1 – 3.6</td>
<td>60 – 70</td>
</tr>
<tr>
<td>Most Dissatisfied</td>
<td>2.6 – 3.1</td>
<td>50 – 60</td>
</tr>
<tr>
<td>Not Recommended</td>
<td>1.0 – 2.6</td>
<td>Less than 50</td>
</tr>
</tbody>
</table>

TABLE IV CATEGORIES OF SPEECH TRANSMISSION QUALITY

Figure 2.9 MOS as a function of R[12]
Further analysis of the E-Model and the effects of delay relating to VoIP are contained in section 3.3.

2.2.3 Potential of Synchronized Time to Optimize QoS

There are a number of methods of measuring delay in a network, including Round Trip Times (RTT), distributed synchronized time and a variable delay estimation mechanism within routers. RTT is an inaccurate mechanism for measuring one-way delay, because delays and paths can be quite different in either direction. However, synchronized time is now becoming more widely available, facilitating precise delay measurements in each direction. This is due to the more widespread deployment of the NTP and the availability of accurate time sources like GPS receivers.

The advantages of using synchronized time are described in [32] where an informed fixed play-out delay was shown to significantly improve voice quality. According to the ITU-T G.114 recommendation, one-way delays should not exceed 150 milliseconds [30]. Therefore, if actual delays are precisely known and well within the G.114 limit, there is room to “manoeuvre” by increasing the play-out time to avoid losing any late packets. The improved QoS is based on the fact that users are more tolerant of increased delay than of increased late loss (once it is within the G.114 limit) [12].

In [32] and [33], the Network Time Protocol was implemented at each end to provide synchronized time. Furthermore, a mechanism was required to relate the RTP time stamps to absolute time. RTCP sender reports (SR), which are nominally used to lip-sync audio/video sessions from the same end device by relating RTP timestamps to common device NTP time, were used to accomplish this. RTCP packets also allow senders to periodically determine round-trip-delay (RTD) time. In a synchronized time environment, RTCP-SR packets allow a sender to determine the delay of incoming packets and thus, with the knowledge of RTD, delays for both legs of the round trip can be calculated in real-time [33]. A mechanism similar to this is used applied to delay calculation in the experimental test-bed described in Chapter 3. It is important to note that all work in [32], [34] and [33] was carried out on wired networks, where synchronization between participants in a VoIP session was less than 10ms. Synchronized time is implemented on wireless networks for the test-bed described in Chapter 3.
2.3 WLAN Networks

The IEEE 802.11 standard [35], extends the 802 Network Standards to the wireless medium by specifying the operation of Wireless Local Area Network (WLAN) communication within the ISM radio bands. First published in 1997, the Physical (PHY) and Media Access Control (MAC) network layers are defined by 802.11. The IEEE 802.11b/g standards use the 2.4Ghz frequency band, whereas 802.11a uses the 5Ghz band, and 802.11n uses a Multiple Input Multiple Output (MIMO) mechanism to utilize both the 2.4Ghz and 5Ghz bands.

2.3.1 IEEE 802.11 WLAN Architecture

The 802.11 wireless LAN standard operates in two modes, ad-hoc mode (peer-to-peer) or infrastructure mode (peer-to-AP). In an infrastructure setup, wireless stations (STAs) connect to, or associate with an Access Point (AP). This grouping of devices (STA(s) + AP) is called a Basic Service Set (BSS) where each STA can connect to an outside network (the Internet) via its associated AP. A BSS uses a Service Set ID (SSID) to identify itself. Multiple APs can be connected via a wired Distribution System (DS) where different BSSs are referred to as an Extended Service Set (ESS). In a scenario where BSSs use different SSIDs, a STA may change association however it must change its association to a different AP which causes a temporary loss of connection. A Basic Service Set ID (BSSID) is the Media Access Control (MAC) address of an AP, this allows a STA to identify a unique BSS AP in an ESS. This research is carried out on an infrastructure WLAN within one BSS, this is illustrated in Figure 2.10.
In order to associate with an AP, a STA must go through a three-phase setup process as illustrated in Figure 2.11. These phases are the scan, authentication, and association phases. On waking or power on, a STA must discover nearby APs by using a passive or an active scan. A passive scan involves listening on each channel for broadcast beacons sent from APs. In an active scan, the STA ‘actively’ sends out a broadcast probe request frame on each channel and then waits for a response from an AP on that channel.
After APs are discovered and one AP is selected, the STA starts the authentication process to authenticate itself with the AP. The STA first sends out an authentication frame, to which the chosen AP responds with additional authentication frames.

The authentication phase controls what nodes can access the AP. It is a network access control mechanism. After successful authentication, the STA moves to associate with the AP by sending an association/re-association request frame to which the AP responds with an association/re-association response frame. Finally, the STA sends an acknowledgement (ACK) frame to the AP. Once the AP receives this ACK frame, the STA is associated with the AP and a valid connection is established between the STA and the AP.

2.3.1.1 **Wireless Medium Access**

The IEEE 802.11 standard provides two services: the contention-based *Distributed Coordination Function (DCF)*, and the polling-based *Point Coordination Function (PCF)*. In DCF, wireless stations must contend for use of the channel at
each data packet transmission. In PCF the medium usage is controlled by the Access Point (AP), which polls the stations (STAs) to ascertain their access requirements. The DCF is the basic medium access mechanism of IEEE 802.11. The DCF is a Medium Access Control (MAC) layer protocol which uses CSMA/CA to maximize data throughput while simultaneously preventing packet collisions. Collisions occur when a single node receives multiple packets at the same time. Collision avoidance aims to mitigate the probability of collisions using a Request to Send / Clear to Send (RTS/CTS) mechanism. The DCF function is widely used in WLAN devices around the world.

![DCF CSMA/CA](image)

When a node has data to transmit it will wait a random back-off time which is measured in slot times and chosen randomly from the interval \([0, CW]\), where \(CW\) stands for the Contention Window. Each node's timer is decremented by one as long as the channel is sensed idle for a Distributed Inter Frame Space (DIFS) time. A DIFS period is equal to \(SIFS + 2 \times \text{SlotTime}\) where \(SIFS\) is Short Inter Frame Space time. If the node senses that another node is using the channel it will pause its own timer until the other node has finished transmitting. When the back-off time reaches zero, the node will “sense” the channel to determine if the channel is clear. If the channel is free, it will transmit an RTS to the destination. The destination will then respond with a CTS frame if it is available. Once the source node receives the CTS frame, it can transmit its data. A Network Allocation Vector (NAV) is sent along with both the RTS and CTS frames. After the sender has sent its data, an acknowledgement frame (ACK) is sent from the destination to the sender to confirm that data was received successfully. If the sender has any more data to transmit, it must begin the
back-off process again. In the CSMA mechanism at the MAC layer, queueing STAs adjust their NAV when data is being transmitted on the medium. The NAV maintains information of waiting traffic on the medium based on the information that is contained in data frames in the contention period. The NAV is essentially a timer that a STA can use to reserve the medium so that communication can proceed uninterrupted. If an ACK is not received due to a collision or if it is corrupted, a source node reactivates the back-off algorithm after an Extended Inter Frame Space (EIFS) interval.

As part of the exponential back-off algorithm, after each unsuccessful back-off iteration, the back-off time is chosen randomly again from [0 –CW] where the Contention Window is doubled up to a maximum value, once the back-off values reaches its maximum, it will remain at that value until it is reset. The value is reset with a successful transmission or when a retry counter reaches its limit. The retry counter is incremented with each failed transmission, and it can be set in order to facilitate the transmissions of a node that experiences multiple failed transmissions. PCF, on the other hand, provides QoS to a certain extent. It was designed to support time-bounded traffic, and defines two periods between consecutive transmissions of Delivery Traffic Indication Message (DTIM) beacon frames: Contention Free Period (CFP) and Contention Period (CP). Beacon frames are sent periodically by the access point (AP), and they carry network BSS information. In particular, DTIM beacon frames are used to indicate the start of a CFP.

2.3.1.2 WLAN Synchronization

Within a BSS, the STAs and the AP maintain a Time Synchronization Function (TSF) timer to be synchronized with one another. STAs synchronize their local TSF timer with the AP’s TSF timer via beacons or probe responses which contain the current value of the AP’s TSF timer. Once the STAs receive the value in the beacons or probe responses, the wireless card hardware automatically sets its TSF timer to the value received from the AP. This synchronization is implemented in the WLAN card hardware therefore its accuracy is not affected by the host CPU. Beacon intervals are typically set 100ms. This synchronization mechanism is particular to the 802.11 WLAN protocol and is separate to the distributed system synchronization implemented later in this work in the real world test-bed using the Networking Timing Protocol (NTP).
2.3.2 IEEE 802.11e: Quality of Service Extensions

To address Quality of Service (QoS) concerns of time sensitive data, the 802.11e (known in industry as WMM or Wireless Multi-Media) was developed as an approved amendment to the IEEE 802.11 standard. It defines a set of enhancements for wireless LAN applications through modifications to the MAC layer. The standard is considered of critical importance for delay-sensitive applications, such as Voice & Streaming Multi-Media over Wireless IP. The amendment has been incorporated into the published IEEE 802.11-2007 standard. In particular, the IEEE 802.11e standard defines a new coordination function called the Hybrid Coordination Function (HCF). HCF includes both a contention-based channel access method, called the Enhanced Distributed Channel Access (EDCA) mechanism, for contention based data transmission, and a controlled channel access, referred to as the HCF Controlled Channel Access (HCCA) mechanism, for contention free data transmission. The HCCA mechanism has not been widely implemented in industry so this research is on the EDCA mechanism.

2.3.2.1 Enhanced Distributed Channel Access (EDCA)

As mentioned in the previous section, the original 802.11 (WiFi) protocol defines two channel access methods:

- DCF, also known as the basic access method, which is a carrier sense multiple access protocol with collision avoidance (CSMA/CA), and
- PCF, which is a polling-based access method and uses a point coordinator to arbitrate access among stations.

In DCF, all the data traffic is transmitted on a first-come-first-served, or best-effort basis. All the stations in the BSS compete for the wireless medium with the same priority. DCF uses an exponential back-off process, which doubles the size of the Contention Window (CW) after each transmission failure. Back-off intervals are chosen randomly from the range \([0 - CW]\). DCF does not provide any means for differentiating traffic classes.

The EDCA mechanism provides differentiated, distributed access to the medium using different priorities for different types of data traffic. EDCA defines four Access Categories (ACs) for different types of data traffic where each AC has a different set
of parameters that are used to contend for the medium. Frames from different types of data traffic (prioritized at the application layer) are mapped into different ACs depending on the QoS requirements of the traffic/application the frames belong to. As illustrated in Figure 2.13, the four access categories are named $AC_{BK}$ (Background), $AC_{BE}$ (Best Effort), $AC_{VI}$ (Video) and $AC_{VO}$ (Voice), where $AC_{BK}$ has the lowest priority and $AC_{VO}$ has the highest priority.

![Figure 2.13 EDCA ACs](image)

Each frame from the higher layers arrives at the MAC layer along with a priority value. This priority value is referred to as User Priority (UP) and is assigned according to the type of application/traffic the frame belongs to (assigned by the Diffserv DSCP value in IP header). There are eight different priority values ranging from 0 to 7 (TABLE V ).
<table>
<thead>
<tr>
<th>UP</th>
<th>AC</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AC_BK</td>
<td>Background</td>
</tr>
<tr>
<td>2</td>
<td>AC_BK</td>
<td>Background</td>
</tr>
<tr>
<td>0</td>
<td>AC_BE</td>
<td>Best Effort</td>
</tr>
<tr>
<td>3</td>
<td>AC_BE</td>
<td>Best Effort</td>
</tr>
<tr>
<td>4</td>
<td>AC_VI</td>
<td>Video</td>
</tr>
<tr>
<td>5</td>
<td>AC_VI</td>
<td>Video</td>
</tr>
<tr>
<td>6</td>
<td>AC_VO</td>
<td>Voice</td>
</tr>
<tr>
<td>7</td>
<td>AC_VO</td>
<td>Voice</td>
</tr>
</tbody>
</table>

**TABLE V USER PRIORITY MAPPING TO ACCESS CATEGORIES**

802.11e provides traffic differentiation by classifying each traffic flow into an AC associated with a MAC transmission queue. Building on the DCF, each AC has its own MAC parameters and behaves independently of, and in parallel with the other queues. The MAC parameters of each AC are: * Arbitration Inter-Frame Space (AIFS), Minimum Contention Window (CW_min), Maximum Contention Window (CW_max) and Transmission Opportunity (TXOP).* The default values of each parameter are reported in TABLE VI where CW_min/max are taken from the DCF and are equal to 15 and 1023 respectively. Similar to the DCF, before starting a transmission, the channel must be detected empty during a time called AIFS. Once the back-off instance is started, the number of back-off slots is computed in the same way using the DCF but also with different CW_min and CW_max parameters for each AC.

To achieve differentiation, instead of using fixed DIFS, as in 802.11 DCF, EDCA assigns higher priority ACs with smaller CW_min, CW_max, and AIFSN to influence the successful transmission probability in favor of high-priority ACs. The AC with the smallest AIFSN has the highest priority, and a station needs to defer for its corresponding AIFS interval. The smaller the parameter values (such as AIFS, CW_min and CW_max) the greater the probability of gaining access to the medium. Each AC within
a station behaves like an individual virtual station: it contends for access to the medium and independently starts its back-off procedure after detecting the channel being idle for at least an AIFS period. The back-off procedure of each AC is the same as that of DCF. When a collision occurs among different ACs within the same station, the higher priority AC is granted the opportunity to transmit, while the lower priority AC suffers from a virtual collision, similar to a real collision outside the station.

<table>
<thead>
<tr>
<th>AC</th>
<th>$AIFS_N_x$</th>
<th>$CW_{min,x}$</th>
<th>$CW_{max,x}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (Background: BK)</td>
<td>7</td>
<td>$CW_{min}$</td>
<td>$CW_{max}$</td>
</tr>
<tr>
<td>1 (Best effort: BE)</td>
<td>3</td>
<td>$CW_{min}$</td>
<td>$CW_{max}$</td>
</tr>
<tr>
<td>2 (Video: VI)</td>
<td>2</td>
<td>$(CW_{min}+1)/2-1$</td>
<td>$CW_{min}$</td>
</tr>
<tr>
<td>3 (Voice: VO)</td>
<td>2</td>
<td>$(CW_{min}+1)/4-1$</td>
<td>$(CW_{min}+1)/2-1$</td>
</tr>
</tbody>
</table>

**TABLE VI** DEFAULT EDCA PARAMETERS

IEEE 802.11e EDCA defines a TXOP$_{limit}$ as the time interval during which a particular station can initiate transmissions. During this period, defined by a starting time and a maximum duration, stations are allowed to transmit multiple data frames from the same AC continuously within the time limit defined by TXOP$_{limit}$. In 802.11e EDCA the higher priority ACs have a longer TXOP$_{limit}$, while lower priority ACs have a shorter TXOP$_{limit}$. Priority differentiation used by EDCA ensures better service to high priority class while offering a minimum service for low priority traffic. Although this mechanism improves the QoS of real-time traffic, the performance obtained is not optimal since EDCA parameters cannot be adapted according to the network conditions.

The values of EDCA parameters are different for each different AC. The higher priority ACs wait a small AIFS time period while the lower priority ACs have to wait a longer AIFS time before they can access the medium. The size of the contention window varies such that the higher priority ACs choose back-off values from a smaller contention window compared to the lower priority ACs. TXOP Limit is also set in a way that the higher priority ACs gain access to the medium for longer durations. As the values of EDCA parameters are AC specific, they are sometimes

- 66 -
referred to as AIFS[AC], CW\text{min}[AC], CW\text{max}[AC] and TXOP Limit[AC]. Thus, basically the main difference between DCF and EDCA is that EDCA uses AC specific parameters AIFS[AC], CW\text{min}[AC] and CW\text{max}[AC] instead of using fixed values DIFS, CW\text{min}, and CW\text{max}. EDCA parameters are periodically advertised by the AP. The AP can adapt these parameters dynamically depending on the network conditions. The 802.11e standard specifies default values of EDCA parameters (TABLE VI) if they are not advertised by the AP.

2.3.2.1.2 Arbitration Inter-Frame Space (AIFS)

The minimum time period for which the medium must be sensed idle before an EDCAF station may start transmission or back-off is not the fixed value DIFS (as in DCF), but is a variable value, AIFS, which depends on the AC for which the EDCAF is contending for. AIFS is derived from the following equation:

$$ AIFS = AIFSN \times \text{aSlotTime} + \text{aSIFSTime} $$

where \text{aSlotTime} is the slot time, \text{aSIFSTime} is the Short Inter-Frame Space (SIFS) time period and Arbitration Inter-Frame Space Number (AIFSN) is used to determine the length of the AIFS. AIFSN specifies the number of time slots in addition to the SIFS time period the AIFS consists of. Different AIFSN values are used for different ACs such that the high priority ACs use smaller values compared to the low priority ACs.

The smaller AIFSN value for a higher priority AC, the shorter time period before it can start transmission or count down its back-off timer compared to the EDCAF for a low priority AC. In this way, the higher priority ACs are guaranteed greater share of the bandwidth. Moreover, smaller AIFS lengths ensure that the higher priority ACs will not suffer from long delays, which are very critical for the delay-sensitive applications/traffics. The lower priority ACs may suffer from longer delays because of the larger AIFS durations they have to wait, but since these ACs are designed for delay-tolerant applications/traffics, a certain amount of delays do not degrade their performance beyond an acceptable limit.
2.3.2.1.3  CWmin and CWmax

The minimum and maximum Contention Window size limits are not fixed as they are in DCF, but are variable depending on the AC. The higher priority access categories have smaller CW values compared to lower priority ACs. A smaller Contention Window for an AC will cause the corresponding EDCAF to choose smaller random back-off values. This gives an AC priority over another AC that has a larger Contention Window, which will have a larger back-off value and thereby longer delays.

The CWmin values for lower priority ACs, AC_BE and AC_BK, are the same as it is for the legacy 802.11 DCF, but the values for higher priority ACs, AC_VO and AC_VI, are as small as one half or quarter of those of the lower priority ACs. This results in smaller back-off values for the high priority ACs and thereby shorter medium access delays. The negative aspect of small contention window sizes for higher priority ACs is that they suffer from higher number of collisions. The reason for this is that the probability of choosing the same back-off values or counting the back-off timers to zero at the same time increases with the decreasing of contention window sizes. CWmax values for high priority ACs are also set such that they are equal or less than the CWmin values for the lower priority ACs so that lower priority traffic doesn’t get locked out of contention in busy traffic periods. High priority access categories are given a consistent and greater share of the bandwidth in the situations when the network has become congested. On the other hand, this may severely degrade the performance of the low priority ACs since they might not be able to decrement their back-off timers because of the smaller post back-off durations of the higher priority ACs. When a back-off process begins, a STA computes a random number uniformly distributed in the range (0, CW) where CW is taken from the CWmin parameter, and initializes its back-off counter with this value. When a collision occurs, the CW is doubled up to a maximum of CWmax, this process is called the Binary Exponential Back-off.

2.3.2.1.4  Transmission Opportunity (TXOP)

The TXOP is the time duration an EDCAF may transmit after winning access to the medium. TXOP is characterized by a maximum duration, called TXOP Limit. As an EDCAF gets the TXOP, it can then start transmitting frames such that the transmission duration does not exceed the TXOP Limit. The transmission duration covers the whole frame exchange sequence, including the intermediate SIFS periods and
ACKs, and the RTS and CTS frames if RTS/CTS mechanism is enabled. A non-zero value of TXOP Limit indicates that the EDCAF may transmit multiple frames in a TXOP, provided that the transmission duration does not exceed the TXOP Limit and the frames belong to the same AC. This is then referred to as Contention Free Bursting (CFB). The consecutive frame transmissions in a TXOP are then separated by SIFS time periods instead of AIFS plus the post back-off periods. The multiple frame transmission is granted to EDCAF (or AC) and not to the station, i.e, it is only allowed for the transmission of frames of the same AC as of the frame for which the TXOP was obtained.

Much research has shown how 802.11e can differentiate between different traffic classes and greatly improve QoS. Nonetheless, where there is significant contention within a traffic class, increased levels of contention delay, packet loss, and jitter can occur resulting in unacceptable one-way delays for some sessions.

2.3.3 WLAN Software Architecture

In order to configure the AP wireless device driver, the Linux Wireless Extension and Wireless Tools were utilised. These are a collection of user space utilities that are part of an Open Source project which was built with the contribution of many Linux users all over the world. The Linux Wireless Extension is a generic API which allows the userspace access to driver configuration and statistics specific to common Wireless LANs. This is very useful in that a single set of tools can allow parameters to be changed in real time without a requirement to restart the driver.

The Linux Wireless Tools are a set of tools that allow for the manipulation of the Wireless Extensions. They use a text interface which provides the following commands:

- **iwconfig**: manipulate the basic wireless parameters
- **iwlist**: allow to initiate scanning and list frequencies, bit-rates, encryption keys
- **iwspy**: allow to get per node link quality
- **iwpriv**: allow to manipulate the Wireless Extensions specific to a driver (private)
- **ifrename**: allow to name interfaces based on various static criteria
Nowadays, most Linux distributions also have integrated Wireless Extensions support in their networking initialisation scripts, for easier boot-time configuration of wireless interfaces. These also include Wireless Tools as part of their standard packages. The latest Linux kernel versions have higher version of Wireless Extension which can enable drivers more functionalities only available on higher version wireless extension. Some important controls provided by the wireless tool package and WLAN device drivers are as follows:

- Basic settings such as the SSID, WEP key
- The WLAN device work mode such as AP mode or STA mode
- When the WLAN device is not working in STA mode, one can set its radio channel
- The encoding rate, the transmission power level, and the retransmission limit

The above controls are common controls supported by most hardware. WLAN device drivers and hardware can extend the control through the “iwpriv” command. Some extensions that were used in this research are as follows:

- EDCA parameter settings: AIFS, $CW_{min}$, $CW_{max}$, TXOP limit
- MAC address-based access control list. It allows users to specify a list of MAC addresses that can or cannot access the AP

Using these tools, a WLAN card and driver can be set to work as an AP or a STA. In order to be an AP, the WLAN card can support all the functionalities of AP such as sending broadcast beacons and accepting authentication/association requests. If the WLAN device is set to be a STA, the WLAN cards can automatically discover APs in the area and connect with them.

The architecture of the WLAN software packages are shown in Figure 2.14. In the kernel, WLAN hardware is controlled by the device driver which is in turn controlled by the Linux Wireless Extensions API in the user space.
In the kernel, the kernel wireless extension interprets the commands between the wireless tool package and WLAN device driver. The setup of the WLAN test-bed is described in Chapter 4.

2.3.4 802.11aa

The IEEE 802.11aa [36] standard was approved by IEEE in 2012. It defines a number of enhancements to IEEE 802.11 that allow for improved audio and video streaming QoS while maintaining data and audio performance. These enhancements include: Groupcast with Retries (GCR), Stream Classification Service (SCS), Overlapping Basic Service Set (OBSS) management, interworking with the IEEE 802.1Q
Stream Reservation Protocol (SRP), and intra-access category (intra-AC) prioritization.

Intra-AC prioritization is an extension of the EDCA function described in 2.3.2.1. Instead of only four transmit queues, six queues are used to provide further prioritization between individual audio and video streams. Two queues are defined for voice traffic (primary—AC_VO and alternate—AAC_VO), and two queues for video traffic (primary— AC_VI and alternate—AAC_VI), one queue for best effort traffic (AC_BE), and one queue for background traffic (AC_BK). These transmit queues are mapped to four independent EDCA functions (EDCAF) to enable traffic differentiation over the wireless channel (Figure 2.15) using four EDCA Access Categories (ACs): VO, VI, BE, and BK. Frames for the voice and video streams belonging to the primary queues are selected with a higher probability than frames from the alternate queues.
The EDCA mechanism provides differentiated, distributed access to the WM for STAs using eight different UPs. The EDCA mechanism defines four access categories (ACs) that provide support for the delivery of traffic with UPs at the STAs. The six transmit queues and ACs are derived from the UPs as shown in TABLE VII:

![Diagram of EDCA mechanism](image)

Figure 2.15 802.11aa intra-AC traffic prioritization
| 1  | AC_BK  | Background | BK  | BK  |
| 2  | AC_BK  | Background | BK  | BK  |
| 0  | AC_BE  | Best Effort | BE  | BE  |
| 3  | AC_BE  | Best Effort | BE  | BE  |
| 4  | AC_VI  | Video       | VI  | A_VI |
| 5  | AC_VI  | Video       | VI  | VI  |
| 6  | AC_VO  | Voice       | VO  | A_VO|
| 7  | AC_VO  | Voice       | VO  | VO  |

**TABLE VII USER PRIORITY TO AC MAPPINGS (802.11AA)**

The research presented in this thesis was initiated before publication of 802.11aa and it is very interesting to see that the approach taken by 802.11aa is quite similar to that presented here. Either way, some mechanism of distinguishing between traffic categories, i.e. intra-AC, is required. This research uses a prioritization mechanism that provides three categories for voice streams with one background category. IEEE 802.11aa provides a different but interesting approach to voice stream prioritization. Work in [37] presents a saturation throughput model which compares the new Intra-AC prioritization with EDCA inter-AC prioritization.

### 2.3.5 VoIP Support over 802.11e

While the 802.11e EDCA function provides a default static parameter configuration, such settings may not yield optimum performance due to traffic characteristics. Alternatively, dynamic, adaptive configurations often provide optimal performance in terms of QoS in real-time applications, traffic throughput, and overall network performance. There has been much research carried out on optimization of 802.11e for different traffic characteristics. This includes work on the *HCF Controlled Channel Access (HCCA)* mechanism and EDCA. Regarding HCCA, work in [38] analyses the
system capacity of HCCA for different traffic conditions when comparing HCCA with a Time-slot Reuse TTXOP (TRTT) mechanism which increases the TXOP allocation to every node. HCCA is evaluated in terms of the capacity of simultaneously supported VoIP calls in [39]. This increase in capacity is brought about by the extension of the Controlled Access Period (CAP) during the Contention Period of 802.11e operation. Further details on the HCCA are available in the 802.11e specification [8]. Due to delays in certification early on, the HCCA mechanism was not widely implemented on devices in industry so it was deemed appropriate to pursue this research using the more widely implemented EDCA channel access mechanism.

2.3.5.1 Optimal EDCA Parameter Configuration

Extensive work exists in the literature regarding the pursuit of optimal parameter configurations for EDCA in many different network scenarios and under many different traffic characteristics. Work by Chen et al. [40] evaluates, via simulation, the capacity of VoIP traffic within both IEEE 802.11e EDCA and HCCA. This work evaluates Constant Bit Rate (CBR) scenarios using the G.711, G.729, and G.723.1 voice codecs where the impact of background traffic (utilizing the contention based EDCA) on the end to end delay of VoIP traffic (that utilizes HCCA) is evaluated. At this point it is worth noting that data traffic traversing an 802.11 network in the downlink direction can experience the negative effects of traffic load asymmetry at the AP, particularly in network saturation scenarios. This asymmetry may be described as AP Throttling [41]. Essentially this scenario arises in an infrastructure network scenario where a BSS contains an AP and n wireless nodes (STAs), where each STA transmits the voice traffic of one speaker on the uplink to a node outside the BSS. According to this role, the AP transmits n times the traffic of each STA on the downlink. However, the 802.11 MAC access mechanism dictates that each wireless node, \{AP, STA1, STA2, ... STAn\}, will access the wireless medium with approximately the same probability. This node equality in transmission opportunities inevitably results in a network bottleneck at the AP on the downlink. Dangerfield et al. in [41], [42] demonstrates that throughput at the AP is the main constraint on capacity for VoIP in infrastructure mode 802.11 networks. The proposed solution alleviates the problem using increased buffer sizes on the AP. Another solution to this problem (AP Throttling) is provided by Clifford et al. [43], in a scenario where there are n wireless nodes, each engaged in a VoIP session, the wireless stations (up-link) each have an
share of the bandwidth, whereas the AP (downlink) has only a:
\[ n/(n+1) \]  \hspace{1cm} (2.28)

share. The proposed solution involves the tuning of EDCA parameters in order to raise the priority of the AP. This proposed 802.11e prioritization approach is very straightforward and imposes only a small computational burden on the BSS nodes.

Much of the research mentioned up to this point focuses on the capacity of 802.11 networks to support VoIP in terms of throughput and the number of active calls. The effect of traffic congestion in the downlink direction that is caused by congestion at the AP is also described along with proposed solutions to this problem. The tuning of 802.11 EDCA parameters provides a means to maximize channel utility and optimize the performance of real-time applications. These QoS improvements can be brought about by tuning the EDCA parameters \textit{AIFS}, \textit{CW}, and \textit{TXOP} depending on the traffic characteristics of the network. This can increase the probability that packets assigned to the top priority category can secure access to the wireless medium faster than packets in the lower priority categories.

In \[44\], Leith et al. use an experimental test-bed to demonstrate that the 802.11e parameters \textit{CW}_{\text{min}}, \textit{TXOP} and \textit{AIFS} work largely in line with their analytic and simulation predictions. They demonstrate that the flexibility provided by the 802.11e MAC can be used in practice to alleviate certain transport layer unfairness involving TCPs assumed path symmetries, and yield significant improvements to TCP’s performance over 802.11. Related work in \[42\] devises a technique for measuring the \textit{MAC level delay} in 802.11 networks using \textit{per-packet interrupts}. This work uses accurate one-way delay measurements to implement the prioritisation of voice in an 802.11e hardware test-bed. The method is capable of achieving accuracies of tens of microseconds, this provides an example of how the 802.11e parameters \textit{AIFS}, \textit{CW}_{\text{min}} and \textit{TXOP} can change the delay experienced by a station. Using these techniques, the authors in \[42\] study delay in the context of protecting a voice call competing against
data traffic in an 802.11 infrastructure network. The work demonstrates that the standard 802.11 DCF function can allow greedy data traffic to seize bandwidth from a low-rate voice call, whereas the 802.11e MAC can be used to protect that voice call against large numbers of data stations by using modest values of $AIFS_N$ while maintaining throughput, mean delays and delay distributions in a range where high voice call quality can be expected.

The $AIFS_N$ parameter is used to implement priority in [45] and in [46] along with the $CWmin$ parameter. Work in [45] provides an experimental evaluation of 802.11e in a mixed voice/background data transmission scenario, using various background traffic loads and packetization schemes for the background traffic. The background traffic (assigned to $AC_{BK}$ category) is initially assigned the same $AIFS_N$ value as the voice traffic (assigned to $AC_{VO}$ category), and then $AIFS_N$ is incremented for $AC_{BK}$ which results in faster medium access for packets in $AC_{VO}$, along with decreased delay, jitter, and packet loss. This work claims that a difference in $AIFS_N$ of 6 ($AIFS_N[AC_{BK}] = 2, AIFS_N[AC_{BK}] = 8$) provides all VoIP STAs with minimum R-factor of 70. However a smaller difference between the $AIFS_N$ doesn’t provide the same level of prioritization to $AC_{VO}$. In all experimental scenarios, the improvement for voice traffic in terms of quality (wired and wireless side), and throughput, is negligible when the $AIFS_N[AC_{BK}]$ is incremented to 3, and only a slight improvement is experienced when $AIFS_N[AC_{BK}]$ is increased to 4. This illustrates that the difference between the $AIFS_N$ values for two ACs must be more than 2 in order to provide significant prioritization.

Further experiments in [46] evaluate the feasibility of the $CWmin$ parameter to improve voice quality and throughput over background traffic. Using a similar test setup to [45], this work examines different settings of $CWmin$ for $AC_{BK}$ traffic ($CWmin[AC_{BK}]$), and for $AC_{VO}$ traffic ($CWmin[AC_{VO}]$). It is observed that STAs with lower values of $CWmin$ experience less average time required to win a transmission opportunity (back-off time), thus experiencing improved quality and throughput. This work notes that in order to avoid a situation where low-priority traffic gets completely blocked out, the sum of $AIFS_N$ plus $CWmax$ for high priority traffic should be greater than $AIFS_N$ for low priority traffic. Work in [47] seeks to find an optimal configuration for EDCA in terms of maximizing data throughput using both analytical model and simulation, and finds that an AIFS parameter is not used in the optimal configuration.
The work in [45], [46], and [47] helped inform the prioritization mechanism described in Chapter 3, where a configuration using a combination of these parameters ($AIFSN$, $CW_{min}$), along with the $CW_{max}$ parameter is used to prioritize certain VoIP sessions over others as required to optimize QoS.

### 2.3.5.2 Game Theoretical Approach towards EDCA Parameter Optimization

Another approach towards the optimization of EDCA parameters is the application of game theory methods to the $CSMA/CA$ mechanism. This area of study seeks to improve network performance by tuning individual self-interested nodes to operate in a way that improves the overall system outcomes. A summary of this area is provided in [48]. In a related work [49], a *game theoretical approach* is used to achieve a fair bandwidth distribution among uplink and downlink flows by tuning the $CW$ parameter in both an application-aware and application-agnostic scenario. A similar approach, this time tuning the $TXOP$ value, is evaluated in [50] where performance is measured against $EDCA$. This work aims to improve QoS for individual nodes along with overall network performance. Work by Banchs et al. [51] extensively evaluates a *game theoretical approach* when using a distributed Proportional-Integral controller to ensure that selfish nodes cannot benefit from their own misbehaviour.

### 2.3.5.3 802.11 VoIP Capacity

Research in [52] finds upper bounds for the number of voice calls that can run concurrently in an 802.11b infrastructure network, while maintaining an acceptable level of QoS. Quality is measured in terms of *packet delay*, *packet loss* and *jitter*. The network VoIP capacity can vary depending on factors such as *network transmission rate*, *voice codec* used, and the extent of *background traffic* present. This work concludes that 5 VoIP calls can be supported in an infrastructure network before some calls begin to experience significant degradations in quality. The VoIP capacity of IEEE 802.11b networks is further studied by Hole and Tobagi [53], where an upper bound on the number of VoIP calls is measured. This work evaluates a wide range of scenarios including different *delay constraints*, *channel conditions*, and *voice encoding schemes* including $G.711$ and $G.729$. The network capacity is found to be sensitive to packetization and wireless network delays where packet size selection can maximize capacity. This work also concludes that the $G.729$ codec provides a higher capacity for VoIP unless VoIP quality MOS scores of 3.65 are required, in
which case the G.711 codec is preferable. Cai et al. [54] compute the CBR voice capacity in 802.11 WLANs using a theoretical model, and verify their model via simulation results. They conclude that the delay bound of real-time applications can only be guaranteed when the AP is not saturated.

Dao et al. [55] find an analytical formulation using throughput analysis of voice traffic to compute the voice capacity and they then compare it with their simulation results using the commercial network simulator OPNET. Trad et al. [39] also use the same formulation to evaluate the voice capacity in IEEE 802.11e. Hegde et al. [56] and Harsha et al. [57] also evaluate the voice capacity in IEEE 802.11e using analytical models. In particular, [57] analyses the effect of TCP traffic on voice calls. Work in [58] analyses the capacity of VoIP in 802.11WLANs, comparing experimental results with simulations and theoretical results. The authors identify certain factors that are often overlooked in the literature, in both experiments and simulations, which affect network capacity. The packet generation offset, or the difference in the times that packet generation begins, in simulation can cause unrealistic buffer overflow when all nodes in a network begin to generate packets at the same time, thus filling the AP buffer. The retry limit for wireless clients can affect packet loss when a short limit is reached causing packets to be dropped. This value can vary depending on wireless cards and simulators used. The buffer size at the AP can fill up when downlink packets queue while waiting for the AP to gain access to the wireless medium, this can cause packet loss as the buffer overflows. These factors can be tuned differently and can affect the QoS of VoIP. They are taken into account in simulation and experimental results presented in Chapter 5 and Chapter 6.

2.4 Time Synchronization

The ongoing development of communications and industrial applications over computer networks increasingly relies on accurate time synchronization. As mentioned in Chapter 1, it is estimated that there will be 25 billion devices connected to the Internet in 2015, with 3.7 connected devices per person [2]. In terms of QoS, real-time applications such as VoIP, Internet Protocol Television (IPTV), and Multiplayer Online Gaming (MOG) can benefit from accurate time synchronization. With the emergence of the Internet of Things (IoT) and Cyber-Physical Systems (CPS) [66], the need for time synchronization is spreading towards providing for the future development of existing applications such as information exchange between
vehicles and highways in transportation systems for example. Similar requirements exist in power grid systems where renewable energy sources will require different control systems governing their access to, and interaction with the grid. There are also emerging time synchronization requirements for high value applications in the medical and financial fields.

2.4.1 Future Timing Requirements

With a view towards meeting these demands, ongoing work at NUI, Galway in the areas described in [66] and [4], is aimed at IoT development with respect to time synchronization. Current work in the Performance Engineering Laboratory (PEL) [67] is contributing to the Time Aware Applications, Computers, and Communications Systems (TAACCS) research project [4]. The research project aims to cater for a new economy that will be built on the massive growth of endpoints on the Internet where precise and verifiable timing will be required. This will apply to existing areas such as intelligent transportation, financial industry, and telecommunications and to developing areas like the Internet of Things (IoT). The TAACCS white paper [4] outlines critical research in six areas to be carried out on this topic. There are trade-offs required on Oscillators in networks between performance, power and cost, while Time Transfer Systems will need to deliver signals to an exponential increase in endpoints. Time Aware Network and Communications systems will require development in a number of areas, while Timing Support for Applications will require cross-discipline research on providing predictable execution in software and hardware and providing scalable supply to large numbers of systems. Development Environments are required for designing, simulating, and generating code for time aware systems, and finally, Time Aware Applications will be the ultimate consumers of timing signals will have diverse requirements, for example, Machine to Machine (M2M) technology which is expected to grow as part of the IoT will be a major consumer of time synchronization [68].

2.4.2 Network Timing

The implementation of timing distribution uses packet-based protocols such as NTP or PTP to distribute timing information over a hierarchy of nodes. A time source such as GPS provides time to the node at the top of this hierarchy. The
802.1as standard [69] aims to ensure that the synchronization requirements are met for time sensitive applications, such as audio and video, in bridged networks and LANs where the transmission delays are fixed and symmetrical. This includes the maintenance of synchronized time during normal operation and in the case of addition, removal, or failure of network components and network reconfiguration. This standard aims to provide accurate time synchronization in 802.11 WiFi networks by invoking MAC specific timestamp reporting elements that are part of IEEE 802.11v [70]. The Precision Time Protocol (PTP – IEEE 1588) [71], allows the synchronization of distributed clocks to an accuracy of less than 1 microsecond. Along with the provision of microsecond synchronization, PTP aims to have minimal processing and networking requirements enabling it to be implemented on low cost devices. PTP is designed for privately managed networks that use specialized hardware and therefore provide more accurate time synchronization. NTP on the other hand, is designed to operate over large, dynamic networks where the transmission delays are varied.

Related work at NUI Galway that is patent pending\(^5\) provides a mechanism that improves time synchronization accuracy in 802.11 networks, as well as an improved PTP. In 802.11 WiFi networks, this work improves time synchronization to the point that it is similar to that achieved over wired networks. This significant improvement dynamically determines the delays incurred by time messages as they traverse a wireless link. These delays may then be used to reduce errors associated with non-deterministic time message latencies, which leads to greatly improved synchronization accuracy. For this work, the Network Time Protocol (NTP) is used to provide sub-millisecond time synchronization accuracy in order to facilitate the accurate calculation of M2E delays for VoIP. This section details issues associated with time synchronization and delay measurement including issues regarding delay calculation due to the asynchronous nature of wireless networks along with details on the operation of the NTP protocol.

2.4.2.1 Delay Measurement & Synchronization

In order to measure the time that events occur on different nodes in a distributed computer network, and the duration of time between the events, it is necessary to

\(^5\) PhD thesis by J. Shannon, NUI, Galway, 2014
provide a common time reference of specified accuracy [72]. Accurate calculation of one-way packet delays in a network test-bed requires synchronization between nodes of within millisecond accuracy. If two nodes need to synchronize their clocks, then they must exchange some information regarding the state of their clocks, whereby a node (N1) sends a message at time $\tau_i$ to another node (N2) containing a timestamp of the clock time $\tau_i$ (Figure 2.16). N2 can then set its own clock to the time of N1. This process is however, sensitive to delay in a distributed computer network as the transmit time $\varphi$ is not factored in to the calculation as it cannot be known.

The delay $\varphi$ is composed of send time, access time, propagation time, and receive time according to [72]. The traversal times are defined as follows:

- **Send Time** – The time interval between recording a timestamp by the sender application and the arrival of that message at the Network Interface Card (NIC) before transmission. This interval varies on different systems with different hardware components and software characteristics.

- **Access Time** – The time delay experienced by a packet awaiting access to the communication medium at the Medium Access Control (MAC) layer, for example the 802.11 CSMA/CA contention based protocol as detailed in 2.3.1.

- **Propagation Time** – The time taken to traverse the communication medium.
- **Receive Time** – The time taken for the receiver to receive the message at the NIC and passing it up the layers of the communication hierarchy to the receiver application.

An improved synchronization mechanism for distributed systems is one where a node (N1) can send a message at \( \tau_i \) to N2 requesting a reply that contains N2s time [73]. N2 then sends a response message to N1 containing the timestamp \( \tau_{i+1} \) at time \( \tau_{i+2} \). On reception of this reply message, N1 records the receive time \( \tau_{i+3} \) and can then calculate the round trip delay \( \Delta \) to be:

\[
\Delta = (\tau_{i+3} - \tau_i) - (\tau_{i+2} - \tau_{i+1})
\]

and calculate an approximate one-way delay value \( \delta \) to be \( \delta \approx \text{RTT}/2 \) Figure 2.17. This discounts the processing time \( \varphi \) at N2. This synchronization method assumes network symmetry which is not always the case.

---

Figure 2.17 Round-trip synchronization
The calculated values for $\delta_i$ and $\delta_{i+1}$ may vary, particularly in wireless networks due to the nature of their medium access. Another synchronization mechanism aims to reduce the effect of send time and access time.

2.4.2.2 The Network Time Protocol

The Network Time Protocol (NTP) is built on the Internet Protocol (IP) and is a ‘connectionless’ protocol that uses the User Datagram Protocol (UDP). It was designed to maintain accuracy and reliability on the internet and to be usable in unreliable networks where uplink and downlink paths are changeable and asynchronous. NTP consists of a network of time servers, clients, and interconnecting transmission paths. ‘Primary’ servers are synchronized by a primary reference time source which is usually an atomic clock [74]. Time synchronization paths usually follow a hierarchical system where the primary reference servers exist at the root and servers on the leaves tend to have decreasing levels of accuracy. The accuracy of an NTP server is defined by a numeric value called a “Stratum”, this naming convention is borrowed from the telephone industry. Primary servers are assigned to level 1, and on each successive level in the hierarchy the values increment [75]. A typical NTP hierarchy is displayed in Figure 2.18(A), where arrows show active synchronization paths and the direction of timing information flow. The thin lines represent backup paths where timing information is exchanged but are not necessarily used to synchronize local clocks [74]. In Figure 2.18(B), the same graph is shown but this time the connection ‘$\chi$’ is missing therefore the network has reconfigured itself automatically and now one of the Stratum 1 servers has become Stratum 2.

![Figure 2.18: NTP Subnet 1 and Subnet 2](image)

NTP servers exchange timestamps with one or more subnet peers that they then use to calculate individual round delays, clock offsets, and error estimates.
Timestamps exchanged between a Server and a Client are represented in Figure 2.19 as \( \tau_i \) and the three more recent timestamps \( \tau_{i-1} \), \( \tau_{i-2} \), and \( \tau_{i-3} \) are used to calculate round trip delay \( \Delta_i \) and clock offset \( \theta_i \) of the Server relative to the client at time \( \tau_i \).

![Figure 2.19 NTP Delay & Offset Measurement](image)

If \( \delta_1 = \tau_{i-2} - \tau_{i-3} \) and \( \delta_2 = \tau_{i-1} - \tau_i \), then

\[
\Delta_i = \delta_1 + \delta_2 \tag{2.30}
\]

and

\[
\theta_i = \frac{\delta_1 + \delta_2}{2} \tag{2.31}
\]

In an iteration of this process, each NTP message contains the previous three timestamps, which are used along with the current message timestamp to calculate delay and offset. Therefore these calculations can be carried out by a server and the peer independently using a single message stream [74]. An NTP timestamp is a 64-bit unsigned fixed-point number with the first 32 bits as the integer part and the second 32 bits as the fraction part. The timestamp is interpreted as the number of
seconds since 00.00h on the 1st of January 1900, the NTP packet header structure is illustrated in Figure 2.20.

<table>
<thead>
<tr>
<th>LI</th>
<th>VN</th>
<th>Mode</th>
<th>Stratum</th>
<th>Poll</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Root Delay</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Root Dispersion</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reference ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reference Timestamp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Origin Timestamp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Receive Timestamp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Transmit Timestamp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Extension Field 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Extension Field 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Key Identifier</td>
</tr>
</tbody>
</table>

Figure 2.20 NTP packet Header

The fields in the NTP packet are as follows

- **LI (Leap Indicator)** – Warning of impending leap second to be added or removed at the end of the current day
- **VN (Version Number)** – NTP version in use
- **Mode** – Indicate the current operating mode
- **Stratum** – Stratum number of the server
- **Poll Interval** – The current desired interval between messages sent
- **Precision** – Precision of servers local clock (secs)
- **Root Delay** – Indicates the total round trip delay from the server to its primary reference (secs)
- Root Dispersion – Indicates the maximum error of the server clock relative to its primary reference (secs)

- Reference ID – Identifies the particular server or reference clock. The interpretation depends on the value in the Stratum field and where the server lies in the hierarchy of the current subnet

- Reference Timestamp – The time when the system clock was last set/corrected*

- Origin Timestamp – The time at the client when the request departed for the server*

- Receive Timestamp – The time at the server when the request arrived from the client, at the server*

- Transmit Timestamp – The time at the server when the response departed for the client*

- Destination Timestamp – The time at the client when the reply arrived from the server*

* VALUES IN NTP TIME FORMAT

The NTP timestamp is used to calculate delays in the test-bed in this research along with the UNIX timestamp which is interpreted as the number of seconds since 00.00h on the 1st of January 1970, therefore the value of 2208988800 seconds is added to the NTP integer part for delay calculations.

The performance of time synchronization protocols such as NTP can degrade when run over wireless networks due to their inherent path asymmetries that can cause variable one-way delays. This is an issue that has been researched at NUI, Galway, as mentioned above, and is detailed in [1].

2.5 Software Defined Networking (SDN)

With the advent of the IoT, mobile networks will need to handle big data, a big increase in network traffic, and new types of devices such as smart cars and appliances [14]. These devices will produce traffic as well as consume it and will share networks with smartphones and tablets. All of this new traffic will be in addition to
users increasingly high expectations for good quality traditional RTC on mobile devices.

Headed by the Open Networking Foundation (ONF), Software Defined Networking (SDN) is an emerging network architecture that will be more suitable for today’s high-bandwidth, dynamic applications. The core concept of this model is that network control is abstracted away from individual devices towards centralized accessible computing devices allowing applications and network devices to treat the network as a logical or virtual entity [3]. In this paradigm, network intelligence is centralized in software-based SDN controllers, in effect, rendering the network as a single logical switch. SDN also simplifies networking devices as these would no longer need to understand multiple protocol standards. Instead they would just accept instructions from SDN controllers. In the long term, it is envisaged that SDN will enable services to experience the required QoS expected from networks. The SDN architecture is illustrated in Figure 2.21.

![Software Defined Network Architecture](image)

Figure 2.21 Software Defined Network Architecture
Although the idea of SDN was originally to improve wired networks, especially backbone networks such as improving the performance of switches, wireless networks will likely experience most of the added value from this new paradigm. Mobile networks are often characterized by very dynamic network conditions which tend to yield decreased QoE when compared to good, symmetric wired connections. In the context of mobile networks topology and load, information from multiple network domains (e.g., mobile and Wi-Fi access networks), combined with application requirements and user context, can enhance mechanisms such as mobile data offloading and vertical (cross-technology) handovers. Especially in the case of Wi-Fi, this can enable efficient management of wireless resources (e.g., reducing wireless interference through appropriate channel selection), Wi-Fi access sharing, and content caching, which can further improve the offered QoE and service performance.

2.5.1 OpenFlow

OpenFlow [59] is the first standard communications interface defined between the Control and Infrastructure layers of an SDN based architecture [3]. OpenFlow enables access to and manipulation of the forwarding functionality of network routing devices via the OpenFlow protocol, this protocol is required to move network control out of the networking switches to the centrally controlled software. The OpenFlow protocol identifies network traffic as flows based on parameters such as usage patterns, applications and cloud services which enables granular control. The use of flows enables the network to respond to real-time changes at application, user, and sessions levels.

2.5.2 Trends Driving Network Evolution

Conventional hierarchical networks are designed for a static client-server based computing model and are increasingly ill-suited to the emerging dynamic computing and storage needs of applications, data centres, and carrier environments. This has been exacerbated by a rapid growth in mobile devices and content, server virtualization and cloud services. The following trends are listed by the Open Networking Foundation [60] as the key drivers towards SDN:

- **Evolving Traffic Patterns:**
The growth of mobile, server virtualization and the cloud is causing a change in traffic patterns. Users now access corporate content and applications from different locations, on different devices, at any time.

- **The “Consumerization of IT”:**
  - Users are increasingly using mobile devices such as smartphones, tablets, and notebooks to access the Internet. These different personal devices must be accommodated while protecting corporate data and intellectual property.

- **The Rise of Cloud Services:**
  - Enterprises have embraced public and private cloud services, resulting in major cloud growth. They now want the ability to selectively access applications, infrastructure, and other resources on demand. This requires an **elastic** scaling of computing, storage, and network resources.

- **Big Data requires More Bandwidth:**
  - Handling today’s “big data” requires parallel processing on a large scale with potentially thousands of servers directly connected to each other, this requires more and more bandwidth.

- **Existing Network Complexity is Slowing Improvements**
  - Networking protocols are typically defined in isolation with each one solving a particular problem. This has resulted in a major complexity in today’s networks where adding a network device might involve dealing with multiple switches, routers, firewalls and Web Authentication portals, as well as updating VLANs, QoS and other settings at the device level. Tackling this complexity increases the risk of service disruption therefore networks have remained relatively static.
  - Existing networks can manually provide differentiated levels of QoS with network devices being configured separately, however this static configuration cannot dynamically adapt to changing traffic, application and user requirements.
• **Inability to scale**
  
  o With today's virtualized data centres, traffic patterns are increasing-
  ly more dynamic and therefore unpredictable. Large companies
  such as Google and Facebook employs large-scale parallel pro-
  cessing algorithms and datasets requiring *hyperscale* networks that
  can provide high-performance, low-cost connectivity among poten-
  tially millions of physical servers. This degree of scalability cannot
  be manually configured in the existing network architecture.

• **Vendor dependence**
  
  o SDN is a response to the mismatch between market requirements
  and network capabilities. Carriers and enterprises ability to respond
  to changing business needs and user demands is affected by the
  typical life cycle of vendors product life cycle which can be in the
  order of years.

  SDN has been developed as a response to these trends and problems, solutions to
  which would not be feasible without such dynamic, logically centralized network
  control. The intelligent Access Point (iAP) presented in this thesis can be viewed as a
  move in the direction of SDN, whereby devices (iAP) are controlled by a central
  ized controller that has full information and access to network devices.

2.5.3 **SDN and WLAN Networks, QoS/QoE Benefits**

  Given the importance of WLAN networks in Internet connectivity, relatively little
  is known about how to exploit the benefits of SDN for WiFi [61]. Work in [90] pre-
  sents a WiFi SDN called AeroFlux which can significantly reduce control plane
  traffic via a 2-tiered control plane. In this model, frequent, localized events are han-
  dled close to where they originate (close to an AP for example) in a *Near-Sighted
  Controller* (NSC), whereas global events which require a broader picture of the net-
  work are handled by a *Global Controller* (GC) which is logically located in a data
  centre for example. Work in [62] presents a wireless test-bed for dense WiFi net-
  works that evaluates how different WiFi management strategies affect user
  experience and network behaviour in both residential and enterprise settings. This
  work presents a prototype SDN WiFi framework that can realise a broad range of
  management techniques for channel, power, and association control by using a virtu-
al AP abstraction to remove the WiFi control logic from the physical infrastructure. Work in [63] discusses the necessary steps required for the migration from today’s residential network model towards a converged access platform based on SDN. It states that the integration of OpenFlow into the entire chain of residential devices will be beneficial in many ways including the simplification of networks and improvement of QoS via the notion of a flow. Four primary benefits that SDN brings to carrier WiFi networks are listed in [14]; these include scalability, policy control, network monitoring and location services. These benefits are provided in the context that users now expect the same level of service in wireless networks although not necessarily via mobile phones.

In terms of SDN benefits in QoS and QoE, work in [65] details an SDN enabled QoE monitoring and enforcement framework for Internet services in mobile networks named Internet Service quality Assessment and Automatic Reaction (ISAAR). Using a three component structure consisting of a classification and monitoring unit (QMON), a decision unit (QRULE), and an enforcement unit (QEN), ISAAR aims to satisfy customers service quality expectations from a network operators point of view by optimizing the transport of the most popular traffic flows (e.g. Video streaming, Voice, and Facebook). Work in [64] presents an example of SDN use in improving QoE where it details an idea for an SDN controller that has a global view of a network domain which can be used to both perform optimized path assignment, and achieve the desired level of QoE performance from the users perspective.

### 2.6 Linux Traffic Control

The prioritization module described in section 3.2.1 implements prioritization amongst VoIP calls in the iAP. This functionality requires a method to prioritize selected VoIP traffic in the 802.11e CSMA/CA framework. The Linux Traffic Control suite [76] carries out this function by changing “mangling” the DSCP value in the header of IP packets belonging to VoIP sessions that are meant to be prioritized. This section provides some background information on the workings of the Traffic Control suite and, in particular the iptables framework which is used in the iAP prioritization module.

The Linux Traffic Control (TC) suite provides a set of queuing systems and mechanisms that by which packets are received and transmitted on a Linux system or router. TC determines if packets should be accepted on a system on the input inter-
face or if they should be forwarded to another destination and assigned to the output interface. In either case, packet manipulation is possible, including manipulation of values in the IP header. Thus it is possible to implement prioritization of data packets in a wireless AP using Linux TC to manipulate the Diffserv Code Point (DSCP) value in the IP header. Within the 802.11e EDCA channel access mechanism, packets are assigned to the appropriate queues based on DSCP values in their IP headers.

2.6.1 Netfilter

As part of the TC suite, Netfilter is a Linux kernel framework that facilitates packet filtering, port translation and Network Address Translation (NAT). Network packets are handled by `iptables` with a set of rules which are ordered together to form a list called `Chains`. These chains are then grouped together to form `Tables`. The Netfilter framework, of which `iptables` a part of, allows a system administrator to define rules for how to deal with network packets. Rules are grouped into `chains` where each chain is an ordered list of rules. Chains are grouped into `tables` where each table is associated with a different kind of packet processing. Every network packet arriving at, or leaving from, the computer traverses at least one chain, and each rule on that chain attempts to match the packet. Netfilter also provides functionality for the modification of packets via packet mangling; this method is used in the iAP to edit DSCP values in IP headers as detailed in Chapter 3. These firewall rules are stored in packet filtering tables that are integrated into the Linux kernel. Inside the packet filtering tables the rules are grouped together in what are known as chains, or rule chains. This process is illustrated in Figure 2.22.
Iptables is used to set up, maintain and inspect the tables of packet filtering rules in the Linux kernel. The Netfilter/Iptables [77] framework defines five major hooks which are used to intercept and manipulate packets. These are PREROUTING, FORWARD, POSTROUTING, INPUT and OUTPUT chains. Incoming packets enter via the PREROUTING and INPUT chains. Outgoing packets use the OUTPUT and POSTROUTING chains. The FORWARD hook allows packets to pass through a device such as a router, or a gateway, by entering on one interface and leaving on another. The INPUT hook allows packets to be processed before they are delivered to a local process, while the OUTPUT hook allows the packets to be processed just after they are generated by a local process. Packets can be processed before they leave a network interface via the POSTROUTING hook and the PREROUTING allows packets to be processed just as they arrive from a network interface.

There are four independent tables in iptables. The FILTER table is the default, while the NAT table is consulted when a packet that creates a new connection is encountered. The MANGLE table is used for packet modification and has five built-in chains; PREROUTING (for altering incoming packets before routing), OUTPUT (for altering locally-generated packets before routing), INPUT (for packets coming into
the box itself), FORWARD (for altering packets being routed through the box), and POSTROUTING (for altering packets as they are about to go out). The iAP prioritization module uses the MANGLE table to alter DSCP values in the IP header of packets that are deemed to be prioritized in the iAP. The RAW table is a new addition to iptables, and is used for configuration exemptions to mark packets that should not be handled by the connection tracking system.

Each of the tables described above has chains. It is also possible to create custom chains to organize the rules that are created. The policy of a chain is to ACCEPT, DROP, RETURN, or QUEUE determines the fate of packets that reach the end of the chain without otherwise being sent to a specific target. Packets that match a rule pass to the ACCEPT target, while packets that don’t match a rule are dropped via the DROP target. If no packets match any rules, then all packets are considered to match and are accepted. A simplified illustration of this process is provided in Figure 2.22 with a more detailed graph of the iptables packet traversal process is provided in Appendix II. Details of how iptables is used to implement prioritization of packets in the iAP are provided in Chapter 3 and Chapter 4.

2.7 Summary

This chapter has provided background information and a review of the literature for areas involved in this research which set the context for the challenges that are addressed by this thesis. Background information is provided on VoIP, voice call quality estimation, WLAN networks including the 802.11e QoS WiFi extension, and computer and network time synchronization including a summary of NTP. Recent developments in the areas of TAACCS – which seeks to provide precision timing on the Internet, and the concept of SDN were also covered. Lastly, background information was provided on the Linux Traffic Control framework which is used to implement a core functionality of the prioritization module described in Chapter 3.
Section III

SYSTEM DESIGN & IMPLEMENTATION
Chapter 3

SYSTEM DESIGN

The previous chapter provided background information and a review of the literature in the areas of VoIP, quality performance evaluation, WLAN networks, and time synchronization. It also reviewed recent developments with relevance to this thesis and its objectives – the first is the Time Aware Applications, Computer and Communications Systems (TAACCS) which aims to highlight the opportunities and challenges of implementing time-awareness into networks and devices. The second is the broad area of Software Defined Networking (SDN) which aims to improve the flexibility, scalability, and responsiveness of networks. These provide a context for the contribution of this thesis where a dual approach of simulation and emulation is taken. As is commonly the case, the simulation approach enables hypothesis testing under scenarios that otherwise would not be possible to run experimentally due to equipment, timing and funding constraints. The experimental test-bed approach serves a number of purposes. Firstly it serves to assess the practical feasibility of implementing the simulated approach by addressing a range of engineering challenges. It is also used to partly validate some subset of the QoS improvements observed in simulations.

This chapter describes in detail the design stages of both the simulation and test-bed/emulation approaches. It firstly describes the simulation environment used as
well as the rationale for the range of tests undertaken, as well as the prioritization mechanism used in simulations. It then outlines the design and reasoning behind the real-world test-bed (Figure 3.1). The test-bed includes at its core an intelligent Access Point (iAP) which can independently, and dynamically calculate one-way, and intra-one-way delays for multiple individual VoIP sessions along with their resulting E-model R-factors, and then provide prioritization via a mechanism that allows certain VoIP sessions to be prioritized over others based on their R-values. The subsequent Chapter 4 in this section then details the implementation issues.

3.1 Network Simulation

As part of a dual-approach of simulation along with experimental test-bed testing, preliminary simulations were designed using the NS-2 network simulator [79], while all remaining simulations including evaluation of the traffic prioritization mechanism were designed using the NS-3 simulator [80]. Both NS-2 and NS-3 are discrete-event simulators that can be used to accurately simulate a variety of network scenarios. They allow users define and construct network topologies which represent real-world network devices and protocols. Simulation provides numerous advantages over real-world test-bed experimentation such as alleviating the need for impractical large scale network construction which can prove costly. Simulations can be set up and implemented with relative speed in comparison to real world test-beds, while physical phenomena such as interference and noise caused by competing devices within the Industrial, Scientific, and Medical (ISM) radio spectrum can be largely discounted. Default settings were used in all simulations unless otherwise stated, details on these can be found at [79] and [80] for NS-2 (version 2.31) and NS-3 (version 3.11) respectively.

While the default settings were used for wired and wireless network setup, as well as traffic setup, simulations in both NS-2 and NS-3 involved Access Category (AC) re-assignment of packets in the 802.11e model in order to implement prioritization where required. The three highest priority ACs in 802.11e (AC_VO, AC_VI, AC_BE) were used to represent three Multimedia Categories (MCs) which are detailed later in this chapter in section 3.4.1. The implementation of this functionality in NS-2/3 involved the assignment of packets at time of generation, to one of three EDCA access categories. Three preliminary tests are carried out on NS-2 while NS-3 simulations evaluate the iAP prioritization module, as detailed later in this chapter.
The simulation approach was used to test the core idea and to assess the likely benefits it might provide. The initial simulations were designed to evaluate the following criteria in preparation for the eventual development of the iAP:

1. *Achievable VoIP capacity in 802.11e networks*
2. *The effects of different one-way delays on different competing VoIP calls*
3. *The feasibility of VoIP prioritization implementation using 802.11e EDCA parameters*

The first of these implemented a multiple VoIP call scenario. All VoIP calls have the same priority and therefore the same probability of securing wireless network access. The second test evaluates the effects that different one-way network delays can have on the overall one-way delay of VoIP calls, simulating endpoints in different geographical locations. The third test evaluates the feasibility of dynamically manipulating 802.11e EDCA parameters to improve the QoS for VoIP calls based on E-Model R-factor scores. Building on these results, the NS-3 network simulator which became available mid-thesis is subsequently used for more extensive simulations involving the multimedia category prioritization mechanism that is detailed in section 3.4. Both packet loss and delay are factored into QoS calculations in the NS-3 simulations.

Further implementation details of simulation setup are provided in Chapter 4 of preliminary tests on NS-2 along with the more comprehensive simulation tests on NS-3. Results for these simulations are discussed in Chapter 5. The remainder of this chapter details the design of the iAP along with the design of the experimental test-bed and its core components.

### 3.2 Experimental Test-bed

The calculation of accurate one-way delays requires the clocks of all end nodes in the test-bed being synchronized with millisecond accuracy. As outlined in section 2.4, the challenges of implementing time synchronization over WiFi wireless networks is an active research topic at the PEL research group at NUI, Galway, and details of some of this work can be found at [67]. From Figure 3.1, the wired LAN
consists of two desktop machines \((W1, W2)\) that run a VoIP application, with an Ethernet connection to the iAP via a switch along with a traffic generator node which also acts as a sink for data from a wireless traffic generator node. The wireless network consists of an 802.11b BSS that is 802.11e (QoS) enabled and which has two VoIP nodes \((WL1, WL2)\) associated with it along with a traffic generator/sink. Each VoIP session runs between a wired node and a wireless node \((W1<===>WL1, \text{ and } W2<===>WL2)\). The technical implementation details for the test-bed are described in Chapter 4. The remainder of this chapter deals with the conceptual aspects of how the test-bed operates and, in particular, the architecture of the iAP and its key components including the novel delay calculation module and a VoIP prioritization mechanism.

![Figure 3.1 Test-bed Architecture](image)

3.2.1 Intelligent Access Point – iAP

At the core of the real world test-bed is an intelligent, QoS enabled, Wi-Fi AP (iAP). The iAP can dynamically and in real-time, perform the following functions at fixed intervals:

1. Identify individual VoIP sessions
2. Calculate one-way delay for each session, including wired/wireless components of the delay, termed intra-one-way delay

3. Calculate each-way QoS R-values for each sessions (based on one-way delays)

4. Run a prioritization algorithm for VoIP sessions (based on R-values)

5. Implement prioritization on AP downlink amongst sessions via modification of the IP DSCP field which is interpreted appropriately at the MAC layer

The functions numbered 2 and 3 above operate on both the uplink and downlink direction of each VoIP session. The implementation of prioritization operates on the downlink only, for reasons explained in 2.3.5. Unless otherwise stated, the ‘downlink’ is defined hereafter as data travelling from a wired client towards a wireless client via the AP, and ‘uplink’ is defined as data travelling from a wireless client towards wired client via the AP.

The design of the modular architecture of the iAP is illustrated in Figure 3.2. As data packets enter the AP, the packet capture application [78] scans for RTCP SR/RR packets. When RTCP packets are found, the AP timestamp is noted and they are passed up to a Delay Calculation module which identifies unique VoIP sessions and calculates the intra-one-way, and one-way delays for each session (up/downlink). These delay values are then passed to the QoS module which generates an R-factor score for all sessions. The R-factor value derived for each session is then passed to the Prioritization module which runs a prioritization algorithm to decide which, if any VoIP sessions should be prioritized.
If a VoIP session is to be prioritized, the assigned ID of the session is passed to the Traffic Re-classification module which modifies (or mangles) the DSCP value in the IP header of VoIP data packets to appropriately prioritize the session in question. Technical implementation details of the iAP setup are provided in Chapter 4.

### 3.2.2 Multiple SDN APs

While a full SDN implementation of the iAP is outside the scope of this thesis, it is envisaged that a future deployment of the test-bed will involve a centrally controlled mesh network linking multiple individual 802.11 BSS networks, as illustrated in Figure 3.3. Although the small-scale test-bed used in the thesis demonstrates the feasibility of the approach over a simple network, the figure below outlines how it can be extended to more complex mesh WiFi networks and is scalable. Figure 3.3 visualizes how, in the more scalable implementation, the intelligence within the iAP is now removed to a remote station that manages a full mesh of APs that are thus demotes from iAPs to slave APs, signifying the SDN approach. Regarding scalability concerns, whilst the extent of mangling can be significant for APs that manage multiple concurrent RTC flows, the processing latencies are negligible using
conventional hardware, relative to the Mouth to Ear (M2E) latencies being considered.

Furthermore, the scalability of the delay calculation mechanism using RTCP was examined. Essentially, the sniffing and analysis of RTCP traffic is undertaken at the first/last hop of our small scale network (a single WiFi AP). However the mechanism described may operate on a full mesh network as long as the following two conditions are met; 1 – all session endpoints and intermediate iAPs are synchronized, and 2 – the session traffic is transported using RTP along with RTCP control. RTCP traffic consumes a very small amount of bandwidth per session, and is \( O(N) \) where \( N \) is the number of concurrent sessions. This approach is further discussed in section 7.2.3 under Future Work.
3.3 QoS Estimation Methodology

The Quality Estimation aspect of the iAP is composed of both a Delay Calculator and a QoS Estimator module. These modules identify and analyse voice traffic that traverses the AP, determine delays, and then estimate QoS scores for each VoIP session. This section presents details of the novel delay calculation mechanism that uses RTCP Sender and Receiver Reports (SR/RR) along with AP timestamps to calculate intra-one-way, or intra-Mouth to Ear (iM2E) delays for VoIP sessions. Time synchronization to millisecond accuracy is required on the iAP and on both endpoints of VoIP sessions in order to calculate accurate delays. Testing of the Delay Calculator and QoS estimator modules is carried out on the proof-of-concept experimental test-bed, results for which are described in Chapter 6.

3.3.1 One-Way Delay over Wi-Fi

According to ITU-T recommendation G.114 [32], the acceptable limit of one-way delay for VoIP applications is approximately 150ms. However, as detailed in [81], and [33], with good echo controls this can be further relaxed, especially for social calls.

![Figure 3.4 Wired / Wireless Network Delay (WND/WLND)](image-url)
The one-way M2E delay for simultaneous VoIP calls through a single access point can vary greatly for a variety of reasons. For example, for geographically local calls, the non-wireless network delays or *Wired Network Delays* (*WNDs*, Figure 3.4) are often <10ms, while other calls may be over long distances with *WND* >100ms (Figure 3.5). With large network delays, due to geographical distance or network problems, the total one-way M2E delay including send, access, propagation, and receive time of packets as detailed in section 2.4.2, could be close to, or may exceed the ITU-T G.114 recommendation of 150ms.

![Figure 3.5 Differing One-Way Delays](image)

A key objective of this research was to develop and implement a prioritization mechanism at the wireless MAC layer that would attempt, where feasible, to equalise the total M2E delays for VoIP sessions as illustrated in Figure 3.6. Figure 3.6(A) illustrates a hypothetical scenario where there is no prioritization implemented between VoIP sessions. In particular, VoIP sessions *S1* and *Sn* experience one-way M2E delays resulting from large geographical distances between the call participants, and that are above the ITU-T recommended acceptable limit for one-way delay. Figure 3.6(B) shows an improved scenario where sessions *S1* and *Sn* have been prioritized at the wireless MAC layer, which then decreases their wireless network delay (*WLND*) and brings the total M2E delay to within the G.114 threshold. The
remaining sessions ($S2 - Sn-1$) have experienced an increase in their wireless network delay, which contributes to an increase in their overall one-way M2E delay. However, their overall one-way M2E delay, are still below the acceptable limit threshold ($G.114$).

In the current 802.11e QoS implementation, all VoIP sessions within the 802.11e EDCA voice category will have similar delays at the wireless MAC layer due to the equal statistical probability that any STA will obtain access to the wireless medium via CSMA. If the precise one-way delay information for each VoIP session is known in both directions, EDCA parameters can be configured differently between VoIP sessions in order to optimise the QoS (Figure 3.6(B)), so that the sessions with the higher one-way M2E delays receive higher priority treatment at the wireless MAC level relative to other VoIP sessions with lower M2E delays. Essentially this is done by prioritizing voice sessions within the voice access category. For a session that has a small one-way delay and, by extension a high R-factor, packets can afford to wait for a longer time at the wireless MAC layer, on the condition that they are delivered within a period (where their one-way M2E delay doesn’t exceed the acceptable limit threshold) that will not degrade their QoS significantly. On the other hand, a VoIP session that has a large M2E delay will benefit from a shorter contention delay at the wireless MAC layer, thus reducing the overall one-way M2E delay for that voice session (Figure 3.6(B)).
An extension of this theory is to apply this logic to encapsulate QoS by using the E-Model (Figure 3.7). Figure 3.7 illustrates an extension of the hypothetical scenario depicted in Figure 3.6, however here the delay values are factored into the E-Model to produce an R-factor score for each session ($S1 - Sn$). Figure 3.7(A) illustrates the corresponding R-values for Figure 3.6(A) where there is no prioritization implemented. Sessions $S1$ and $Sn$ have QoS scores that are below the E-Model threshold for good QoS of $80R$ (E-Model “Good QoS” Threshold). Figure 3.7(B) illustrates the corresponding R-values for the scenario in Figure 3.6(B) where sessions ($S1, Sn$) are prioritized at the expense of the remaining sessions ($S2 - Sn-1$).
R-factor equalization provides a framework within which to pursue a net quantifiable improvement in the quality of multimedia applications over wireless networks using synchronized time. To prove the hypothesis, this work set out to quantify and qualify QoS improvements via both simulation on NS-2, NS-3, and via empirical experimental results on a real-world test-bed (iAP) which would validate simulations and demonstrate that the approach is technically feasible.

3.3.2 Delay Calculation Mechanism

In order for an AP to achieve delay and R-factor optimization, it must calculate accurate one-way delay values for each VoIP session. Working under the assumption that VoIP session endpoints and the AP itself are synchronized\(^6\) to a common timescale such as UTC with millisecond accuracy, the AP can calculate one-way delay for each session as well as intra-one-way delays (iM2E). This is illustrated in Figure 3.8 whereby iM2E has two components; 1: the delay between an endpoint and the AP,

---

\(^6\) AP synchronization is not necessarily required for accurate one-way M2E delay calculation in the mechanism that is proposed. This extension to the process is detailed in section 4.3.1.
and 2: the delay between the AP and the other endpoint. It is important to note that the delays are determined using the RTCP transport layer timestamps and thus include contention delays at the MAC layer. Note also that packets on the AP are captured on the wireless interface as shown in Figure 3.2, so there are undoubtedly some AP delays in the calculations. However, these are relatively negligible in the context of MAC contention and network delays so they are disregarded.

![Diagram](image)

Figure 3.8 Intra-One-Way Delay

By calculating these delays, the AP can determine a complete picture of the VoIP delays within a BSS. This process is based on the capture of RTCP Sender Report and Receiver Report (SR/RR) packets in the uplink and downlink direction. Information within these packets is then combined with the timestamps of those packets arriving on the AP. The values required from the RTCP packets for calculation are the NTP timestamp (MSW and LSW) and the Delay since Last SR (DLSR). The RTP/RTCP protocol and these timestamps are described in detail in section 2.1.1.2.

In order to calculate the necessary one-way delays for each session, seven time values and three RTCP packets must be obtained which can then be used to deduce the intra-one-way delays for the downlink and the uplink, Figure 3.9 illustrates the process. The seven values are labelled $T_i$ to $T_7$. 

- 109 -
Where a VoIP session takes place between wired node (N1) and wireless node (N2) via an AP, the AP must gather a sequence of three consecutive RTCP SR/RR packets (labelled in Figure 3.9 as RTCPsr1, RTCPrr1, and RTCPrr2) relating to a particular VoIP session in order to calculate all seven time values $T_1, T_2,..T_7$. These time values are used to calculate two downlink $iM2E$ delays ($dn_1$, $dn_2$), and two uplink $iM2E$ delays ($up_1$, $up_2$) so that they can be combined to provide downlink ($\Delta_d$), and uplink ($\Delta_u$) one-way ($M2E$) delays. The time value $T_7$ is required to calculate $T_6$, but is not itself directly used to calculate $iM2E$ delays. The first $iM2E$ delay $dn_1$ is calculated as follows:

![Figure 3.9 Delay Calculation](image-url)
where $T_1$ is the time that the packet $RTCPsr1$ is sent by $N1$, and is extracted from the $NTP$ timestamp field in the $SR$ packet header as described in section 2.1.1.2.1. The time value $T_2$ is the time when $RTCPsr1$ arrives at the wireless interface in the AP as determined by the capture process. This is the timestamp provided by tcpdump (via the PCAP interface). The second $iM2E$ value ($dn_2$) is calculated by:

$$dn_2 = (T_3 - T_2)$$ (3.2)

where $T_3$ is the time $RTCPsr1$ arrives at $N2$. The value for $T_3$ is calculated as:

$$T_3 = (T_4 - DLSR_{r1})$$ (3.3)

where $T_4$ is extracted from the $NTP$ timestamp field of the packet $RTCPrr1$. The value for $DLSR_{r1}$ is extracted from the Delay since Last SR ($DLSR$) field in the $RTCPrr1$ packet (Figure 3.9). The $iM2E$ delay $up_1$ is calculated as:

$$up_1 = (T_5 - T_4)$$ (3.4)

with $T_4$ the time that the packet $RTCPrr1$ is generated for departure from $N2$, and is extracted from the $NTP$ timestamp field in the $RR$ packet header. The time value $T_5$ is the time when $RTCPrr1$ arrives at the AP via the capture process. The value for $iM2E$ delay $up_2$ is calculated as:

$$up_2 = (T_6 - T_5)$$ (3.5)
where $T_3$ is the time the packet RTCPrr1 arrives at the wireless interface in the AP, this is the same value that is used in the previous equation. The value for $T_6$ is calculated as follows:

$$T_6 = (T_7 - DLSR_{rr2})$$  (3.6)

where $T_7$ is extracted from the NTP timestamp field of the packet RTCPrr2. The value for $DLSR_{rr2}$ is taken from the RTCPrr2 packet. $T_7$ is not used directly in any of the iM2E delay calculations, however, it is illustrated to aid in the explanation of the calculation of $T_6$, which is required for the calculation of $up_2$. Figure 3.10 illustrates the RTCP header values that are used in the calculation along with their source packets.

Figure 3.10 Values obtained from RTCP packets
Once the intra-mouth-to-ear delays (iM2E) are calculated, the downlink (Δ_d) and uplink (Δ_u) mouth-to-ear delays (M2E) can be calculated for each session as using equations:

\[
\Delta_d = (dn_1 + dn_2) \quad (3.7)
\]

\[
\Delta_u = (up_1 + up_2) \quad (3.8)
\]

This process is repeated for all concurrent VoIP sessions. Once one-way delays are calculated, the values are then passed on to the QoS module in the iAP to calculate an R-factor for each session. Default values for the E-Model are used where appropriate as described in [12]. This delay calculation mechanism is used in all experimental results. The RTCP interval as described in section 2.1.1.2.1 regulates how often RTCP packets are sent by an application. This interval therefore governs how regularly delays are calculated for each session. According to relevant RFCs, it is usually every few seconds so that in a typical VoIP call for example, it would allow delay calculation multiple times so that optimization can be carried out multiple times.

### 3.3.2.1 Time Synchronization Challenges

As mentioned in section 2.4, network time synchronization in wireless networks presents additional challenges compared to that experienced in wired networks. As outlined by Shannon in [82], the same AP bottleneck that disrupts data over WiFi seriously degrades synchronization protocol performance by introducing large delay asymmetries. However, he then outlines a mechanism that determines asymmetry in the AP in real-time and thus delivers synchronization levels equivalent to wired networks. His research, also at PEL NUI Galway, is currently patent pending. In the absence of such a mechanism, synchronization levels can degrade to the order of tens to hundreds of milliseconds and even to seconds over time which would undoubtedly result in very inaccurate delay measurements, rendering the approach outlined in this
thesis unfeasible. The successful operation of the experimental test-bed requires the implementation of time synchronization to within single figure millisecond accuracy across the full test-bed. Delay values cannot be determined via the RTCP packet mechanism unless synchronization is robustly implemented. In order to verify that all the nodes in the test-bed are sufficiently synchronized before each experiment, a separate synchronization test was designed and is detailed in section 4.3. It is also worth noting that significant efforts have been taking place within the IEEE under the umbrella of Time Sensitive Networking (TSN) to improve synchronization over WiFi networks in the medium to long term.

3.4 Traffic Prioritization

Once the R-factors have been calculated, a prioritization algorithm decides which, if any, sessions will be prioritized by the AP. Any changes in priority once decided upon, are implemented by the traffic re-classification module. As mentioned in Chapter 2, much research, including work in [83], [84], and [85], has been undertaken to investigate the tuning of EDCA parameters for QoS in inter/intra-traffic type scenarios. Furthermore, in [86] and [87], examples of adaptive approaches are considered for each STA. For the iAP Proof-of-Concept (PoC), a hybrid approach that sits between a static and an adaptive approach is used with an EDCA MAC parameter tuning mechanism. VoIP sessions may switch between three pre-defined MCs via a prioritization algorithm (dynamic) whereas MAC parameters within each MC are pre-defined (static).

3.4.1 Intra-AC Multimedia Categorization

In order to implement prioritization for concurrent VoIP sessions, a three-tiered categorization mechanism is used in the PoC whereby all sessions initially reside in the lowest category, and thereafter, certain sessions may be promoted (prioritized) to a higher priority category or demoted (de-prioritized) where possible, based on their QoS score. A background traffic category also exists in order to cater for non-VoIP traffic. The three Multimedia Categories are termed $MC_1$, $MC_2$; and $MC_3$, where $MC_1$ has the highest priority and $MC_3$ has the lowest priority, along with a background access category ($BK$). All VoIP traffic resides within the three MCs. These four categories were chosen so that existing 802.11e equipment could be used in the PoC.
as 802.11e implements a four category system. The term ‘multimedia traffic’ is used to account for the possibility of accommodating other RTC traffic types as opposed to just VoIP traffic. At the beginning of a VoIP session, all traffic is allocated to $MC_3$ on first access to the iAP. When an R-factor is calculated, a prioritization algorithm may promote sessions to $MC_2$ or $MC_1$ depending on their R-factor score. This provides two higher priority categories that a VoIP session may be promoted to.

As mentioned in section 3.3.1, the iAP aims to optimize the R-factors of multiple active VoIP sessions, thus the prioritization algorithm includes a facility for VoIP sessions to be demoted to a lower category if all sessions have a good QoS. This feature is only activated when all VoIP calls have an R-factor that is above 90. Session demotion is intended to restore the prioritization mechanism to its default state after a scenario where sessions occupy $MC_1$ due to contention with VoIP sessions that have since finished.

Due to the well-recognised downlink traffic bottleneck at the AP, as explained in section 2.3.5, this method operates only on the downlink, whereby packets entering the iAP and heading to the wireless interface are re-prioritized by tuning the DSCP value in the IP header in order to facilitate EDCA. Each MC has a set of 802.11e EDCA parameters that are chosen heuristically, and are designed to provide a tiered level of service. This process is illustrated in Figure 3.11.
The EDCA parameters within each category are heuristically chosen such that they generate increased probability of packets accessing the wireless medium in one MC over another MC. These parameters are based on the analysis of research summarized in section 2.3.5 and are outlined in TABLE VIII. The values for $CW_{\text{min}}$ and $CW_{\text{max}}$ are dependent on the physical layer protocol used (802.11b) as described in [8].
TABLE VIII MULTIMEDIA AC PARAMETERS

<table>
<thead>
<tr>
<th>AC</th>
<th>AIFS</th>
<th>$CW_{\text{min}}$</th>
<th>$CW_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Multimedia1: MC₃)</td>
<td>2</td>
<td>$(CW_{\text{min}}+1) / 4 - 1$</td>
<td>$(CW_{\text{min}}+1) / 2$</td>
</tr>
<tr>
<td>2 (Multimedia2: MC₂)</td>
<td>2</td>
<td>$(CW_{\text{min}}+1) / 4$</td>
<td>$(CW_{\text{min}}+1) / 2$</td>
</tr>
<tr>
<td>3 (Multimedia3: MC₁)</td>
<td>3</td>
<td>$(CW_{\text{min}}+1) / 2$</td>
<td>$CW_{\text{max}}$</td>
</tr>
<tr>
<td>4 (Background: BK)</td>
<td>7</td>
<td>$CW_{\text{max}}$</td>
<td>1023</td>
</tr>
</tbody>
</table>

By default, all VoIP sessions at the start are assigned to the access category MC₃. It is assumed that numerous sessions will simultaneously occupy MC₃ while a smaller number of sessions will be subsequently prioritized to categories MC₂ and MC₁. The CW values listed in TABLE VIII are taken from the EDCA AC_VO access category with the values ($CW_{\text{min}}=7$, $CW_{\text{max}}=15$). When an EDCA function (EDCAF) wants to transmit a packet it waits for the channel to be idle for a time that is equal to the AIFS parameter.

As described in detail in section 2.3.2, if the channel is sensed to be busy after the initial AIFS time, the EDCAF begins the back-off process and selects a back-off counter from the range [0-$CW_{\text{min}}$]. When the counter reaches zero, the EDCAF then transmits the packet. If another STA transmits a packet at the same time, a collision occurs and both packets must select a new random back-off counter from the range [$CW_{\text{min}} - CW_{\text{max}}$], and the retry counter is also incremented by one (up to a retry limit). This process is described in more detail in the IEEE 802.11e standard specification [8]. Packets assigned to the MC₁ category select their back-off counter from a range of [0-1]. This range is lower than MC₂ where the back-off counter is selected from a range of [0-2].

The $CW_{\text{min}}$ for MC₁ ($MC₁_{CWmin}$) is 1 as opposed to 2 for MC₂. The $CW_{\text{max}}$ is the same for both MC₁ and MC₂. The MC₃ parameters $MC₃_{CWmin}$ and $MC₃_{CWmax}$ are set to 4 and 15 respectively. This CW range for MC₃ is higher and wider than the CW range for MC₁ or MC₂ as it is the lowest category. Allied to this is the fact that collisions can occur more frequently when an increased number of STAs occupy high priority access categories that have a narrow CW range [88], [89]. This can increase the number of internal collisions, which in turn causes the doubling of the back-off counter and can lead to delay and packet loss when the retry limit is reached.
and the packet awaiting transmission is dropped. The background traffic category (BK) parameters are chosen with the aim of supporting any non-real-time data traffic that is present in a network. The AIFSN and CW\textsubscript{max} values are the same as the default 802.11e parameters, while the CW\textsubscript{max} is equal to the CW\textsubscript{max} of MC\textsubscript{3}.

These PoC parameters are chosen heuristically based on analysis of research in the literature along with preliminary simulations; however, they are not necessarily optimal parameters. As mentioned in section 2.3.5, the pursuit of the optimal EDCA parameter configuration is an ongoing question in the research community with quality improvements sought using methods ranging from dynamic optimal parameter configuration to game theoretical methods. These parameters are simply used as an example to illustrate the effectiveness of the PoC approach whereby synchronized time is used to precisely determine delays which informs the QoS mechanism that can prioritize some VoIP sessions over others based on R-factor values.

### 3.4.2 Traffic Prioritization

When the QoS Estimation routine is initiated in the iAP, RTCP SR and RR packets are continually captured using the pcap interface. The source and destination IP addresses of these RTCP packets are filtered and added to an array which maintains a reference of all active VoIP sessions that traverse the iAP. When the QoS R-factor score has been calculated for each VoIP session, the next phase of the process is the design of an algorithm that assesses whether or not that VoIP session should be prioritized or de-prioritized. The RTCP packet generation interval is described in section 2.1.1.2.1. Once three consecutive RTCP packets are received for a particular VoIP session, an R-Value is calculated for that session as described in section 3.3. Once the R-factor is calculated for all sessions, the array containing the R-factors for all VoIP sessions (CallArray[]) is passed to the prioritization algorithm, the design of which is described below.
Algorithm 1

1. loop
2. R=FindLowestR(CallArray[])
3. if R < 50 then
4.   exit loop
5. else if R < 80 then
6.   if currentMC = MC3 || currentMC = MC2 then
7.     promote session
8.   else if currentMC = MC1 then
9.     exit loop
10. end if
11. else if R > 90 then
12.   if currentMC = MC1 || currentMC = MC2 then
13.     demote session
14. else
15.   exit loop
16. end if
17. end if
18. end loop

The prioritization algorithm identifies the VoIP session with the lowest R-factor using the $\text{FindLowestR()}$ function. The algorithm operates on one session at a time. If a VoIP session has an R-factor that is below 50 it is deemed to be experiencing QoS that is too low to qualify for prioritization at the MAC level (according to the E-Model). It is highly unlikely that this mechanism will have any significant positive impact on its QoS score and therefore is deemed not worth trying to improve. If the lowest R-factor is below 80 and the session currently resides within the $MC_3$ or $MC_2$ category, then that session is promoted to the next highest MC ($MC_2$ or $MC_1$). If the session with the lowest R-factor is already in $MC_1$, then no action is taken. In a scenario where the lowest R-factor is above 90, and the session currently resides within $MC_1$ or $MC_2$, then that session is demoted to the next lowest category ($MC_2$ or $MC_3$).
This is intended to provide functionality to restore VoIP sessions that have good QoS to the default category, MC3, in order to improve fairness and facilitate prioritization should any new VoIP sessions associate with the iAP at a later point in time.

As outlined, the design approach taken here is best described as a hybrid approach between a static prioritization mechanism and an adaptive EDCA MAC parameter tuning mechanism. VoIP sessions may switch between any of the three prescribed MCs via the prioritization algorithm however MAC parameters within each MC are pre-defined. It must be stated that this algorithm has not undergone rigorous evaluation, as the overall objective is to develop a proof-of-concept. In conclusion, as rationalized in section 2.3.5, this mechanism works on the downlink only, however, some suggestions for uplink prioritization are provided in the conclusion of this thesis.

3.4.3 Traffic Re-Classification

When a VoIP session is chosen for prioritization or de-prioritization, the iAP needs to implement this quickly. The design approach utilizes the Netfilter framework to modify the Diffserv Code Point (DSCP) value in the IP header of all downlink packets (RTP) belonging to the session. As described in Chapter 2, a mapping exists between DSCP values and the 802.11e traffic categories, and this process is utilized in the design to implement re-classification between $MC_1$ – $MC_3$. This has the effect of ensuring that all voice packets for a session will follow the categorization framework described in the previous section. The implementation of this process is detailed in Chapter 4.

3.5 Summary

This chapter has presented the reader with an overview of the design stages relating to both the simulation and test-bed approaches. In particular, it focuses on the design of the operation of the intelligent Access Point (iAP) and its core components. Its design can be broadly broken down into two categories, QoS estimation, and Traffic Prioritization. These combine to calculate QoS scores for multiple VoIP sessions, and then prioritize certain VoIP sessions (if required) over others based on their QoS scores. Four core modules are required to implement the iAPs functionality; QoS Estimation is composed of a Delay Calculator and a QoS Calculator module, while
Traffic Prioritization is composed of a *Prioritization Algorithm* and a *Traffic Re-classification* module.

Once QoS scores, in the form of *E-Model R-factor* scores, are calculated, a prioritization algorithm identifies which, if any, VoIP sessions should be prioritized over others using a *Multimedia Categorization (MC)* classification system. The MC system provides three multimedia categories that allow VoIP sessions to be promoted (prioritized) and demoted (de-prioritized) by moving up and down the three levels of the classification system. There also exists a category for background (non-real-time multimedia) traffic. The MC prioritization is supported by *802.11e EDCA MAC* level parameters which control data traffic access to the wireless medium. Implementation details for the simulation and test-bed approaches are considered in the next chapter.
Chapter 4

IMPLEMENTATION

The previous chapter described the main design issues relating to both the simulation and test-bed approach. It focused more so on the system architecture of the iAP, which is at the core of the experimental test-bed. It outlined the design of the core QoS Estimation and Traffic Prioritization modules that are central to the operation of the iAP. This chapter provides implementation details relating to both approaches that describe the various tests undertaken to answer each of the research questions from Chapter 1. In relation to the first research question, a number of preliminary simulation tests are carried out using the NS-2 simulator. Simulation Test 1 evaluates the capacity of VoIP over 802.11 wireless networks, while Simulation Tests 2 and 3 evaluate both the feasibility of implementing prioritization amongst VoIP sessions over 802.11e networks, and also the effectiveness of EDCA parameter optimization. It then describes the subsequent NS-3 simulation setup of Simulation Test 4 that evaluates the Traffic Prioritization mechanism as detailed in the previous chapter. In relation to the second research question posed in Chapter 1, Section 4.2 details the implementation issues relating to the test-bed and its constituent components. Additionally, this chapter describes details of the implementation of time synchronization in the experimental test-bed in section 4.3.
4.1 Simulation Setup

4.1.1 Preliminary Simulations - NS-2

As part of the first research question addressed in this thesis, NS-2 provides the facility to simulate large scale networks involving many nodes and parameters in a timely manner. The relatively low configuration time is useful when setting up network simulations with multiple VoIP nodes communicating between wired and wireless networks. The NS-2 WLAN EDCF model is detailed in [90]. The results of all simulation tests are presented in Chapter 5.

4.1.1.1 NS-2: IEEE 802.11e VoIP Capacity Evaluation (Simulation Test 1)

The first simulation test examines the network capacity for VoIP in an 802.11e enabled network in terms of QoS when delay and packet loss are taken into account. Each VoIP session has the same priority and is set up between a wireless and a wired node via an AP (Figure 4.1). Simulations are carried out on the widely used 802.11b network without contention from other types of traffic. 802.11e is modelled via the TU Berlin model [90] and the VoIP packet generation at the application layer is distributed evenly by initiating VoIP sessions at different times to ensure an initial low collision probability. Whilst WiFi bandwidths have increased significantly in recent years beyond that used in the test-bed, the viability of the underlying approach remains intact as application bandwidth demand will always expand to meet network capacity and thus a need will always exist to prioritize certain delay sensitive traffic. This is to ensure that all VoIP sessions do not begin transmitting packets at precisely the same moment, because such a scenario would cause a large number of collisions at the beginning of the simulation leading to buffer overflow and thus cause unrealistic packet loss. This phenomenon is described in more detail by Shin et al. in [58]. With regards to the capacity of voice calls that can run concurrently in an 802.11 infrastructure network while maintaining an acceptable level of QoS, the number can vary depending on technical factors such as network transmission rate, voice codec used, and the extent of background traffic present.
For this test all VoIP calls reside within the 802.11e EDCA AC_VO access category using the default EDCA parameters thus all sessions have equal priority. This simulation measures the mean session delay and packet loss rate for all VoIP sessions. One VoIP call is activated when the simulation begins, then subsequent sessions are added to the network at 30 second intervals up until there are 20 active VoIP sessions. The G.711 codec (64kbps) is used with a 20ms frame interval and thus 160 bytes RTP payload. On the 802.11b physical layer the frequency is 2.472GHz, and there are no hidden terminals. The results for this test are provided in section 5.1.

4.1.1.2 NS-2: Effects of Differing One-Way Delay on VoIP (Simulation Test 2)

As the overall goal of the thesis is to evaluate QoS benefits over wireless networks using synchronized time to precisely measure delays, the impact of delay on VoIP was examined. The next tests carried out as part of the first core research question are to simulate differing delays among VoIP sessions that represent different geographical distances between endpoints. This was implemented by setting different individual wired network delays (WNDs) in the wired network (illustrated in Figure 4.2 as 100ms and 50ms for sessions S1 and S2 respectively). Continuing on from the previous simulation test in Test 1 with the same VoIP call setup, this time 12 VoIP sessions were run in an 802.11e network based on the results of the preliminary simulations detailed in 4.1.1.1. The goal of this simulation is to evaluate whether it is
possible to tune 802.11e EDCA parameters based on M2E delays so as to prioritize those VoIP sessions that have relatively high delays. As such, designing the test with 12 sessions ensures that the overall wireless capacity is reaching saturation thus ensuring that delays at MAC level will be significant in the context of VoIP interactivity requirements and thus will help highlight the benefits of the proposed approach. The simulation topology is illustrated in Figure 4.2.

![Figure 4.2 NS-2 Test 2 Simulation Topology – WNDs of 100ms and 50ms (S1 & S2)](image)

For these simulations, 12 simultaneous voice sessions are run between endpoints on an 802.11e enabled 802.11b network and endpoints on a wired network. These tests have a duration of 160 seconds. One session (S1) experiences a WND of 100ms, while another session (S2) experienced a WND of 50ms, and the remaining 10 sessions (S3-S12) experience a small pre-set one-way delay (WND) of 5ms. The reasoning behind running this test is to illustrate the effect that different one-way delays can have on a VoIP session in both the uplink and downlink direction. There are three scenarios carried out; in the first scenario (Test 2.1) all of the VoIP sessions have equal priority. In the second scenario (Test 2.2), S1 and S2 have prioritized EDCA settings on the uplink, and in the third scenario (Test 2.3), S1 and S2 settings are prioritized on both the uplink and the downlink. The results for this test are provided in section 5.2.
4.1.1.3 **NS-2: Preliminary EDCA Parameter Optimization (Simulation Test 3)**

The third simulation tests builds on *Test 2*, again evaluating the potential to improve the QoS of VoIP by tuning 802.11e EDCA MAC parameters as part of the first research question. However, this time the M2E delays calculated are plugged into the E-Model, producing R-factors for each session, therefore the tuning of EDCA parameters for VoIP calls is based on R-factor values. The topology for this simulation consists of running 13 simultaneous VoIP calls over an 802.11e enabled 802.11b network as illustrated in Figure 4.3.

Two scenarios are evaluated for *Test 3*. The first scenario *Test 3.1* was carried out with all sessions existing within the same AC and thus with the same priority, while in the second scenario, *Test 3.2*, the session S1 is prioritized on the uplink and the downlink over all of the remaining sessions.

![Figure 4.3 NS-2 Test 3 Simulation Topology – WND of 100ms S1](image)

For both scenarios, the first session (S1) has a pre-set WND of 100ms as illustrated in Figure 4.3, this is to simulate a longer geographical delay than the remaining sessions and thus cause S1 to have a lower QoS that might be improved by tuning EDCA parameters. The pre-set WND added to sessions 2 – 12 was increased from 5ms in *Test 2* to 20ms in *Test 3* in order to more accurately reflect the packetization...
delay of the G.711 codec for 160 byte payload, which was not considered in tests so far but which is a significant contributor to overall M2E delay.

As mentioned above, in the first scenario (Test 3.1) all VoIP sessions have equal priority as they all exist within the same AC. For the second scenario, Test 3.2, S1 is given priority by incrementing the AIFSN of the remaining traffic by 1 unit to 3, where the prioritized session (S1) maintains an AIFSN value of 2. All sessions maintain the default CWmin/max values of 7 and 15 respectively. Obtaining a lower AIFSN value than competing STAs provides some priority to S1 in 802.11e. The results for this test are discussed in section 5.3.

4.1.2 iAP Implementation - NS-3

NS-2 was used to preliminarily evaluate various aspects relating to network capacity in the EDCA AC_VO access category, and also the feasibility of EDCA parameter optimization in improving VoIP QoS. However, the NS-3 simulator was released during the course of this research and it provides an improved structure, better documentation, and a framework to run a more thorough set of simulations. Therefore it was decided to take advantage of these features and change to NS-3 for a range of further tests as described below.

4.1.2.1 NS-3: Multimedia Category Evaluation (Simulation Test 4)

In order to more fully evaluate the aforementioned prioritization mechanism, simulations were run where multiple VoIP sessions were deployed, each of them with differing WNDs. The EDCA parameters used are listed in TABLE VIII and represent the three separate ACs (MC1, MC2, and MC3). As mentioned in Chapter 3, much research has been carried out on the pursuit of optimal values for EDCA parameters under many different sets of traffic characteristics. The capacity of a WLAN for voice calls varies depending on many variables such as voice sampling period, codec, MAC layer and PHY protocol settings.

As part of the first core research question, and building on all of the preliminary simulations, these simulations (Test 4) were designed to investigate whether, and to what extent the QoS of VoIP sessions that have a large wired network delay can be improved by reducing the wireless access delays of those sessions, without significantly affecting the QoS of remaining sessions. Both one-way delay and packet loss
are factored into the QoS scores for this Test 4. The simulation topology shown in Figure 4.4 implements an infrastructure AP accommodating multiple G.711 VoIP calls, modelled as a Constant Bit-Rate (CBR) 160 Byte VoIP payload, with an RTP/UDP/IP header size of 40Bytes. There is a simultaneous uplink/downlink New-Reno TCP traffic session utilizing the AP with packet size of 1500 Bytes. The 802.11b SIFS value is set to 10μs. At the beginning of the simulation, there are four VoIP sessions (S1, S2, S4, and S6) that have relatively benign (low) WNDs in terms of QoS. There are also four sessions that have large WNDs of above 165ms (S3, S5, S7, S8), with these four sessions are more likely to experience a negative impact on their QoS.

![Simulation Topology](image)

Figure 4.4 Simulation Topology

There are two scenarios simulated. Firstly in Test 4.1 (results in section 5.4.1), a sequence of VoIP calls are run using the default EDCA channel access mechanism. Then, in Test 4.2 (results in section 5.4.2), the same sequence of calls is carried out using the Multimedia Category (MC) mechanism that is described in Chapter 3. At the beginning of the simulations, in both scenarios, six VoIP sessions commence at five second intervals, beginning with session 1 (S1) at T = 0s up to S6 at T = 25s,
thus at $T = 25s$, six sessions are active. The call S6 expires at $T = 55s$ and another call (S7) begins at $T = 65s$. S2 expires at $T=85s$ and session 8 (S8) begins at $T=95s$, this sequence is illustrated in Figure 4.5.

This VoIP call sequence provides an example of a real-world scenario where calls begin and terminate at random intervals, and where the one-way delays can vary. The results for this simulation are provided in section 5.4. The Call Duration (CD) of the VoIP calls in this test are designed to typify a range of call lengths, the CD is thought not to have a correlation with MOS quality scores [91].

4.2 Experimental Test-bed Setup

As outlined in Chapter 3, the experimental test-bed developed in this research consists of a wired LAN connected to a wireless LAN via an 802.11e enabled wireless router (iAP). VoIP sessions are run using the Ekiga softphone application (v.3.3.2) [92] between two Ubuntu Linux clients on the LAN ($W1 \& W2$) and two Linux clients that are associated with an 802.11b network ($WL1 \& WL2$). All calls utilize the PCM - G.711 voice codec which is broadly supported by client software. The Linux network emulator NetEm [93] is used to emulate delays on the wired network side in order to increase delay for VoIP sessions (especially in the downlink direction). Background traffic is generated as required using the D-ITG traffic generator [94]. The iAP runs DD-WRT firmware [95] which is 802.11e enabled and is connected to the wired network via an Ethernet switch. RTCP SR and RR traffic which is adequately supported by the Ekiga softphone is mirrored to the delay esti-
mator via the `libpcap` interface in `TCPDump` [78]. The overall test-bed is illustrated in Figure 4.6.

![Test-bed Architecture](image)

**Figure 4.6 Test-bed Architecture**

Time synchronization is provided via the NUI Galway NTP server, distributed over the wired LAN to wired clients and the iAP, and over the wireless interface to the wireless clients. An Asterisk server is used for call setup with future scalability in mind. IP addresses are distributed by a DHCP server on iAP. A Linux monitoring station is connected to the network which acts as a gateway to the iAP via the terminal (SSH) or web based (HTTP) interface. This monitoring functionality is akin somewhat to a Software Defined Network (SDN) architecture [3], where a node on a network can be dynamically configured by a remote and centralized monitoring node to optimize a certain set of traffic characteristics based on quality or throughput requirements. Two bi-directional VoIP calls take place between wired clients and wireless clients via the iAP, and VoIP session set up is carried out by SIP. The G.711 codec (u-law) is used with a maximum jitter buffer of 50ms. The experimental test-bed runs 32bit Ubuntu 11.10 (Oneiric Ocelot). 

- 130 -
4.2.1 iAP Setup

The wireless AP device used is a Linksys DD-WRT v24-sp2 mega-build. DD-WRT is a Linux based, alternative open source firmware that is used in many different WLAN routers and embedded systems. DD-WRT provides extensive user functionality in both the user and kernel space. This software can be downloaded onto a router in order to provide more configuration options and an extended feature set, particularly on the wireless interface, and provides both a web-based and a command line interface to adjust configuration parameters. DD-WRT allows testing of 802.11e QoS provisions and configuration of EDCA parameters via the Linux Wireless Tools iwpriv command. The CPU is a Broadcom BCM5352 running at 200Mhz with 25.5Mb of memory. The WiFi setup is an 802.11b WLAN connection operating at the default 2.437 Ghz band (channel 6) with a beacon interval of 100ms. Further AP setup details are provided in Appendix III.

4.2.1.1 TCPDump

The TCPDump application on the iAP logs a description of packets captured on the wireless network interface. This description includes a timestamp in seconds since January 1, 1970, 00.00.00, UTC and fractions of a second since that time. TCPDump is configured via command line arguments to filter only packets of interest. An example of a TCPDump command used in the iAP is as follows:

```
tcpdump -nn -s0 -tt -Tr tcp -x port 5061-5099
```

The argument –x allows the data of each packet (minus its link level header information) to be printed in hex. This facility is required in order to overcome the incompatibility of the iAP and NTP timestamps. The NTP timestamp is represented as a 64-bit unsigned fixed point number in seconds relative to January 1, 1900, 00.00.00, whereas the UNIX timestamp is in seconds relative to January 1, 1970, 00.00.00. Therefore a value of 2208988800s is added to the NTP timestamp before performing delay calculations in the QoS estimation module.
4.2.1.2 Prioritization Implementation

As mentioned in the previous design chapter, the VoIP session prioritization mechanism is implemented by editing the DSCP value in IP headers of VoIP packets via the `iptables` command which is part of the Linux traffic control suite. When the iAP is powered up, all VoIP traffic is automatically assigned to the MC3 access category by setting the DSCP value in the IP header of all RTP packets to “10”. A mapping exists between DSCP values and 802.11e categories and this process is utilised for re-classification between MC1-MC3 (+BK). The MCs and their corresponding DSCP values are listed in TABLE IX:

<table>
<thead>
<tr>
<th>EDCA Category</th>
<th>MC Category</th>
<th>DSCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC_VO</td>
<td>MC1</td>
<td>46</td>
</tr>
<tr>
<td>AC_VI</td>
<td>MC2</td>
<td>18</td>
</tr>
<tr>
<td>BE</td>
<td>MC3</td>
<td>10</td>
</tr>
<tr>
<td>BK</td>
<td>BK</td>
<td>2</td>
</tr>
</tbody>
</table>

TABLE IX EDCA & MULTIMEDIA CATEGORY MAPPING TO DSCP VALUES

In a situation where the prioritization algorithm decides that a VoIP session should be prioritized, the DSCP value in the IP header of all packets belonging to that session will be tuned (or ‘mangled’) to the corresponding DSCP value of the next highest priority category. All VoIP packets that are forwarded by the iAP, according to the iptables framework described in section 2.5, are passed to the FORWARD chain. Packets that are to be prioritized by the Prioritization Module in the iAP are processed by a ‘–mangle’ operation. This `iptables` command is performed on their DSCP value in the IP headers before being sent to their destination via the output interface, whereas packets that are not to be prioritized are sent on to their destination via the output interface without being processed by the –mangle operation.

In a hypothetical scenario where one VoIP call (S1) takes place between a wired client `W1` and a wireless client `WL1` as illustrated in Figure 4.6, and another call (S2)
takes place between wired client W2 and wireless client WL2, a mangle operation could implement prioritization as follows:

```
ssh root@192.168.1.1 \"echo admin | iptables -t mangle -I FORWARD -d DEST_IP -j DSCP --
set-dscp DSCP\";
```

In this scenario, the monitor machine logs into the iAP via SSH and issues the `iptables` mangle command (`-t mangle`). This operation will edit the DSCP value in all of the IP packets in the `FORWARD` chain that are destined to the IP address `-d DEST_IP`. These packets are destined to the client WL1 on the wireless network, and are associated with the VoIP call that is being prioritized (S1). This operation is also possible in the opposite direction where the `iptables` command would also contain the IP address of W1, which is the source of the packets (`-s SRC_IP`). Testing of the `iptables` mangling process is carried out by analysis of IP packets arriving at destination client machines using the Wireshark packet capture application [96]. When packets have been prioritized, the DSCP value observed in their IP header on the destination client is different to its initial value.

### 4.2.2 iAP Operation Routine

At the beginning of an experiment, when the iAP is powered up, the QoS Estimation, and Traffic Prioritization modules are designed to run continuously. When VoIP calls begin, TCPDump will filter out the RTCP packets to be processed. A struct is defined to store the following data for each call:

- char *Wired_IP_address
- char *Wireless_IP_address
  - Hardcoded in Experimental results in Chapter 6
- char *RTCP_packets_captured
- int *Number_of_RTCP_packets_captured
- double *Current_R_value
- char *Current_MC
This struct maintains all of the required information associated with each VoIP call for the duration of an experiment.

The determination of uplink and downlink traffic is important in the operation of the iAP, however, for experiments, the IP addresses of the wired and wireless endpoints of the VoIP calls are hardcoded. The RTCP packet data for three consecutive packets is temporarily stored for each session along with the total number of RTCP packets captured for that session. The current R-value for a session is stored along with the current MC category that a VoIP session resides in. The sequence of events for iAP is as follows:

```
iAP Event Sequence
1. while (true) {
2.     packets=run tcpdump for RTCP packets
3.     find wired & wireless IPs of packets (Hardcoded)
4.     if IPs not in IP_array
5.         add IPs to call array
6.     add RTCP packets to appropriate calls
7.     if 3 packets are stored for call
8.         for each session
9.             calculate R
10.        end for
11.    end if
12.    run Prioritization Algorithm
```

Once an R-value has been calculated for a VoIP session, the prioritization algorithm is initiated, which may in turn initiate the prioritization implementation mechanism.

### 4.3 Time Synchronization Implementation

A key requirement for the successful operation of the experimental test-bed is the implementation of time synchronization to within single figure millisecond accuracy across the full test-bed. Without this level of accuracy, delay values determined via
the RTCP packet mechanism will add significant error margins to calculations, thus providing inaccurate QoS estimation. An early challenge encountered was the clock skew between the iAP and the client endpoints. In order to verify that all the nodes in the test-bed are sufficiently synchronized before each experiment, a separate synchronization test was designed. This involves sending a single ICMP packet (P1) from a wired client (wired client 1 - W1) at time T1 addressed to the wireless client 1 (WL1) via the AP. The packet P1 is also recorded at wireless client 2 (WL2). Let’s firstly consider the exchange with WL1. Upon receiving P1, the wireless client WL1 sends a reply packet (P2) back to W1 via the AP. The test setup is depicted in Figure 4.7. P1 is sent at timestamp T1 from W1 and arrives at the wireless interface of the AP at T2. The wireless client WL1 receives P1 at T3.

![Figure 4.7 Test-bed time synchronization setup 1](image)

A copy of the packet P1 is also received at wireless client WL2 at time T4, however WL2 does not send a response packet in this instance. When WL1 receives P1 it sends a reply packet (P2) back to W1 at time T5 and this reply packet is recorded at the AP at T6 and received at W1 at T7.

In a separate test the level of synchronization of wired client 2 (W2) is ascertained by repeating the same process for W2 that was carried out for W1. This time a multicast ICMP packet (P3) is sent at time T1 from W2 to WL2 as illustrated in Figure 4.8. The packet is received at the wireless interface of the AP at time T2 and is received at wireless client 2 (WL2) at time T3, while a copy of P3 is received at wireless client 1 (WL1) and T4. A reply packet (P4) is sent by WL2 to W2 at time
T5 and is received at the AP at T6. The packet P4 is finally received at W2 at time T7.

Figure 4.8 Test-bed time synchronization setup 2

This test was designed to ensure that all nodes in the test-bed are synchronized to within single figure milliseconds before carrying out any delay calculations with the iAP. Once this requirement is satisfied, the delay values provided by the calculation mechanism can be deemed accurate. Results for this test are presented in Chapter 6.

4.3.1 One-Way Delay Calculation without Synchronized AP

The delay calculation mechanism described in section 3.3.2 involves a synchronized AP along with synchronized client endpoints. This ensures that the AP (iAP) has knowledge of iM2E delays, and therefore of wireless MAC delays as detailed in section 3.3.1. This is significant in the context of distinguishing between delays caused in the wired network (WNDs) as opposed to the wireless MAC (WLNDs), as illustrated in Figure 3.4 in Chapter 3. However, if the total M2E delay is required for R-factor calculation, as opposed to the iM2E delay, then the AP does not necessarily need to be synchronized. Consider a scenario where a VoIP session takes place between two synchronized endpoints, and the AP is not synchronized. The iAP delay calculation mechanism will still provide accurate one-way M2E delays due to the fact that the calculations take place on the AP. The following example explains this process.
The iAP clock is not synchronized and lies between two synchronized endpoints N1, and N2. As can be seen, the time on the AP clock is 32 minutes and .5 seconds (1920.500s) ahead of the two endpoints clocks (Figure 4.9). All intra-one-way delays ($iM2Es$) are $= 1ms$. Both the minimum downlink delay ($\Delta_d$) and the minimum up-link delay ($\Delta_u$) are each 2ms. The delays are calculated as follows:

<table>
<thead>
<tr>
<th>$dn_1$</th>
<th>$dn_2$</th>
<th>$\Delta_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1920.501s</td>
<td>-1920.499s</td>
<td>0.002s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$up_1$</th>
<th>$up_2$</th>
<th>$\Delta_u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1920.501s</td>
<td>-1920.499s</td>
<td>0.002s</td>
</tr>
</tbody>
</table>

One-Way Delay Calculation
With round trip time ($\Delta_R$) equal to ($\Delta_d+\Delta_u$) plus processing time on N2, $\Delta_R$ is calculated as 5ms despite the lack of synchronization in the AP clock. This was an interesting observation during the development of the iAP. However, for the results presented in this thesis, the iAP is synchronized with all endpoints in the experimental test-bed.
Section IV

RESULTS
Chapter 5

SIMULATION RESULTS

The previous chapter described implementation details relating to both the simulated and test-bed approach. It firstly dealt with setup of preliminary simulations designed to evaluate the capacity of 802.11 networks to support VoIP, in terms of numbers of simultaneous calls, as well as the feasibility of implementing prioritization of certain VoIP calls over others in 802.11e. These simulations were set up in the NS-2 network simulator. It also described more detailed simulations using the NS-3 simulator which implemented a traffic prioritization mechanism involving a Multimedia Category (MC) hierarchical structure. The results for all simulations in NS-2 and NS-3 are presented and evaluated in this chapter. Results relating to an evaluation of the experimental test-bed are presented in Chapter 6.

5.1 Test 1 – NS-2: IEEE 802.11e VoIP Call Capacity

As described in section 4.1.1.1, this simulation looks at network capacity in an 802.11e enabled network in terms of QoS when delay and packet loss are considered. In a voice only scenario, Figure 5.1 presents the combined mean uplink and mean downlink delay experienced by multiple VoIP sessions due to the MAC layer. These
results suggest that 802.11b can support up to 12 concurrent G.711 VoIP calls before the mean delay on the downlink is noticeably affected. When a thirteenth session is added, the mean contention delay at the wireless MAC layer is significantly increased to above 100ms. This is due to an increase in collisions of packets on the downlink within the CSMA channel access mechanism.

Figure 5.1 One-Way delay vs. Number of VoIP sessions over 802.11b

This test also suggests that 14 VoIP sessions is the upper bound for simultaneous VoIP calls. When a 15th VoIP call is added to the network, the QoS for all sessions will become unsatisfactory according to the recommendation G.114 [30] as the one-way delay rises above 150ms.

When taking packet loss percentage into account, the packet drop rate surpasses 1% when an 11th VoIP session is added to the network as observed in Figure 5.2. The drop rate increases to 17% when the 12th VoIP session is added to the network.
With 11 simultaneous VoIP sessions, a high drop rate coupled with an increased mean delay suggests that 10 simultaneous VoIP calls is the upper bound for 802.11b in this scenario. This limit may be increased with the implementation of packet loss mitigation techniques such as PLC or with the use of a different codecs.

5.2 Test 2 – NS-2: The Effects of Differing One-Way Delay on VoIP

When taking delay alone as a factor affecting QoS, 12 simultaneous VoIP calls can occupy an 802.11b network without any considerable negative effect due to MAC contention on the QoS of all sessions according to Test 1. For the NS-2 simulation described in section 4.1.1.2, twelve VoIP calls are run simultaneously where two sessions, (S1, and S2), have WNDs of 100ms and 50ms added to their one-way delays respectively. In terms of prioritization, there are three scenarios simulated here:

1. All VoIP sessions have equal priority
2. S1 and S2 are prioritized on the Uplink
3. S1 and S2 are prioritized on the Uplink and Downlink
Figure 5.3 illustrates the one-way delay values on the Y-Axis and the packet numbers for the uplink and downlink on the X-Axis for Test 2.1. The M2E delays for S1 on the downlink are between 180 and 200ms, which is above the ITU-T recommendation of 150ms. The uplink delay for S1 is within the ITU-T recommendation, as is the uplink delay for S2. The downlink delay for S2 only just meets the one-way delay requirements for a VoIP call. All the short range VoIP sessions (S3 – S12) experience one-way delays that maintain an acceptable level of QoS on both up and downlink.

As evident, there is a significant difference observed in the uplink and downlink values recorded. As described in section 2.3.5, and known as AP bottleneck, this is due to an asymmetry in 802.11 between the AP and the wireless nodes when accessing the wireless medium. Within the 802.11 CSMA/CA mechanism, the AP contends for access to the medium with n nodes, and receives approximately the same number of transmission opportunities, although the AP has approximately n times the traffic to transmit. This problem is studied in further detail in [43].
In Test 2.2 the uplink traffic for VoIP sessions S1 and S2 is assigned higher priority status. This simulation scenario represents a real-world case of prioritizing uplink traffic where the 802.11e EDCA parameters would be configured on each individual wireless endpoint. The prioritized uplink traffic is assigned the EDCA Voice parameters (AC_VO AC), and all other voice traffic is assigned EDCA Video category parameters (AC_VI). This is a pre-cursor to the MC categories used in the later simulations.

Although the uplink traffic for the two long distance VoIP calls is now prioritized, there is no considerable improvement in the uplink delay values, this is primarily because their uplink delays were already low. The un-prioritized uplink traffic (sessions 3 – 12) experiences a slight overall increase in wireless MAC contention delay in comparison with Test 2.1 due to operating within the lower priority AC, however this increase is minimal (Figure 5.4). There is a change in the downlink delay values for all sessions. This improvement has come about due to packets originating from the prioritized STAs (S1 & S2) gaining quicker access to the medium via selecting their countdown timer from the AC_VO CW range in CSMA/CA. This ensures that uplink packets for S1 & S2 experience faster medium access and also reduce the virtual collision rate for downlink packets within the AC_VI AC (the AP, with downlink traffic for 12 sessions, now competes with 10 other nodes for medium access instead of 12 within AC_VI), thus reducing the overall queuing time (delay) for packets in the downlink. The downlink traffic for S1 still experiences one-way downlink delays of up to 150ms but this is below the delays experienced in Test 2.1.

![Figure 5.4 Up/Downlink Delay – Test 2.2: S1 & S2 Prioritized on Uplink](image)

Figure 5.4 Up/Downlink Delay – Test 2.2: S1 & S2 Prioritized on Uplink
The final test scenario builds on Test 2.2. Where the previous test prioritized the two delayed VoIP sessions (S1 & S2) in the uplink direction only, this test also prioritizes the downlink traffic for these two VoIP sessions. The most notable improvement stemming from this is that the two prioritized voice sessions experience significantly reduced delay in the downlink direction (Figure 5.5).

![Figure 5.5 Up/Downlink Delay – Test 2.3: S1 & S2 prioritized in Uplink and Downlink](image)

The two prioritized sessions (S1 & S2) now experience lower delays on the downlink at the wireless MAC layer (Y-Axis: ~57ms and at ~108ms). This improvement is at the expense of the remaining ten VoIP sessions which now have one-way delays varying from 18 – 35ms in the uplink and 60 – 130ms in the downlink. These values are still however within the ITU-T recommended time of 150ms.

These simulations contributed to the planning and development of the iAP by providing an insight into the behaviour of one-way delay characteristics for VoIP when implementing prioritization amongst VoIP sessions. The most noticeable improvements are for VoIP sessions that are prioritized in the downlink direction. This reinforced similar findings in the literature and pointed towards a greater need to focus on downlink rather than uplink traffic QoS improvements in the iAP.
5.3 Test 3 – NS-2: Preliminary EDCA Parameter Optimization

As described in section 4.1.1.3, this test evaluates the feasibility of improving the QoS of VoIP by tuning 802.11e EDCA parameters. Whereas the previous test looked at one-way delay only, this test goes one step further in that it calculates a QoS score for each session and prioritizes VoIP sessions based on their R-factor values.

In the first scenario, Test 3.1, all sessions exist within the same AC therefore all sessions are of equal priority. The one-way delays for each session are provided in Figure 5.6, as well as uplink and downlink R-factor values - however they are more clearly presented in Figure 5.7. Average delay values are calculated for each individual session and are then plugged into an E-Model calculator to produce a QoS R-factor. The session S1 has a visibly higher mean delay than the remaining sessions in the network which is of course due to the WND of 100ms.

![Graph showing one-way delays and R-values](image)

Figure 5.6 Test 3.1 - Mean Delay without Prioritization, with Associated R-Values

The downlink one-way delay for S1 is above 240ms whereas the remaining VoIP sessions experience a one-way delay on the downlink of approx. 150ms due to contention at the AP. All one-way delays on the uplink are below the ITU-T
recommendation of 150ms. When taking a closer look the R-Values (Figure 5.7), the QoS for S1 is “medium, with some users dissatisfied” according to the E-Model, due to S1 having an R-factor of 79 on the downlink.

Figure 5.7 Test 3.1 - R-Values for Uplink and Downlink without Prioritization

When prioritization is implemented for S1 (by incrementing the AIFSN value for the remaining sessions on their uplink and downlink), there is a noticeable improvement in the delay characteristics for S1 downlink (250 to 190msec) as shown in Figure 5.8. This improvement does not come at the expense of the remaining sessions to any large degree, as the downlink traffic for the remaining sessions experiences no significant change. Although the mean delays for the traffic on the uplink did increase slightly although this was not enough to have any noticeable effect on the QoS of the sessions.
With a closer analysis of the R-factors (Figure 5.9), the QoS of S1 is dramatically improved with the R-factor on the downlink increased from 79 to 86 for S1, this occurs again without any significant effect on remaining VoIP sessions. A comparison between the R-factors for the two scenarios is provided in Figure 5.10 where a clear improvement in downlink R-factors is illustrated, whereas uplink remains similar.
Figure 5.9 Test 3.2 - R-Values for Uplink and Downlink, With Prioritization

Figure 5.10 Test 3.2 - R-Value Comparison

The improvement in QoS for S1 is due to the priority treatment that packets belonging to S1 receive in the AP queue on the downlink. These packets have a higher probability of accessing the wireless medium, both against contending downlink traf-
fic within the AP queue, and against the remaining wireless nodes. This simulation test illustrates how tuning EDCA parameters in an 802.11e infrastructure network can improve the QoS for VoIP session where a certain VoIP session has a lower R-factor value in comparison with other sessions. This test contributed to the development of the iAP by showing that the optimization of EDCA parameters can improve the overall QoS for a VoIP session, without necessarily degrading the QoS of the remaining sessions in the network.

5.4 Test 4 – NS-3: Multimedia Category Evaluation

As described in section 4.1.2.1, two simulation tests were carried out in NS-3 to evaluate the MC traffic prioritization mechanism. As described in Figure 4.4, multiple VoIP sessions operate between wired and wireless clients with different WND delays. The first test is run using the default EDCA channel access parameters while the second test uses the proposed MC prioritization mechanism for channel access based on R-factor values as opposed to the default EDCA.

5.4.1 Test 4.1 – NS-3: Default EDCA

For this scenario, all VoIP sessions are categorized within the 802.11e AC_VO voice access category. The start and stop times for each session are illustrated in Figure 4.5. At the beginning of the simulation, there is no noticeable QoS degradation among VoIP sessions as there are only 5 sessions active (S1 – S5). However the network becomes significantly congested 25 seconds into the test (at T=25s) when a 6th session (S6) begins (Figure 5.11). At this point, the downlink R-factors for all active sessions drop to very unsatisfactory levels due to both high packet loss and one-way delay rates (Figure 5.12, Figure 5.13). The high packet loss rate is due to buffer overflow of packets awaiting transmission at the AP buffer which is set to 65 packets.
As there is no active prioritization mechanism in operation in this simulation, the session statistics remain the same until there is an alteration in the traffic characteristics. The session S6 finishes at T=55s, at which point there is a clear improvement in quality for all sessions as the network is no longer congested.

![Figure 5.11 Default EDCA scenario - Downlink R-factors](image)

![Figure 5.12 Default EDCA scenario - Downlink Percentage Packet Loss](image)
The session S7 begins at T=65s and the resulting effect on the remaining sessions is illustrated at T=70s where a reduced QoS of all remaining sessions is observed. The downlink R-factor for S7 is 42, however the QoS of sessions S1 and S2 is not dramatically degraded and S1 maintains a downlink R-factor above 70. The session S2 finishes at T=85s which again reduces contention on the network and thus returns all remaining sessions to a state of experiencing good QoS as seen at T=90s. The final VoIP session to join the network (S8) begins at T=95s. With the largest WND of all sessions in the network, S8 experiences a packet loss rate of up to 20% (Figure 5.12). This is due to the fact that S8 packets start arriving at the AP last at the start of simulations and are thus impacted more by AP loss due to buffer overflow. It is worth noting that this characteristic would diminish over time but is evident in above due to the nature of simulation setup. With one-way delay values as high as 560ms, these values contribute to an R-factor of less than 10. Note that an R-factor below 50 is not recommended in any case by the ITU-T. The effects of exceptionally high one-way delays are also observed in S5 and S3 where downlink R-factors of <10 are also experienced. The sessions S4, and S7 experience downlink R-factors of <30 while the session with the lowest WND, S1, experiences a significant loss in quality with one-way delay values of almost 300ms (Figure 5.13).
The uplink R-factors are presented in Figure 5.14, and show that the QoS of S3 and S8 becomes non-satisfactory at T=100s.

![Figure 5.14 Default EDCA scenario - Uplink R-factors](image)

The prime contributing factor to the drop in R-factor for S3 is the percentage packet loss as illustrated in Figure 5.15 where 3% of packets are dropped for S3 after session 8 joins the network. The low QoS level for S8 clearly caused by the high one-way delay values recorded of over 250ms (Figure 5.16). Most of this delay is attributed to the pre-set 245ms WND delay on S8 along the remaining 15-20ms of contention delay however these values still contribute to an unsatisfactory level of QoS. The uplink one-way delays for the remaining VoIP sessions largely correlate with the WND delays added to each session.
As described in section 3.3.1, the one-way wireless MAC delay for all simultaneous VoIP sessions should be similar due to the largely equal probability that packets in the same access category will access the wireless medium. The difference between the delays experienced by VoIP sessions at the wireless MAC layer is minimal when compared with the difference in the overall one-way delay experienced by these sessions, which is determined principally by the configured WND values (Figure 5.17, Figure 5.18).
For the last 25 seconds of this simulation, four of the six active sessions (S3, S5, S7, S8) that have WNDs of over 100ms, which contributes to a low R-factor on the downlink (Figure 5.11). However the two sessions with lower WNDs (S1 & S4), also have R-factors < 70. All sessions have an equal opportunity to access the wireless medium, and overflow at the AP buffer on the downlink contributes to relatively high packet loss rates for all sessions.
5.4.2 Test 4.2 – NS-3: Multimedia Categorization (MC) Implementation

The previous test scenario carried out on the default EDCA channel access mechanism represents a scenario where multiple VoIP calls with different one-way delays compete for network resources within the same access category (AC_VO). The exact same scenario is replicated here in terms of traffic characteristics and call sequence, whereby all VoIP calls begin and end at the same time as those in the previous scenario, and the WNDs applied to the individual calls are the same. This scenario however applies the Multimedia Categorization (MC) prioritization mechanism as described in 3.4, which means in effect that VoIP sessions will be prioritized according to their R-factors, based on the prioritization algorithm detailed in section 3.4.2.

As evident from Figure 5.19, and similar to the default EDCA scenario, when the first five VoIP sessions begin, there is no considerable negative effect on the QoS of any of the competing sessions. However, the addition of each new VoIP session to the network at T=25s increases the delay of the pre-existing sessions in the downlink, this is observed at T=30s when the R-factors are calculated for each session. Before any prioritization is implemented, session 6 (S6) has a non-satisfactory QoS rating which is caused primarily by packet loss at the AP (Figure 5.20), despite its low WND of 20ms. Two VoIP sessions (S3 & S5) with large WNDs have significant one-way delays which contribute to a lower R-factor on the downlink for these sessions (Figure 5.21). At T=30s, S5 has a M2E delay on the downlink of 256ms and delay for S3 is 240ms, while the remaining sessions have downlink M2E delays of less than 122ms.
For each iterative calculation of R up to T=25s, all of the sessions downlink R-values are above 80 (Figure 5.19) and thus no action is taken by the prioritization algorithm. At T = 35s the prioritization algorithm selects S3 for prioritization as it has the lowest R value and it satisfies all the conditions of the algorithm. S3 is then prioritized and an improvement in QoS is evident at the next step of T=40s where S3 now has an R-Value of 86 due to its promotion to MC2.
At $T=40s$, the downlink R-factor for S5 remains below 80 and thus S5 is at this point the only session that is eligible for prioritization. Therefore at $T=45s$, the algorithm promotes S5 to $MC_2$ along with S3. An improvement is observed at $T=50s$, where all sessions are in a state of satisfactory QoS, and thus there is no need to promote any of the session. At $T = 55s$, the VoIP session S6 is terminated, this effects the one-way delay of sessions 1, 2, and 4 as illustrated in Figure 5.21, whereby the downlink one-way delay of S1, S2, and S4 decreases due to less contention on the downlink in the AP. Note that this improvement does not provide a significant positive contribution to QoS, as these sessions already had a good QoS score before S6 was terminated.

![Figure 5.21 MC Scenario - Downlink M2E Delay](image)

Session 7 (S7) becomes active at $T=65s$ and immediately experiences a QoS score of 77 which is primarily due to the one-way delay as observed at $T=70s$ in Figure 5.21. S7 is thus selected by the prioritization algorithm at $T=75s$ and is thus promoted to $MC_2$ along with S3 and S5. At $T=80s$, session 7 experiences good QoS with a considerable R-factor improvement caused by a decrease in one-way delay of over 50ms. This operation has the effect of marginally reducing the QoS experienced by S2, S3, and S5, however the R-factors for all sessions remain above 80R and thus all
sessions maintain a state of good QoS. The session S2 terminates at T=85s causing a minimal improvement in the QoS of S1, S3, and S4. The most notable impact on the remaining sessions is on S4 whose one-way downlink delay is reduced by 81ms, from 136ms to 55ms at T=90s, and its corresponding R-value is improved to 92 as illustrated in Figure 5.19 and Figure 5.21. This improvement in one-way delay for S4 is due to the reduced contention for the wireless medium as there are less packets contending for channel access at the AP.

The session S8 begins at T=95s. As per the prioritization algorithm, it joins the network in the default lower MC (MC₃) and therefore has a lower probability of access to the medium than the sessions S3, S5, and S7 which are at this point in the MC₂ category. With a WND of 245ms, the one-way delay of S8 is expected to be relatively high. At T=100s, the downlink one way delay of S8 is calculated to be 306ms, which contributes to an R-value of 72. This results in a Some Dissatisfied QoS classification and the algorithm then proceeds to promote S8 to MC₂, joining with S3, S5, and S7. This operation improves the downlink R-factor of S8 to 79 primarily due to the reduction in one-way downlink delay of 56ms to 250ms as illustrated at T=110s in Figure 5.21. Despite the promotion to MC2 and the fact that it is much improved on its initial score, the QoS of S8 at T=110s remains classified as Some Dissatisfied according to the E-Model with an R-factor of 79. Therefore, on the next iteration of the prioritization algorithm, S8 again satisfies all of the prioritization conditions of the algorithm, this time however, S8 is promoted to MC₁ and will be the only session to occupy this category. At T=120s, the downlink one-way delay of S8 is reduced by 2ms, which serves to improve its R-factor score to 80 (Figure 5.19), which is within the E-Model category of Satisfied. The most notable impact of prioritization of S8 to MC₁ is on S4, where the downlink one-way delay is increased by 30ms, to 106ms, and percentage loss is increased from 0% to 1% (Figure 5.20). These factors contribute to a reduced R-factor for S4 of 87. Despite this however, S4 still has a satisfactory level of QoS.

The overall QoS of all VoIP sessions on the uplink is largely unaffected. Figure 5.22 illustrates the R-factors for all sessions in the uplink direction. No VoIP calls experience any packet loss in the uplink throughout this test (Figure 5.23). When S5 is prioritized at T=45s, its uplink R-factor is slightly improved due to a reduction in its one-way uplink delay as observed in Figure 5.23 and Figure 5.24.
When session 8 joins the network, it experiences non-satisfactory QoS at first as evident from Figure 5.22, again due to the high one-way delay caused by the WND of 245ms. This is supplemented by a small contention delay at the wireless MAC which causes a one-way delay of over 256ms. When S8 is promoted to MC1 at T=120s in the downlink direction, the uplink one-way delay for that session is reduced to 252ms and thus improves the uplink QoS of S8 to a satisfactory level.
5.4.3 Wireless MAC delay Improvement

Taking a closer look at the one-way delays for all the VoIP sessions, Figure 5.25 and Figure 5.26 illustrate the downlink and uplink wireless MAC delays respectively. On the downlink, the wireless MAC delay for the sessions S3, S5, S7, and S8 is reduced when each session is prioritized in the MC framework at times T=30s, T=50s, T=80s, and T=110s respectively.
This reduces the overall downlink one-way delay for these sessions as shown in Figure 5.21, thus improving QoS. The one-way MAC delay for the remaining session is not considerably affected with the most notable effect is on sessions 2, 4, and 5 at T=40s where their wireless MAC delay is marginally increased which causes a small reduction in their QoS. At the end of Test 4.2, all active sessions have a satisfactory QoS score, showing that the prioritization algorithm can improve the overall QoS of VoIP sessions that have large WNDs, at the expense of competing VoIP ses-

Figure 5.25 MC Scenario - Downlink Wireless MAC Delay

Figure 5.26 MC Scenario - Uplink Wireless MAC Delay
sions that have lower WNDs, while still maintaining acceptable QoS for all sessions. This validates the core concept of the thesis as illustrated in Figure 3.6 and described in section 3.3.1 whereby synchronized time is used to precisely measure delays and dynamically optimize downlink MAC parameters to optimize QoS.

5.5 Summary

This chapter presented the results for all simulations in NS-2 and NS-3. The NS-2 results evaluated the VoIP capacity of 802.11 networks, the effects of differing one-way delays on VoIP, and examined EDCA parameter optimization. Further simulations with NS-3 presented results that more clearly highlighted the potential of the proposed QoS optimization mechanism using the MC hierarchical system.
Chapter 6

EXPERIMENTAL RESULTS

The previous chapter provided simulation results that validated the core thesis contribution – using delay/loss measurements to determine QoS and implement the multimedia category prioritization mechanism as described in Chapter 3. These results that involved many network nodes and varied network conditions facilitate a scale of testing that otherwise would prove too costly and timely to implement in a real-world test-bed. For this reason, the core of the MC prioritization validation was carried out through simulation. However – key questions and challenges remain regarding the technical feasibility of implementing the approach dynamically and in real-time. This chapter presents results arising from the implementation of the key components within the so-called intelligent Access Point (iAP). These include the precise delay calculation module, QoS estimator, prioritization algorithm and the prioritization mechanism. However, another basic requirement was the need for effective (single figure msec) synchronization. This synchronization mechanism is also validated.
6.1 Synchronization Validation

6.1.1 Wired client 1

Accurate synchronization of all devices in the test-bed to within single figure milliseconds is required to ensure precise delay calculations provided by the iAP as detailed in section 4.3. Figure 6.1 illustrates the timeline resulting from sending an ICMP packet from W1 to WL1 and back. It shows that the sequence of timestamps is within 3ms where the total RTT for \( W1 \leftrightarrow WL1 \) (also recorded at WL2 at T4) is 2.7ms in the test-bed. The same process is repeated in Figure 6.2 where a packet P3 is sent from W2 addressed to WL2 where RTT for \( W2 \leftrightarrow WL2 \) (also recorded at WL1 at T4) is 3.4ms (Figure 6.2).

Figure 6.1 Wired host 1 Time Synchronization

Figure 6.2 Wired host 2 Time Synchronization
As shown in both instances (Figure 6.1, Figure 6.2), the two wireless clients received the multicast packet within 0.6ms of each other and considering the full timeline shown above, and considering the overall delay budget of ITU-T G.114 (150ms each way) and thus the precision of delay measurements required for the prioritization process on the iAP, it is clear that all nodes are synchronized sufficiently. These tests were a prerequisite to running experiments on the iAP. The trends evident in Figure 6.1 and Figure 6.2 were repeated during all tests with monotonically increasing timestamps and RTT <4ms. As such, the test-bed is deemed to have a sufficient level of synchronization for experiments.

6.2 Experimental Test-bed Delay & R-factor Validation

Simulation results in Chapter 5 validated the multimedia category prioritization mechanism described in Chapter 3. These simulation tests were carried out in order to assess the extent to which the overall concept was feasible and thus was a necessary step before a physical proof-of-concept could be considered. In this section, comparisons are made between the results derived via simulations and the test-bed in terms of one-way delay calculation, QoS evaluation, along with other results that prove the viability and feasibility of the prioritization mechanism.

Once the core functionality of the test-bed was operational a simple experiment was run using the default 802.11e EDCA channel access mechanism. This compared real test-bed results to NS-3 simulations in order to validate the previous simulation results. Two simultaneous G.711 VoIP calls were run on the test-bed as illustrated in Figure 6.3 that had different one way wired network delays (WNDs), implemented using NetEm. The same scenario was then repeated in an NS-3 simulation as previously presented in Chapter 5.
At the beginning of the test, both sessions have the same delay characteristics, and all calls reside within the EDCA AC_VO access category. R-factors are calculated twice (Figure 6.3) via the RTCP mechanism (Chapter 3) for each session at an interval of ~15s in both the uplink and downlink direction. The calculation interval is dependent on the rate of RTCP SR/RR packets sent by each VoIP application. Once the R-factor has been calculated twice for each session, delay is increased by steps of 50ms on the downlink of Session 1 (S1) using NetEm emulator. An R-factor is calculated twice for each delay setting until the WND for S1 on the downlink is 400ms (Figure 6.4). Note that G.114 recommends that one-way delay should not exceed this for general network planning [30].
A high correlation is observed between the R-factor calculated in the test-bed and those recorded in NS-3. The R-factors for S2 on the downlink were unchanged throughout the tests as there was no added WND (Figure 6.5).

Figure 6.5 Experimental and Simulation R-factors for S1 (Uplink) & S2 (Uplink & Downlink)
All uplink R-factors recorded for both sessions were above 90R, as WNDs were only applied on the downlink. These initial tests confirm the accuracy of test-bed delay calculation operation where RTCP packets are used to calculate one-way delays.

6.3 Validation of Proof-of-Concept

The results in section 6.2 validated the delay and QoS calculation mechanism of the iAP by comparing results with NS-3 under the same test-bed used in earlier simulations outlined in Chapter 5. As detailed in the more extensive simulations, the prioritization mechanism works optimally when the wireless network is in near congested state. One-way delay and packet loss were factored into QoS estimations in the NS-3 simulations in Chapter 5, however, experimental test-bed tests only factor in one-way delay.

6.3.1 MC Prioritization

6.3.1.1 Preliminary Test

A further key requirement of the test-bed was to demonstrate that it is technically possible to implement real-time dynamic prioritization of traffic streams based on the iAP QoS calculation mechanism, using readily available network devices. According to the 802.11e protocol, all VoIP calls that access an 802.11e enabled router will be assigned to the same AC (AC_VO). Prioritization is implemented via the iptables command in the Linux Traffic Control suite as detailed in Chapter 4. The purpose of this test was to prove that real-time prioritization could be achieved by mangling the DSCP field in relative VoIP stream packets. This experiment also involved two simultaneous VoIP sessions (S1, S2) where additional Wired Network Delay (WND) was added to the one-way downlink (wired to wireless) delay of S1 via NetEm (Figure 6.3). A 50ms step delay was incrementally added to VoIP session (S1) at 30 second intervals up to a maximum of 400ms as illustrated in TABLE X and Figure 6.6.
Initially, both VoIP sessions reside within the MC$_3$ category. The delay for both sessions is shown in Figure 6.7 where the effect of the increasing WND on the down-link delay of S1 is clearly visible. The R-factor values for both sessions are shown in Figure 6.8, where the downlink R-factor for S1 drops to 78 after 180s in the experiment, with a WND of 300ms. There is a steady decrease in QoS for S1 as the WND is increased on the downlink. When the WND is 400ms on S1, nearly all users would be dissatisfied according to the E-Model with an R-factor of 57.

<table>
<thead>
<tr>
<th>Time</th>
<th>0</th>
<th>30</th>
<th>60</th>
<th>90</th>
<th>120</th>
<th>150</th>
<th>180</th>
<th>210</th>
<th>240</th>
<th>270</th>
</tr>
</thead>
<tbody>
<tr>
<td>WND</td>
<td>0</td>
<td>50</td>
<td>100</td>
<td>150</td>
<td>200</td>
<td>250</td>
<td>300</td>
<td>350</td>
<td>400</td>
<td>400</td>
</tr>
</tbody>
</table>

TABLE X: WND ADDED BY NETEM TO S1

Figure 6.6 WND on Downlink for S1
6.3.1.2 Multimedia Category Prioritization

The subsequent test built on the above by implementing a simple prioritization mechanism that reassigned dynamically the session that had dropped below the satisfactory threshold to the higher priority MC. The timeline of this test was somewhat different as illustrated in Figure 6.9. Firstly, NetEm WND delay was incremented every 10 seconds by 50ms until 200ms was reached. Subsequent to this point, (observed at 50s in Figure 6.10), 10ms of WND delay is added every ten seconds until the QoS dropped to the not satisfactory category.
The overall one-way delays of both sessions are plotted in Figure 6.10. The delay for the downlink of S1 is largely composed of the WND added by NetEm, and contention delay on the wireless interface on the AP. The corresponding R-factors for the downlink of S1 and S2 are shown in Figure 6.11 while the R-factors for the uplink are shown in Figure 6.12.
Calculated delay for S1 & S2

The QoS of S1 drops to \textit{not satisfactory} 80 seconds into the experiment where R falls to 79 as a result of a 230s WND implemented by NetEm. At this point the iAP assigns all VoIP traffic for S1 to MC\textsubscript{2} and thus gives it priority treatment.

To realize prioritization, the DSCP values in the RTP packets carrying S1 voice data are changed (mangled) in real time so that the 802.11e controller reassigns them to the MC\textsubscript{2} queue on the wireless interface within the iAP. At the next QoS calculation (90s in Figure 6.10), the delay for S1 has improved from 246ms to 239ms on the downlink, which translates to an R-factor improvement of 2, increasing the R-factor to 81. At the 100 seconds point of the test the WND delay in S1 was increased to 240ms using NetEm. This caused an increase in the overall M2E delay of S1 and thus a small decrease in R-factor, though still above 80. The WND was then increased to 250s which caused a further decrease in QoS for S1 dropping R to 79. At this point, S1 was promoted to MC\textsubscript{1} however this does not improve QoS sufficiently.
As there are only two active VoIP sessions in this example, the wireless MAC contention delay is not very large and thus the potential for improvement is minimal which explains the above results. As the simulation results in section 5.4 show however, with more competing VoIP calls and/or background traffic, there would be increasing contention delay and the thus greater scope for QoS improvement. Although improvements are minimal in terms of QoS, the main point illustrated is the feasibility of the mechanism. Considering the timescale of typical VoIP calls, this proof-of-concept clearly illustrates that off-the-shelf hardware can be used to implement a so called intelligent Access Point that works in real time to optimize QoS across various VoIP sessions.
6.4 Summary

This chapter has served a number of purposes. It firstly served to validate the simulated results by showing a very strong correlation between simulated and real test-bed data. More importantly, it also demonstrated the technical feasibility of the so-called intelligent Access Point (iAP). Earlier simulations had validated this concept more thoroughly through extensive simulation. The central point demonstrated in this chapter is to show how the mechanism can be practically realized via a proof-of-concept.
Section V

Conclusions
This thesis has addressed two key research questions and in doing so has made a number of significant contributions. The research questions firstly examined, largely through simulation, the potential of synchronized time to facilitate a significant improvement in QoS management of RTC over WiFi under certain conditions. The second question related to the practical feasibility of the approach, and addressed this through development of a proof-of-concept (PoC) implementation. In the following section, the research questions are revisited and core contributions outlined. Some future ideas that build on this thesis are also presented.

7.1 Core Research Questions & Contributions

Two core research questions were raised at the beginning of this thesis. The first question addressed the possibility of using synchronized time to improve QoS over WiFi, while the second question addressed the feasibility of overcoming the real-world technical challenges associated with the first question. These are reproduced and assessed below:
1. *Can the implementation of time synchronization over WiFi improve the QoS of VoIP sessions that have relatively high one-way M2E delays?*

This thesis has shown largely through simulation that synchronized time and resulting delay information can be used to deliver significant QoS improvements. This is done by manipulating MAC parameters to optimize QoS in the downlink. Simulations of a Multimedia Categorization (MC) prioritization mechanism in an intelligent Access Point (iAP) are shown to dynamically improve the QoS of multiple RTC streams based on their R-factor scores. QoS improvements are most significant when the network is nearing its bandwidth limit and thus where contention at the MAC is becoming problematic. Simulations also show the need to focus on the downlink due to AP bottleneck issues. Note that simulations examined both delay, and packet loss in R-factor determination.

2. *What are the key engineering challenges required to prioritize certain VoIP calls over others using the 802.11e EDCA traffic access mechanism in a real-world RTC environment?*

Simulation is a very useful tool to evaluate complex network scenarios facilitating the evaluation of the RTC traffic prioritization mechanism. However, in order to assess the feasibility of this in the real world, an experimental test-bed was developed. The development of the real-world test-bed and, at its core; the iAP, presented considerable technical challenges which were all overcome and are described as follows:

a. *Is it possible to determine in real-time, the accurate one-way M2E delays for multiple RTC streams?*

In order to exploit the presence of synchronized time and calculate accurate one-way M2E delays, the iAP implements a novel delay calculation technique whereby it uses *NTP*
timestamps and DLSR values contained in RTCP packets along with AP timestamps to calculate intra M2E (iM2E) delays, which are summed together to provide one-way M2E delays for individual RTC sessions in real-time.

Tests also illustrate the additional value of having the AP synchronized. Without it, M2E can be determined, but with it, the full breakdown of delays can be achieved.

b. Is it possible to translate dynamic network conditions into QoS scores based on delay/loss for RTC streams in real-time?

RTCP traffic is generated at regular intervals (in the order of seconds), which ensures that useful feedback can be received if acted upon quickly by the iAP. RTCP information is shown to facilitate measurement of precise delay information and tests show that a QoS estimator module can translate these values in real-time into R-values. Note that packet loss was not considered in the QoS estimator for a number of reasons. Firstly, the key concern if this thesis is with regard to delay measurement. Secondly, the inclusion of packet loss is a trivial addition as RTCP information readily includes packet loss information without any processing. Finally, the primary purpose of the test-bed was to prove the feasibility of the approach rather than carry out extensive testing. For the limited range of tests carried out on the test-bed, packet loss was not an issue in any event.

c. How feasible is it to manage/configure multiple RTC streams via a single management device, by building on the EDCA Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) mechanism in real-time?

Once delay calculation and QoS estimation has taken place for individual sessions, a prioritization mechanism was required to improve QoS for RTC sessions based on their R-factor scores. A solution to this is provided in the utilization of the iptables
facility in the Linux Traffic Control suite which enables the tuning (or ‘mangling’) of the DSCP values in the IP header of packets belonging to sessions that are to be prioritized. This mangling operation enables the re-directing of packets to one of three Multimedia Categories (MCs) which each have different levels of priority. The scripts developed were all tested in real-time using standard AP hardware and a basic, separate management station. Undoubtedly, these could be merged into a single device, but in separating them, the approach is better aligned with the SDN concept of centralized management and northbound/southbound interfaces.

In terms of contribution to the State of the Art (SOTA), the development and testing of an intelligent Access Point (iAP) represents the core contributions of this work. Its intelligence is embodied as follows:

- **A Novel VoIP Delay Calculation Mechanism** – Using RTCP SR/RR packets and AP timestamps, this mechanism calculates accurate one-way delays for multiple VoIP sessions on a WiFi AP. This requires that endpoints are time synchronized to single digit milliseconds for successful operation. The E-Model is invoked taking the delay values for each call once delay values are calculated. Note that for real-world experiments, packet loss was not considered.

  - **Intra-Mouth-to-Ear Delay (iM2E) Calculation** – With the addition of AP synchronization, network management software implementing this technique can obtain intra-one-way delays for VoIP. This provides additional useful information regarding medium access delays in a WiFi network. These delay values are calculated by default in the iAP and are combined to provide Mouth-to-Ear (M2E) delay values for each VoIP call.

- **An E-Model based VoIP Prioritization Algorithm** – This algorithm works within the boundaries of the E-Model where VoIP sessions that are considered to have ‘bad’ QoS are prioritized over VoIP sessions that are considered to have ‘good’ QoS. This algorithm was shown to provide major improvements for VoIP calls that have large one-way network delays (WNDs) in a scenario where they compete with many other VoIP calls in a simulated
802.11 network environment. A scaled down instance of this scenario was also shown to improve QoS for VoIP in a proof-of-concept experimental test-bed.

- **A Multimedia Categorization (MC) system** – While the existing 802.11e access categories distinguish VoIP from other types of traffic, there is no differentiation between VoIP sessions. The MC categorization allows certain VoIP sessions to be prioritized over other sessions, based on their E-Model R-factor scores. This facilitates the prioritization algorithm by allowing VoIP sessions to move up and down between three tiers while also providing a category for any background traffic present in the network. It’s important to note that whilst the algorithm developed worked well in both simulations and within the experimental test-bed, it would likely benefit from further testing.

- **A Traffic-Reclassification module** – This implements the actual prioritization in the MC hierarchical system in real-time by mangling the DSCP value in IP headers as the relevant RTP media packets pass through the AP. These DSCP values are then mapped to the appropriate MC category, which implements the MAC prioritization.

- **A Synchronization Test** – A verification test was proposed and verified which accurately checks whether endpoints and the AP in the test-bed are synchronized. This test provides clarity on the precision of synchronization of multiple wired and wireless nodes.

## 7.2 Future Work

While an appropriate combination of simulation and experimentation was used to address research questions and resulted in significant research contributions presented in this work, possible future extensions and improvements to the work are presented as follows:
7.2.1 Traffic Prioritization

7.2.1.1 EDCA Parameter Optimization

The iAP presented in this thesis has demonstrated that it is possible to dynamically re-assign traffic to different 802.11e categories in real-time. In this work, traffic was assigned to one of three multimedia categories, with a fourth traffic category available for background traffic. The parameters used for these categories could be changed to suit specific scenarios. With a considerable body of research carried out, and ongoing in the area of EDCA parameter optimization, there exists opportunities to improve the EDCA parameter selection process in the prioritization mechanism. Game theoretical approaches towards finding the optimal EDCA parameters present possible fruitful avenues for future work in this area.

7.2.1.2 Evolution of the VoIP Prioritization Algorithm

As the overall objective of the prioritization algorithm was to aid in the development of the proof-of-concept, it did not, as discussed above, undergo rigorous evaluation and further testing could yield better results. For example, the algorithm distinguishes between “good” and “bad” QoS, this could be extended to incorporate different degrees of “good” and “bad” QoS as reflected in the E-Model categories. Furthermore, as mentioned above, the PoC test-bed utilized delay values in generating R-factor values. More extensive testing of the practical approach, introducing more complex scenarios (e.g. multiple users) would benefit from an approach that incorporates both delay and loss similar to that done during simulations. This work focused on downlink delay as traffic travelling in the downlink direction typically experiences higher contention delays at the wireless MAC layer than uplink traffic, the algorithm could be extended to factor in uplink delay.

Finally, in its current state, the prioritization algorithm operates on one RTC session at a time (the session with the lowest R-factor that is below 80 and above 50). This may not be optimal when there are tens of active competing RTC sessions, therefore a more complex prioritization algorithm might be feasible which can simultaneously prioritize multiple RTC sessions.

7.2.1.3 QoE Subjective Testing

A possible further avenue for future work is to evaluate the impact on QoE of this iAP approach using subjective testing methods. Whilst the E-Model is a widely used planning tool and also widely used in literature to study the impact of delay on QoS,
subjective testing would provide an additional validation of the quality improvements brought about to VoIP using time synchronization. Subjective testing is however very time consuming and requires very specific test environments. This is especially so for conversational tests that would be required in this case, and there are few such laboratories that provide such facilities.

7.2.2 TAACCS – A new era in QoS/QoE Management

This work demonstrates how precise time synchronization distributed across wireless networks can facilitate a new breed of so-called Time Aware Applications, Computers, and Communication Systems (TAACCS). While the goal of the TAACCS group is much broader in scope by bringing time awareness all nodes and networks, the approach in this thesis represents a tangible example of the potential benefits. The PEL research group at NUI Galway are thus working closely with TAACCS on this and other related work.

7.2.3 SDN Development

With a broad industry effort to transform the current network architecture towards an SDN model, one of the key aspects of SDN is the centralized management of network devices via a common API, where data traffic management and shaping is abstracted to higher level software. While work in this thesis represents a move in this direction, as illustrated in Figure 3.3 in section 3.2.2 in Chapter 3, whereby a centralized and remote approach is taken towards network QoS management that can operate in real-time, further opportunities exist in this area particularly in the management of the iAP over the OpenFlow interface. This is also an active research area within the PEL group.

7.2.4 Extension to other types of Traffic

The MC prioritization system was demonstrated in this work for VoIP traffic. The mechanism may be extended to incorporate other types of traffic such as IPTV or Video Conferencing. For example, the concept of HbbTV whereby broadcast and broadband media are consumed on a single device is a growing phenomenon that requires QoS support that fits between VoIP and Best Effort. PEL at NUI Galway is also active in this space.
REFERENCES


[23] ITU-T Recommendation E.800, “Quality of telecommunications services: concepts, models, objectives, and dependability planning – Terms and definitions related to the quality of telecommunications services”, September 2008


[27] COST Memorandum of understanding Action IC 1003


[44] D. Leith, P. Clifford, D. Malone, and A. Ng, “TCP Fairness in 802.11e WLANs” IEEE Communications Letters, VOL. 9, NO. 12, December 2005


[46] M. Narbutt, M. Davis, “Experimental Tuning of AIFSN and CWmin Parameters to Prioritize Voice over Data Transmission in 802.11e WLAN Networks”, Pro-
ceedings of the 2007 International Wireless Communications & Mobile Computing Conference (IWCMC 2007), Hawaii, USA, August, 2007


[56] N. Hegde, A. Proutiere, and J. Roberts, “Evaluating the Voice Capacity of 802.11 WLAN under Distributed Control,” Proc. IEEE Workshop Local and Metropolitan Area Networks (LANMAN ’05), September, 2005


[67] PEL research group http://www.pel.it.nuigalay.ie


[76] Iptables Manpage 1.2.17.


[79] NS-2 The Network Simulator 2 ver. 2.31, 2007

[80] NS-3 The Network Simulator 3 ver. 3.11, 2011


[86] B. Bellata, C. Cano, M. Oliver, M. Meo, “Modelling the IEEE 802.11e EDCA for Parameter Optimization” Proceedings of the Performance Modelling and evalua-
tion of Hererogeneous Networks Conference (Het-Nets 06), Ilkley, UK, September 2006


Section VI

APPENDICES
APPENDIX 1

E-MODEL DEFAULT VALUES

The default values for all input parameters used in the algorithm of the E-model, are listed below [12]. It is strongly recommended to use these default values for all parameters that do not vary during planning calculation. If all parameters are set to the default values, the calculation results in a very high quality with a rating factor of $R = 93.2$.

Table 2 – Default values and permitted ranges for the parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Abbr.</th>
<th>Unit</th>
<th>Default value</th>
<th>Permitted range</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send loudness rating</td>
<td>SLR</td>
<td>dB</td>
<td>+8</td>
<td>0 ... +18</td>
<td>(Note 1)</td>
</tr>
<tr>
<td>Receive loudness rating</td>
<td>RLR</td>
<td>dB</td>
<td>+2</td>
<td>−5 ... +14</td>
<td>(Note 1)</td>
</tr>
<tr>
<td>Sidetone masking rating</td>
<td>STMR</td>
<td>dB</td>
<td>15</td>
<td>10 ... 20</td>
<td>(Notes 2, 4)</td>
</tr>
<tr>
<td>Listener sidetone rating</td>
<td>LSTR</td>
<td>dB</td>
<td>18</td>
<td>13 ... 23</td>
<td>(Note 2)</td>
</tr>
<tr>
<td>D-Value of telephone, send side</td>
<td>Ds</td>
<td>–</td>
<td>3</td>
<td>−3 ... +3</td>
<td>(Note 2)</td>
</tr>
<tr>
<td>D-Value of telephone, receive side</td>
<td>Dr</td>
<td>–</td>
<td>3</td>
<td>−3 ... +3</td>
<td>(Note 2)</td>
</tr>
<tr>
<td>Talker echo loudness rating</td>
<td>TELR</td>
<td>dB</td>
<td>65</td>
<td>5 ... 65</td>
<td></td>
</tr>
</tbody>
</table>
### Table 2 – Default values and permitted ranges for the parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Abbr.</th>
<th>Unit</th>
<th>Default value</th>
<th>Permitted range</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weighted echo path loss</td>
<td>WEPL</td>
<td>dB</td>
<td>110</td>
<td>5 ... 110</td>
<td></td>
</tr>
<tr>
<td>Mean one-way delay of the echo path</td>
<td>T</td>
<td>ms</td>
<td>0</td>
<td>0 ... 500</td>
<td></td>
</tr>
<tr>
<td>Round-trip delay in a 4-wire loop</td>
<td>Tr</td>
<td>ms</td>
<td>0</td>
<td>0 ... 1000</td>
<td></td>
</tr>
<tr>
<td>Absolute delay in echo-free connections</td>
<td>Ta</td>
<td>ms</td>
<td>0</td>
<td>0 ... 500</td>
<td></td>
</tr>
<tr>
<td>Number of quantization distortion units</td>
<td>qdu</td>
<td>–</td>
<td>1</td>
<td>1 ... 14</td>
<td></td>
</tr>
<tr>
<td>Equipment impairment factor</td>
<td>Ie</td>
<td>–</td>
<td>0</td>
<td>0 ... 40</td>
<td>(Note 5)</td>
</tr>
<tr>
<td>Packet-loss robustness factor</td>
<td>Bpl</td>
<td>–</td>
<td>4.3</td>
<td>4.3 ... 40</td>
<td>(Notes 3, 5)</td>
</tr>
<tr>
<td>Random packet-loss probability</td>
<td>Ppl</td>
<td>%</td>
<td>0</td>
<td>0 ... 20</td>
<td>(Notes 3, 5)</td>
</tr>
<tr>
<td>Burst ratio</td>
<td>BurstR</td>
<td>–</td>
<td>1</td>
<td>1 ... 8</td>
<td>(Notes 3, 6)</td>
</tr>
<tr>
<td>Circuit noise referred to 0 dBr-point</td>
<td>Nc</td>
<td>dBm0p</td>
<td>−70</td>
<td>−80 ... −40</td>
<td></td>
</tr>
<tr>
<td>Noise floor at the receive side</td>
<td>Nfor</td>
<td>dBmp</td>
<td>−64</td>
<td>–</td>
<td>(Note 3)</td>
</tr>
<tr>
<td>Room noise at the send side</td>
<td>Ps</td>
<td>dB(A)</td>
<td>35</td>
<td>35 ... 85</td>
<td></td>
</tr>
<tr>
<td>Room noise at the receive side</td>
<td>Pr</td>
<td>dB(A)</td>
<td>35</td>
<td>35 ... 85</td>
<td></td>
</tr>
<tr>
<td>Advantage factor</td>
<td>A</td>
<td>–</td>
<td>0</td>
<td>0 ... 20</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE 1** – Total values between microphone or receiver and 0 dBr-point.
**NOTE 2** – Fixed relation: LSTR = STMR + D.
**NOTE 3** – Currently under study.
**NOTE 4** – Equation 3-24 provides also predictions for STMR > 20 dB. However, such values can hardly be measured in a reliable way because the measurement device will mainly cover the acoustic coupling, and not the electrical one.
**NOTE 5** – If Ppl > 0%, then the Bpl must match the codec, packet size, and PLC assumed.
**NOTE 6** – E-model predictions for values of BurstR > 2 are only valid if the packet loss percentage is Ppl < 2%.

The 2000 version of this Recommendation provided an enhanced version of the E-model algorithm.

Due to the changes made in the 2000 version of this Recommendation, the resulting rating $R$ with all parameter values set to default has slightly changed (from $R = 94.2$ to $R = 93.2$). For practical planning purposes, however, this slight deviation should be considered insignificant.
APPENDIX 2

NETFILTER DATA FLOW

Netfilter (iptables) Packet Flow Graph: J. Engelhardt, Creative Commons, Linux
2.6.36+, Feb 2014
APPENDIX 3

LINKSYS DD-WRT 54GGL
SYSTEM DETAILS

The following is a summary of the technical details and facilities provided by Linksys\(^7\) of the Linksys router used for the development of the iAP in the experimental test-bed.

This router because of its large Flash memory and RAM lends itself to be upgraded to the full featured DD-WRT Mega pack. The WRT54GL and most of the other Linksys routers on the other hand does not support the megapack because they have limited memory. Specifically this router (WRT54G-TM) has a speedy 200Mhz processor, 32MB RAM and 8MB flash where as its step brother the WRT54GL has only 16MB RAM and 4MB flash

- **Adjust transmit power** -- Boost the wireless transmitter up to a whopping 251mW. *Note:* Please keep in mind that FCC regulations restrict

\(^7\) http://support.linksys.com/en-us/support/routers/WRT54G
wireless transmit levels for devices like these. Turn up the power incrementally until you get the level of service you need; you do not want to get noticed by the Fed. You could also "drown out" other wireless signals that use the same channel. A lot depends on the antenna configuration used and other factors, so just be careful. If you want more range, check out WDS below.

- **Afterburner** -- WRT54G-TM routers support SpeedBooster technology, and with wireless clients that support this feature, wireless performance can be increased significantly.

- **QoS** -- Use Quality of Service to prioritize types of network traffic. Let some applications have more bandwidth than others!

- **Dynamic DNS** -- Even without a static IP address from your service provider, you can access your router with a DNS name. DynDns.com provides a free service to associate a DNS name with your router (There are other DDNS providers, as well). If you need to get to your router from the outside world, dynamic DNS lets you have a consistent name that stays the same, even when its dynamic address changes (which it will, frequently).

- **Advanced routing features** -- Configure your router as a border router using BGP or OSPF routing protocols.

- **VLANs!** -- establish virtual network segments using VLAN IDs and create more sophisticated network configurations.

- **WDS** – Wireless Distribution System allows your router to talk to other routers as access points. This means you can extend the range of your network by letting routers talk with each other as bridges, resulting in wider network coverage.

- **WEP, WPA and WPA2** wireless security methods

- **RADIUS authentication** -- provide for strong network authentication by accessing an external server to verify credentials.

- **Virtual Private Network access** -- reach your internal network resources from the outside world using a secure VPN tunnel. (This requires the VPN firmware version)

- **UPnP** – Universal Plug and Play allows applications to automatically setup port forwarding.
- Command shell -- Add specific startup and/or firewall commands to be run when the router starts up to create even more customized configuration
APPENDIX 4

EXPERIMENTAL TESTBED iAP

CODE SAMPLE

The C code for the operation of the iAP is contained in this appendix. This is the script used in the experimental test-bed results in Sections 6.2 and 6.3 of Chapter 6 where R-factor values are calculated by the iAP based on the RTCP packets captured for each VoIP session. This code is compiled on the management station, and commands are passed via a pipe to the iAP. Functions are clearly marked in the code. The parsing of RTCP packets and calculation of delay values is carried out in the \texttt{calcR()} function. R-factors are calculated by the \texttt{rvalCalc()} function which is called from within \texttt{calcR()}. 
Calculates one-way delay for each session, then provides an accurate R-Value once test-bed is synchronized.

Padraig O Flaithearta

/*****************************************************************************/

#include <stdio.h>
#include <stdint.h>
#include <mysql/mysql.h>
#include <string.h>
#include <stddef.h>
#include <sys/time.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

/*****************************************************************************/

// FUNCTION TO MERGE AP TIMESTAMP (UNIX TIMEINTEGRAL & FRACTION) ***

double merger(unsigned long Sec, unsigned long Fra, int i){
    //To make sure 'second' integral values are 10 digits long,
    if(Fra<10){Sec=Sec*100000;}
    else if(Fra<100){Sec=Sec*10000;
    }
    else if(Fra<1000){Sec=Sec*1000;
    }
    else if(Fra<10000){Sec=Sec*100;
    }
    else if(Fra<100000){Sec=Sec*10;
double Time=(Sec*10)+Fra;
unsigned pow=10;
while(Fra >= pow)
    pow *= 10;
    Time=Sec*pow+Fra;
if(i==0)
    Time=Time/1000000;
else if(i==1)
    Time=Time/1000000000;
return Time;
}

*******
///*** FUNCTION FOR ERROR CHECKING CLOCK DRIFT ***
*******

[+] double verif(double val){
    
}

*******
///*** FUNCTION TO CALUCLATE R-FACTOR ***
*******

[+] double rvalCalc(double delay){
    
}

*******
///*** FUNCTION TO RUN TCPDUMP AND CALCULATE ONE-WAY DELAYS ***
*******

double calcR(char *ip1, char *ip2, int session){
    int sess=session;
    FILE *in;
    FILE * updelayptr;
    FILE * dndelayptr;
    FILE * uprvalptr;
    FILE * dnrvalptr;
    //files for statistics
const char *upDelayChar = "DelayUp.dat";
const char *dnDelayChar = "DelayDn.dat";
const char *upRvalChar = "RvalUp.dat";
const char *dnRvalChar = "RvalDn.dat";

extern FILE *popen();
char buff[556];
char *delim = " ";
char token[10000][300];
int count=0, mycount, counter, whilecount=0;

char *tcpdumpcommand = "ssh root@192.168.1.1 tcpdump -c 4 -s0 -tt -T rtcp udp -x port 5061 or port 5063 or port 5065 or port 5067 or port 5069 and host ";
char *ipone=ip1; //"192.168.1.100"; //Wired Host 1
char *iptwo=ip2; //"192.168.1.102"; //Wired Host 2
int sizer(sizeof(tcpdumpcommand)+sizeof(ipone));
char command[sizer];

//char *ntpcclientcommand = "ssh root@192.168.1.1 cd /usr/sbin/ ntpclient -h 140.203.16.5";
char *chPtr;
//
//add host IP address at end of string

//Temporary TS value
unsigned long apSecDec, apFraDec, ntpSecDec, ntpFraDec;
double dl, d2, d3, d4, apTime, ntpTime, dlsrDec, ntpTemp, upDelay, dnDelay, upRval, dnRval;
double t0, t1, t2, t3, t4, t5, t6, t7, t8, t9;
if(sess==1){
    printf("n
n
n***nChecking delay and QoS R-Value of Session 1.... n\tWN IP Address: %s n***n\n", ipone);
    command[0] = '\0';
    strcat(command,tcpdumpcommand);
    strcat(command,ipone);
    in = popen(command, "r");
}
else if(sess==2){
    printf("n
n
n***nChecking delay and QoS R-Value of Session 2.... n\tWN IP Address: %s n***n\n", iptwo);
}
command[0] = '\0';
strcat(command, tcpdumpcommand);
strcat(command, iptwo);
in = popen(command, "r");

//checkSrc="IP 192.168.1.101"; //To set source for Uplink/Downlink
}

// GET INPUT STREAM: read the output of tcpdump pipe, one line at a time
while (fgets(buff, sizeof(buff), in) != NULL) {

// IF line contains "sr", get timestamp
if (strstr(buff, " sr ") != NULL) {// && checkSrc != NULL)

//printf("\nline is %s\n", buff);
    int mycount = 0;
    chPtr = strtok(buff, delim);
    while (chPtr != NULL) {
        strcpy(token[count], chPtr);
        if (mycount == 0) {
            int i = 0;
            char * pch;
            sec = chPtr;
            pch = strtok(sec, ", . -");
        //Tokenise AP timestamp to seconds and Fractions
            while (pch != NULL && i < )
                {
            //Get Second part of timestamp
                if (i == ) {
                    sec = pch;
                    apSecDec = atoi(sec);
                    apSecDec = apSecDec * 2208988800;
                }
            pch = strtok(NULL, ", . -
                     i++;
                     whileCount++;
                }
        } else if (i == ){

            //Get Fraction part of timestamp
            fra = pch;
            apFraDec = atoi(fra);
        }
apTime = merger(apSecDec, apFraDec, 0);

// printf("%napTime is \t%f\t%d", apTime, whilecount);
    if(whilecount==1){t2=apTime;
    else
    if(whilecount==6){t5=apTime;
    else
    if(whilecount==11){t8=apTime;

    apTime=0;
pch = strtok (NULL, " ", ", -");
i++;
whilecount++; 
}
    mycount++;
}
else{mycount++;}
}
chPtr = strtok(NULL, delim);

//points to next token in string
}
//end while 3
}
// IF line contains 0x0020, get NTP timestamp
else if(strstr(buff, "0x0020") != NULL){

// printf("\nline is %s\n", buff);
    int mycount = 0;
    chPtr = strtok(buff, delim);
    while(chPtr != NULL) {
        strcpy(token[count],chPtr);
        if (mycount==3){mycount++;}
        chPtr = strtok(NULL, delim);
    //points to next token in string
        if(mycount==3){
            ntpSec=chPtr;
            mycount++;
        }
    }
    if(mycount==3){
        strcat(ntpSec, chPtr);
    }
ntpSecDec = strtol(ntpSec, 0, 16);

//Convert val from Hex to Decimal

//Add 1 hour due to timezone setting on AP London GMT
mycount++;
whilecount++;
}
if(mycount==7){
    ntpFra=chPtr;
    mycount++;
}
if(mycount==9){
    strcat(ntpFra, chPtr);
    ntpFraDec = strtol(ntpFra, 0, 16);
    //Convert val from Hex to Decimal
    ntpTemp=ntpFraDec/4294967296.0;
    //printf("\nntp is %f ", ntpTemp);
    ntpTime=ntpSecDec+ntpTemp;
}
//printf("\nntpTime is\t%lu\t\", ntpTime, whilecount);
    if(whilecount==3){t1=ntpTime;}
    else if(whilecount==8){t4=ntpTime;}
    else if(whilecount==13){t7=ntpTime;}
    mycount++;
    whilecount++;
}
else{mycount++;}
//end inner while

//end else if
else if(strcmp(buff, "0x0040") != NULL){

//printf("\nline is %s", buff);
    int mycount = 0;
    chPtr = strtok(buff, delim);
    while(chPtr != NULL) {
        strcpy(token[count], chPtr);
        if (mycount==0){mycount++;}
        // 207
    }
chPtr = strtok(NULL, delim);

// points to next token in string
if (mycount == 7) {
    dlsr = chPtr;
    mycount++;
}
if (mycount == 9) {
    strcat(dlsr, chPtr);
    printf("\nslr is %s", chPtr);
    dlsrDec = strtol(dlsr, 0, 16);
    // convert char hex to uInt decimal
    printf("\nds is %f", dlsrDec);
    dlsrDec = dlsrDec / 65536;
    printf("\nds is %f", dlsrDec);
    if (whilecount == 9) {
        t3 = t4 - dlsrDec;
        printf("\nDLSR is t%f t%dt%f", t4 - dlsrDec, whilecount, dlsrDec);
    }
    else if (whilecount == 14) {
        t6 = t7 - dlsrDec;
        printf("\nDLSR is t%f t%dt%f", t7 - dlsrDec, whilecount, dlsrDec);
    }
    mycount++;
    whilecount++;
}
else {
    mycount++;
}
// end inner while
}
// end else if
// end outer while

// NTP - DSLR

// CALCULATE DELAYS
    d1 = t2 - t1, d2 = t3 - t2, d3 = t5 - t4, d4 = t6 - t5;
    d1 = verif(d1); d2 = verif(d2); d3 = verif(d3); d4 = verif(d4);

// Error check - (nanosecond drift negative values shite)
upDelay=d1+d2, dnDelay=d3+d4;

// Print Results to Screen For Checking
printf("Results for Session number:%d, are as follows:\n", sess);
printf("***** Client ***** AP ***** Client *****\n");
printf("t1, t2, t3, t4, t5, t6, t7, t8\n");
printf("***** Cl\n");
printf("t %f t %f t %f t %f t %f t %f t %f t %f\n", d1, d2, d3, d4, upDelay, dnDelay);
upRval=rvalCalc(upDelay);
dnRval=rvalCalc(dnDelay);
printf("Uplink Delay:%fs, Downlink Delay:%fs\n", upDelay, dnDelay);
printf("Uplink R-Value is:%f Downlink R-Value is: %f\n", upRval, dnRval);
// close the pipe
pclose(in);

// flush output buffer
fflush(stdout);

updelayptr= fopen(upDelayChar,"a+");
dndelayptr= fopen(dnDelayChar,"a+");
uprvalptr= fopen(upRvalChar,"a+");
dnrvalptr= fopen(dnRvalChar,"a+");
if(sess==1){
    fprintf( updelayptr, "%f\t", upDelay);
    fprintf( dndelayptr, "%f\t", dnDelay);
    fprintf( uprvalptr, "%f\t", upRval);
    fprintf( nrvalptr, "%f\t", nrval);
} else if(sess==2){
    fprintf( updelayptr, "%f\n", upDelay);
    fprintf( dndelayptr, "%f\n", dnDelay);
    fprintf( uprvalptr, "%f\n", upRval);
    fprintf( nrvalptr, "%f\n", nrval);
}
fflush( updelayptr );
fflush( dndelayptr );
fflush( uprvalptr );
fflush( dnrvalptr );
fclose( updelayptr );
fclose( dndelayptr );
fclose( uprvalptr );
fclose( dnrvalptr );

//Return Downlink R-Value
    return dnRval;
}

//close Function

extern FILE *popen();
char *MC1="MC1";
char token[10000][300];
//This line needs to be here for some unknown reason...

//char *ipone1="192.168.1.100";
char *iptab1 = "ssh root@192.168.1.1 iptables -t mangle -I FORWARD -s ";
char *iptab2 = " -j DSCP --set-dscp ";
int iptabsize=sizeof(iptab1)+sizeof(ipone1)+sizeof(iptab2)+sizeof(MC1);
char cmd[iptabsize];
cmd[0] = '\0';
strcat(cmd,iptab1);
strcat(cmd,ipone1);
strcat(cmd,iptab2);
strcat(cmd, MC1);
printf("\n IPTABLES command is: %s\n", cmd);
    return 1;
}

//Fn to prioritize packets from IP address; 'IP'
/+ int prioAlgrm(char *IP, double R, char *MC)[}

//******
```c
int main() {
    int s1=1;
    int s2=2;
    char *ip1 = "192.168.1.106"; //Wired Host 1
    char *ip2 = "192.168.1.107"; //Wired Host 2
    char *MC3="46", *MC2="18", *MC1="10", *MC0="2";
    //define multimedia category DSCP integer values
    char *IP;
    double R=0.0;
    int currMC=1, val=0;
    while(currMC<4 || currMC>1) {

        // CALL DELAY/R-VAL CALCULATION Fn, RESULT IS DOWNLINK R-VAL FOR THAT SESSION
        double R1=calcR(ip1, ip2, s1);
        double R2=calcR(ip1, ip2, s2);

        if(R1<R2){IP=ip1;R=R1;}
        //Find lowest R-Value, ITS THEN PASSED INTO PRIORITIZATION ALGORITHM
        //Prioriti
        else if(R2<R1){IP=ip2;R=R2;}
        printf("\nthe Downlink R-values are for Session 1: %f\n   Session 2: %f\n   R is %f", R1, R2, R);

        if(R<80 && R>50) {
            if(currMC==1) {
                val=tuner(IP, MC2);
                currMC=currMC+val;
                printf("\nCurrent MC is %d", currMC);
            }
            else if(currMC==2) {
                val=tuner(IP, MC3);
                currMC=currMC+val;
                printf("\nCurrent MC is %d", currMC);
            }
            else if(currMC==3) {
                val=tuner(IP, MC1);
                currMC=currMC+val;
                printf("\nCurrent MC is %d", currMC);
            }
        }

    }
}
```
else if(currMC==3){
    printf("\n\t\t Call is in top Multimedia Catego-\
    ry\n");
}
else if(R>80){
    printf("\n\t\t*** THE LOWEST R-VALUE RECORDED IS\
    ABOVE 80, SO ALL CALLS HAVE GOOD QOS!!! *** \n");
}
else if(R<50){
    printf("\n\t\t*** THE LOWEST R-VALUE RECORDED IS\
    BELOW 50, SO THE CALL IS BEYOND THE SCOPE OF PRIORITIZATION ALGORITHM!!! ***\
    \n");
}
else{
    printf("\n\t\t***ERROR***\n");
}
}