Western Graduate&PostdoctoralStudies

Western University

Scholarship@Western

Electronic Thesis and Dissertation Repository

11-18-2016 12:00 AM

Radio Resource Management Optimization For Next Generation Wireless Networks

Karim Ahmed Hammad
The University of Western Ontario

Supervisor
Dr. Abdallah Shami
The University of Western Ontario Joint Supervisor
Dr. Serguei Primak
The University of Western Ontario

Graduate Program in Electrical and Computer Engineering
A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

© Karim Ahmed Hammad 2016

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the Systems and Communications Commons

Recommended Citation

Hammad, Karim Ahmed, "Radio Resource Management Optimization For Next Generation Wireless Networks" (2016). *Electronic Thesis and Dissertation Repository*. 4224.
https://ir.lib.uwo.ca/etd/4224

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlswadmin@uwo.ca.

Abstract

The prominent versatility of today's mobile broadband services and the rapid advancements in the cellular phones industry have led to a tremendous expansion in the wireless market volume. Despite the continuous progress in the radio-access technologies to cope with that expansion, many challenges still remain that need to be addressed by both the research and industrial sectors. One of the many remaining challenges is the efficient allocation and management of wireless network resources when using the latest cellular radio technologies (*e.g.*, 4G). The importance of the problem stems from the scarcity of the wireless spectral resources, the large number of users sharing these resources, the dynamic behavior of generated traffic, and the stochastic nature of wireless channels. These limitations are further tightened as the provider's commitment to high quality-of-service (QoS) levels especially data rate, delay and delay jitter besides the system's spectral and energy efficiencies. In this dissertation, we strive to solve this problem by presenting novel cross-layer resource allocation schemes to address the efficient utilization of available resources versus QoS challenges using various optimization techniques.

The main objective of this dissertation is to propose a new predictive resource allocation methodology using an agile ray tracing (RT) channel prediction approach. It is divided into two parts. The first part deals with the theoretical and implementational aspects of the ray tracing prediction model, and its validation. In the second part, a novel RT-based scheduling system within the evolving cloud radio access network (C-RAN) architecture is proposed. The impact of the proposed model on addressing the long term evolution (LTE) network limitations is then rigorously investigated in the form of optimization problems.

The main contributions of this dissertation encompass the design of several heuristic solutions based on our novel RT-based scheduling model, developed to meet the aforementioned objectives while considering the co-existing limitations in the context of LTE networks. Both analytical and numerical methods are used within this thesis framework. Theoretical results are validated with numerical simulations. The obtained results demonstrate the effectiveness of our proposed solutions to meet the objectives subject to limitations and constraints compared to other published works.

Keywords: Ray Tracing, LTE, C-RAN, OFDMA, Resource Allocation, QoS, Delay Jitter, Energy-Efficient Communications

Co-Authorship

The following thesis contains material from previously published papers and manuscripts submitted for publication that have been co-authored by Karim Hammad, Dr. Abdallah Shami, Dr. Serguei Primak, Dr. Maysam Mirahmadi, Mohamad Kalil, and Abdallah Moubayed. All the research, developments, simulations, and work presented here were carried out by Karim Hammad under the guidance of Dr. Abdallah Shami and Dr. Serguei Primak. Original manuscripts which make up parts of Chapters 3-6 in this thesis were also written by Karim Hammad.

Publications

[J1] K. Hammad, S. Primak, M. Kalil, and A. Shami, "QoS-Aware Energy-Efficient Downlink Predictive Scheduler for OFDMA-based Cellular Devices," IEEE Transactions on Vehicular Technology, 2016.

[J2] K. Hammad, A. Moubayed, A. Shami, and S. Primak, "Analytical Approximation of Packet Delay Jitter in Simple Queues," IEEE Wireless Communications Letters, 2016.

[J3] K. Hammad, S. Primak, A. Moubayed, and A. Shami, "Investigating the Energy-Efficiency/Delay Jitter Trade-off for VoLTE in LTE Downlink," *Submitted to the IEEE Transactions on Vehicular Technology*.

[J4] K. Hammad, A. Shami, S. Primak, and A. Moubayed,"QoS-Aware Energy and Jitter-Efficient Downlink Predictive Scheduler for Heterogeneous Traffic LTE Networks," *Submitted to the IEEE Transactions on Mobile Computing*.

[C1] K. Hammad, M. Mirahmadi, S. Primak, and A. Shami, "On a Throughput-Efficient Look-Forward Channel-Aware Scheduling," IEEE International Conference on Communications (ICC), June 2015

To the memory and love of my best friend, best teacher and best father, Ahmed Hammad,
who could not wait to see this thesis completed

Acknowledgements

With none but Allah is the direction of my affair to a right issue, in him I trust and unto him I repent. My thanks goes to God Almighty who enlighten the path and allowed the sources of knowledge for me, and gave me the patience, will and certainty.

I am deeply and forever indebted to my parents. You are the greatest source of love, encouragement, and inspiration. Thank you, Mom and Dad for the love and support throughout my entire life, and for all of the sacrifices you have made on my behalf. Without both of you I would never have advanced a single step ahead. This project is dedicated to the loving memory of my father, may God bless his soul, who could not wait to see his dream come true.

My greatest recognition and sincere gratitude go to my supervisors and mentors Dr. Abdallah Shami and Dr. Serguei Primak for their boundless amount of support, motivation, guidance, and the advices they provided me throughout the course of my PhD journey. Frankly, I consider myself very lucky to be their student. They inspired both my academic and personal aspects of my life. I could not imagine better support than having two smart, decent and patient characters like them. They made my PhD experience at Western University successful, amazing and unforgettable. I appreciate every single minute they gave me from their precious time, and all the countless contributions they made to this work. Without them this work would have never materialized.

I would also like to thank my examiners: Dr. Raveendra K. Rao, Dr. Dimitrios Makrakis, Dr. Arafat Al-Dweik, and Dr. Hanan Lutfiyya for taking the time to review and examine my thesis.
My thanks have to be extended to my Master's thesis supervisor, Dr. Khaled Shehata, for his mentorship since the early beginning of my undergraduate studies and until putting me on the beginning of the research road. I would like to thank Dr. Salwa El-Ramly and Dr. Mohamed Abul-Dahab, my Master's thesis co-supervisors and my undergraduate lecturers, for the huge knowledge I gained from them.

My sincere gratitude goes to my sister, Yasmin for being always my strong support and for the huge love she gave me throughout my entire life. A heartfelt thanks to my lovely wife Hager for her continued support, love, and for being my best friend and great companion who helped me get through the unexpected troubles of research in the most positive way.

Words cannot describe how lucky I am to have you in my life.

Special thanks go to my colleagues and friends at Western university who were great company during my PhD journey. For most, a BIG thanks to Dr. Maysam Mirahmadi, Mohamed Kalil and Abdallah Moubayed for their valuable collaboration and contribution to this dissertation, and to Mohamed Abu Sharkh who provided me with a great linguistic support while writing this thesis manuscript. He was always there for me and ready to help. Also, I would like to extend my thanks to my friends: Khaled Alhazmi, Abdulfattah Noorwali, Elena Uchiteleva, Oscar Filio, Aidin Reyhani, Dan Wallace, Bradley de Vlugt, Emad Aqeeli, Manar Jammal, Hassan Hawilo, Mohamed Hussein, Mohammad Noor Injadat, Fuad Shamieh, Anas Saci, M. Ajmal Khan, and Dr. Khalim Meerja.

Last but not least, to all my friends around the world, thank you for your well-wishes, phone calls, e-mails, and being there whenever I needed a friend. For most, I would like to thank Dr. Ahmed Refaey, Dr. Saleh Eissa, Dr. Safa Gasser, Dr. Hanady Hussein, Aly Swelam, Omar Swelam, Hesham Swelam, Shady Ashraf, Sherif Rady, Hesham Amr, Ahmed Bassioni, Mohamed Magdy, Hesham Ahmed, Ahmed Afifi, and Mohamed Farouk.

Table of Contents

Abstract . . . . **ii**

Dedication . . . . **v**

Acknowledgements . . . . **vi**

Table of Contents . . . . **viii**

List of Tables . . . . **xii**

List of Figures . . . . **xiii**

Acronyms . . . . **xvi**

1 Introduction **1**
- 1.1 Background and Motivation 2
- 1.2 State-of-the-Art and Thesis Outline 4
- 1.3 Contributions of the Thesis 5
  - 1.3.1 Contributions of Chapter 2 5
  - 1.3.2 Contributions of Chapter 3 6
  - 1.3.3 Contributions of Chapter 4 6
  - 1.3.4 Contributions of Chapter 5 7
  - 1.3.5 Contributions of Chapter 6 7

2 Ray Tracing Propagation Prediction: From Theory to Implementation . . . 9
2.1 Introduction 9
2.2 Applications, Motivation and Recent Trends 11
2.3 Ray Tracing Algorithms 12
2.3.1 Shooting and Bouncing Rays Method 14
2.3.2 Image Method 18
2.3.3 Electric Field Evaluation 19
2.4 Ray Tracing Acceleration Techniques 21
2.4.1 Binary Space Partitioning 22
2.4.2 Space Volumetric Partitioning 23
2.4.3 Angular Z-Buffer 24
2.5 Ray Tracing Engine Architecture 24
2.6 MATLAB Ray Tracer 26
2.7 Numerical Simulations and Verification with Commercial Software 31
2.7.1 Line-of-Sight Test with Fixed Receiver Distance 31
2.7.2 Free Space Path Loss Test 35
2.7.3 Two Ray Model Test 36
2.7.4 Urban Scenario Propagation Test 38
2.8 Chapter Summary 43

3 On a Throughput-Efficient Look-Forward Channel-Aware Scheduling 44
3.1 Introduction 44
3.2 Link Adaptation Analysis 46
3.3 Ray Tracing Based Prediction 50
3.4 Channel-Aware Scheduling 51
3.4.1 Conventional Channel-Aware Scheduler 52
3.4.2 The Proposed Optimal RT-Assisted Scheduler 52
3.4.3 Heuristic RT-Assisted Scheduler 55
3.5 Numerical Results 58
3.6 Chapter Summary 61

4 QoS-Aware Energy-Efficient Downlink Predictive Scheduling for OFDMA-Based Cellular Devices 62
4.1 Introduction 62
4.1.1 Related Work 63
4.1.2 Scope and Contribution 64
4.2 System Description 66
4.2.1 System Model 66
4.2.2 UE Circuit Power Consumption 71
4.2.3 Channel Model 72
4.3 Proposed Predictive Scheduling System 75
4.4 Optimal Scheduler 78
4.4.1 General Formulation 78
4.4.2 Penalty Method-Based Formulation 80
4.4.3 Practical ON-OFF Formulation 82
4.5 Heuristic Scheduler 84
4.5.1 Heuristic Algorithm 85
4.5.2 Complexity Evaluation 87
4.6 Numerical Results 88
4.6.1 Scenario 1: QSBR Channel 91
4.6.2 Scenario 2: RT-based Channel 99
4.7 Chapter Summary 102

5 Modelling and Optimizing Delay Jitter in Communication Networks 105
5.1 Introduction 105
5.2 Analytical Approximation of Packet Delay Jitter in Simple Queues 106
5.2.1 Background 106
5.2.2 Jitter modelling framework 107
5.2.3 Underutilized queue 108
5.2.4 Heavy loaded queue 111
5.2.5 Bridging case 113
5.2.6 Numerical Simulations 114
5.3 Investigating the Energy-Efficiency/Delay Jitter Trade-off for VoLTE Traffic in LTE Downlink 117
5.3.1 Background 117
5.3.2 System Model and Problem Formulation 119
5.3.3 Heuristic Algorithms 125
5.3.4 Numerical Simulations 129
5.4 Chapter Summary 134

6 Studying the Energy/Delay Jitter Efficiencies Trade-off in Heterogeneous Traffic LTE Networks with QoS Awareness . . . 135
6.1 Introduction . . . 135
6.1.1 Research Problem . . . 136
6.1.2 Related Work . . . 137
6.1.3 Scope and Contribution . . . 143
6.2 System Model . . . 145
6.2.1 Utility-Based Scheduling . . . 148
6.2.2 Energy-Efficient Scheduling . . . 152
6.3 Delay Jitter Background . . . 153
6.4 Problem Formulation . . . 156
6.5 Heuristic Solutions . . . 160
6.5.1 SRV-Based Schedulers . . . 161
6.5.2 PRV-Based Schedulers . . . 167
6.6 Numerical Results . . . 172
6.7 Chapter Summary . . . 183

7 Conclusion . . . 185
7.1 Thesis Summary . . . 185
7.2 Future Work . . . 189
7.2.1 Interference-Aware Scheduling for LTE-A D2D Communication . . 189
7.2.2 Virtualization in Next Generation Radio Access Network . . . 190
7.2.3 On The Application of Ray Tracing Prediction in Physical layer Security . . . 191

References . . . 193

Curriculum Vitae . . . 205

List of Tables

3.1 Heuristic Algorithm For RT-Based Scheduler 56

4.1 List of MCS Indices [1] 74
4.2 Complexity comparison 95
4.3 Simulation parameters 100

5.1 Heuristic Algorithm 1 126
5.2 Heuristic Algorithm 2 128

6.1 Some of the recent EE schedulers reported in literature 139
6.2 SRV-CO algorithm 162
6.3 SRV-PO algorithm 165
6.4 PRV-CO algorithm 170
6.5 PRV-PO algorithm 171

List of Figures

2.1 Three-dimensional scene rendered using real-time FPGA ray tracer [2] . . . 13
2.2 SBR method 14
2.3 Geodesic sphere with tessellation frequency of 4 16
2.4 A MATLAB generated geodesic sphere with tessellation frequency of 4 . . 16
2.5 Ray reflections in image method 18
2.6 Facet reflection space 19
2.7 Ray tracing system architecture 25
2.8 Flow chart describing the MATLAB implementation of the ray tracer 28
2.9 Reading the Wireless Insite 3D map in MATLAB 29
2.10 Snap shot for a sample of geodesic rays vs. corrected rays deviation angle . 32
2.11 Geodesic rays vs. corrected rays weighting factors summation at different receiver points along the surface of a sphere 33
2.12 The distribution of weighting factors for the geodesic rays on a geodesic sphere centered at the transmitter 34
2.13 The distribution of weighting factors for the corrected geodesic rays on a geodesic sphere centered at the transmitter 34
2.14 3D view for the Matlab/Wireless Insite FSPL test 35
2.15 FSPL prediction using MATLAB ray tracer vs. Wireless Insite with respect to the theoretical model 36
2.16 3D view for the Matlab/Wireless Insite two ray model test 37
2.17 Path loss prediction for two ray model using MATLAB ray tracer vs. Wireless Insite with respect to the theoretical model 38
2.18 Google map view for the urban scenario environment 39
2.19 Wireless Insite/MATLAB urban scenario simulation environments 40
2.20 The urban scenario path loss prediction results for MATLAB vs. Wireless Insite 41
2.21 Debugging the Wireless Insite resolution problem 42
2.22 Received ray paths traced by the MATLAB and Wireless Insite at receiver point number 5 42

3.1 Outage probability at different mobile velocities and T values 50
3.2 Proposed RT-assisted vs conventional MT-Max/Min scheduler 53
3.3 Throughput performance comparison 59
3.4 Computational complexity analysis 60

4.1 Energy efficiency for downlink predictive scheduling 65
4.2 C-RAN based model 67
4.3 Simplified block diagram for LTE UE downlink physical layer processing chain 71
4.4 Ray tracing based downlink scheduler system 76
4.5 Heuristic algorithm flowchart for the proposed scheduler 86
4.6 3D ray tracing experiment for part of Ottawa city 89
4.7 SNR correlation coefficient for a Gaussian process at different mobile speeds and carrier frequency of 2.6GHz 91
4.8 Energy efficiency comparison for the proposed vs. the GRA scheduler . . . 92
4.9 Quasi-instantaneous rate satisfaction for the proposed vs. the GRA scheduler 93
4.10 The proposed vs. the GRA scheduler capability for satisfying the 13.6kb/sec connection 94
4.11 UE battery lifetime for the proposed vs. the GRA scheduler 94
4.12 UE's buffer queue stability 96
4.13 UE's average achieved rate 97
4.14 Distribution of the total cell rate in the case of 15 UEs 98
4.15 Average energy consumption per UE 99
4.16 UE's buffer queue stability 101
4.17 Snap shot for 5 sec of the QSBR and RT channels 101
4.18 UE's average achieved rate 102
4.19 Average energy consumption per UE 103

5.1 Packet delay jitter evaluation for the *M/M/1* queue 114
5.2 Packet delay jitter evaluation for the *D/M/1* queue 115
5.3 Packet delay jitter evaluation for the *M/ExpCorr/1* queue 116
5.4 Packet delay jitter evaluation for underutilized *M/GE/1* queue 116
5.5 Packet delay jitter evaluation for heavy loaded *M/GE/1* queue 117
5.6 System model 120
5.7 Energy efficiency 130
5.8 Average packet delay jitter 131
5.9 EE vs. packet delay jitter trade-off 132
5.10 Average packet delay 133
5.11 JFI 133

6.1 System model 146
6.2 Proposed utility-based energy/jitter efficient packet scheduler 150
6.3 Sliding window predictive scheduling mechanism 168
6.4 EE performance for VoIP UEs 173
6.5 EE performance for Video UEs 173
6.6 EE performance for FTP UEs 174
6.7 Average packet delay jitter for VoIP UEs 175
6.8 Average packet delay for VoIP UEs 176

6.9 EE/delay jitter trade-off for VoIP UEs 178
6.10 Average throughput for Video UEs 179
6.11 Average throughput for FTP UEs 179
6.12 JFI for VoIP UEs 180
6.13 JFI for Video UEs 181
6.14 JFI for FTP UEs 181

Acronyms

AZB	*Angular Z-Buffer*
AMR	*Adaptive Multi-Rate*
ATM	*Asynchronous Transfer Mode*
ADSP	*Advanced Digital Signal Processor*
ADC	*Analog-to-Digital Converter*
BSI	*Buffer State Information*
BSP	*Binary Space Partitioning*
BB	*Baseband*
BPF	*Band Pass Filter*
BBU	*Baseband Unit*
BLER	*Block Error Rate*
BER	*Bit Error Rate*
BS	*Base Station*
BIP	*Binary Integer Programming*
CSI	*Channel State Information*
C-RAN	*Cloud Radio Access Network*
CR	*Cognitive Radio*
CRC	*Cyclic Redundancy Check*
CPU	*Central Processing Unit*
CDMA	*Code Division Multiple Access*
CDF	*Cumulative Distribution Function*
DSP	*Digital Signal Processor*
DRX	*Discontinuous Reception*
D2D	*Device-to-Device*
EDGE	*Enhanced Data Rates for GSM Evolution*
EE	*Energy Efficiency*

EM	*Electromagnetic*
eNB	*Evolved Node-B*
EXP	*Exponential Rule*
EARA	*Energy Aware Resource Allocation*
FDMA	*Frequency Division Multiple Access*
F-LWDF	*Fair-Largest Weighted Delay First*
FDTD	*Finite Difference Time Domain*
FDD	*Frequency Division Duplexing*
FFT	*Fast Fourier Transform*
FPGA	*Field Programmable Gate Array*
FSPL	*Free Space Path Loss*
GSM	*Global Systems for Mobile Communications*
GPRS	*General Packet Radio Services*
GBR	*Guaranteed Bit Rate*
GPU	*Graphics Processing Units*
GIS	*Geographic Information Systems*
GUI	*Graphical User Interface*
GRA	*Green Resource Allocation*
GTD	*Geometrical Theory of Diffraction*
GE	*Gilbert-Elliot*
HPC	*High Performance Computing*
HoL	*Head-of-Line*
IoT	*Internet of Things*
IP	*Internet Protocol*
IID	*Independent and Identically Distributed*
IBP	*Interrupted Bernoulli Process*
InP	*Infrastructure Provider*
ISM	*Industrial, Scientific and Medical*
LWDF	*Largest Weighted Delay First*
LNA	*Low Noise Amplifier*
LPF	*Low Pass Filter*
LOS	*Line of Sight*

LTE	*Long Term Evolution*
MIMO	*Multiple-Input Multiple-Output*
MAC	*Media Access Control*
MT	*Maximum Throughput*
MCS	*Modulation and Coding Scheme*
MCC	*Mobile Cloud Computing*
MATP	*Maximum Allowable Transmit Power*
MMBP	*Markov-Modulated Bernoulli Process*
MDP	*Markov Decision Process*
MNO	*Mobile Network Operator*
M-LWDF	*Modified-Largest Weighted Delay First*
NGBR	*Non Guaranteed Bit Rate*
OFDMA	*Orthogonal Frequency Division Multiple Access*
OFDM	*Orthogonal Frequency Division Multiplexing*
PHY	*Physical Layer*
PDSCH	*Physical Downlink Shared Channel*
PUSCH	*Physical Uplink Shared Channel*
PDF	*Probability Density Function*
PF	*Proportional Fair*
PU	*Primary User*
PO	*Physical Optics*
QoS	*Quality of Service*
QCI	*Quality of Service Class Identifier*
QSBR	*Quasi-Static Block Rayleigh*
RT	*Ray Tracing*
RB	*Resource Block*
RAM	*Random Access Memory*
RF	*Radio Frequency*
RTP	*Real-time Transport Protocol*
SU	*Secondary User*
SBR	*Shooting and Bouncing Rays*
SNR	*Signal to Noise Ratio*

SVP	*Space Volumetric Partitioning*
TDMA	*Time Division Multiple Access*
TTI	*Transmission Time Interval*
UMTS	*Universal Mobile Telecommunication System*
UE	*User Equipment*
UDP	*User Datagram Protocol*
V2V	*Vehicle-to-Vehicle*
VGA	*Variable Gain Amplifier*
VoLTE	*Voice Over Long Term Evolution*
V2I	*Vehicle to Infrastructure*
VANET	*Vehicular Ad-Hoc Networks*
VoIP	*Voice Over Internet Protocol*
WCDMA	*Wide-band Code Division Multiple Access*
Wi-Fi	*Wireless Fidelity*
3GPP	*3rd Generation Partnership Project*

Chapter 1
Introduction

The growing demand for today's high density and high volume mobile data traffic services is leading to high sophistication in the design and capabilities of both, the cellular devices and the wireless networks. Today's statistics [3] show that over 1 billion mobile users around the globe are intensely using the social networking media, streaming, and gaming services on a daily basis. Nevertheless, market analysts predict that the growth will remain in the forseen future. This number, in conjuction with the increasing demand of high data rates and user/service mobility generated by new diversified wireless applications, creates serious challenges to the wireless systems' designers. As a result, diligent efforts are currently in place to shift the cellular world from the currently deployed fourth generration (4G) system (*i.e.*, Long-Term Evolution (LTE)) to a more sophisticated technology, the so called fifth generation (5G) [4]. According to [5], the 5G wireless network is expected to achieve 1000 times higher capacity, 10 times higher spectral efficiency (*i.e.*, 10Gb/s for users located at the center of a cell, and 5Gb/s for users located at a cell's edge), 5 times reduction to the end-to-end delay, and 10 times longer battery life compared to the current LTE networks.

Despite the fact that the aggregation of cutting-edge technologies (*e.g.*, massive multiple-input and multiple-output (MIMO), visible light communication, millimeter waves, femto-cellular architecture, spatial modulation and device-to-device communication) within the future 5G system can potentially meet the evolving challenges facing wireless networks, the current LTE system is showing satisfactory ability to cope with the current demand [6]. Consequently, extensive research efforts [7, 8] and industrial investments are made towards achieving a fully functional, spectral and energy efficient LTE system from both the network and user equipment's sides. Considering these facts, in this dissertation, we aim to present a different vision and opportunity for the medium access control (MAC)- physical (PHY) cross-layer design and optimization in LTE systems.

1.1 Background and Motivation

The LTE/LTE-A [5, 9] system is the latest deployed stage of the telecommunications systems advancement series. The system is considered an evolution of its predecessor, termed the universal mobile telecommunication system (UMTS). The LTE standards are developed by the currently dominating standards development group for mobile radio standards, known as the third generation partnership project (3GPP) [10]. During the past two decades, the 3GPP has also produced the second generation (2G) of mobile systems (*i.e.*, global systems for mobile communications (GSM), General Packet Radio Services (GPRS) and Enhanced Data rates for GSM Evolution (EDGE)) that was employing the time and frequency division multiple access (TDMA/FDMA) schemes. The third generation (3G) UMTS, which belongs to the code division multiple access (CDMA) family, was then launched during the early years of the 21st century to evolve the mobile data communications towards higher spectral efficiency as compared to the 2G. Finally, the current 4G LTE was commercialized in 2010 and afterwards. In contrast to previous generations, the LTE adopts the latest access technology, known as the orthogonal frequency division multiplexing (OFDM).

The LTE technology is distinct for its superior data transfer speeds, which theoretically can reach up to 75Mb/s in the uplink and 300Mb/s in the downlink [11]. The total bandwidth is divided into several hundreds of sub-carriers, each of which has a bandwidth of 15kHz and carries 14 OFDM symbols in a 1 msec subframe duration. In a practical LTE system, each 12 sub-carriers (*i.e.*, contiguous in frequency) are grouped together in the smallest resource unit called *resource block* (RB). The available RBs are then scheduled among multiple users in time and frequency across the LTE frames (*i.e.*, typically 10msec of duration) to carry their traffic data. This technique is termed orthogonal frequency division multiple access (OFDMA). The key idea behind OFDMA is that the RB allocation (*i.e.*, typically done by the base station) addresses the mutli-path fading problem of wireless channels in time and frequency. In particular, the base station allocates for each mobile user the best possible RBs in time and frequency, those for which the mobile device is experiencing the strongest signal (*i.e.*, lowest fading), in order to satisfy its traffic demands (*e.g.*, rate and delay). The problem is commonly known in literature as wireless resource allocation or *radio resource management*.

The wireless resource allocation optimization problem in LTE has been widely studied in literature [12, 13]. As highlighted above, the decision of allocating a certain RB to a certain user requires sharing knowledge between the PHY and MAC layers. The PHY layer information comprises of the received signal strength, which randomly fluctuates due to the wireless channel induced noise and fading. According to Shanon capactiy theorem [14], these fluctations limit the wireless channel capacity for accomodating high transmition data rates subject to target bit error rate (BER) performance. On the other hand, the MAC layer involves the MAC buffers' state information related to the upper layer data packets arrivals. The importance of the MAC layer information lies in provisioning the buffers length, which might unfavorably grow during periods of network congestion or deep channel fading leading to queuing delays and overflow losses. Thus, cross-layer resource allocation schemes jointly utilize both; the PHY layer's channel state information (CSI) and the MAC layer's buffer state information (BSI), to optimally manage the wireless network resources (*i.e.*, power and bandwidth) [15] subject to target quality of service (QoS) requirments.

The major challenging factor, which limits the cross-layer optimization in OFDMA networks, is the prediction of the signal behavior and error propagation in wireless media. The subject of wireless propagation prediction (summarized in Chapter 2) has been studied over the years by the research community. However, there is still a need for agile and accurate solutions with real-time response capabilities, which enables solid cross-layer resource allocation schemes in stochastically (and rapidly) varying characteristics of wireless environments (*e.g.*, vehicle-to-vehicle (V2V) communications, intelligent traffic and transportation systems). Such solutions would provide the network's base stations with reliable information about the long-term CSI experienced by mobile users, and hence, would exploit robust cross-layer predictive scheduling capabilities for the available resources to improve the network's spectral and energy efficiencies while meeting the user's QoS needs. Consequently, most of the reported works in the literature mitigate the lack of accurate CSI by scheduling resources in short time horizons, during which the CSI is assumed to be invariant. Meanwhile, in practical LTE system, the CSI is provided by the channel quality indicator (CQI) reporting mechanisms [10]. In this thesis, we propose a novel and agile solution for enabling accurate predictive cross-layer scheduling methodology for LTE networks, and study its impact on the network performance.

1.2 State-of-the-Art and Thesis Outline

Predictive resource allocation techniques have been recently perceived as a promising solution for efficiently designing wireless networks [16]. A predictive resource allocation scheme is capable of utilizing the channel's future capacity in making optimal allocation decisions (*e.g.*, allocation of power, rate, and frequency resources in both time and frequency domains). This is in contrast to traditional schedulers which exploit a short-term knowledge of the wireless channel propagation statistics as commonly provided by various channel reporting mechanisms [11]. Recent advances in localization techniques [17] have enabled location-based services, and facilitated the detection of mobility patterns to track mobile users' daily activities. However, the major challenge facing the practicality of the predictive scheduling techniques remains the absence of a solid predictive scheduling paradigm with accurate and real-time channel prediction capabilities. In this dissertation, we successfully tackle that challenge by proposing a predictive cross-layer resource allocation solution based on the state-of-the-art ray tracing (RT) propagation prediction model [18]. Although studied in the context of LTE, the proposed RT-based model represents an effective and practical solution, which fits the computing capabilities of the future cloud-based cellular architecture. This concept will be elaborated in the remaining chapters of this thesis. Thanks to today's high performance computing platforms (HPC) (*e.g.*, field programmable gate arrays (FPGAs), and graphical processing units (GPUs)) that make our proposed predictive solution realizable. Hence, and in order to reach the thesis objectives, we start in Chapter 2 by completing a major milestone towards the practical implementation of the proposed RT-based solution by providing a thorough survey and software implementation for the RT propagation prediction engine. The subsequent chapters then put emphasis on studying the impact of utilizing the RT engine's knowledge about the channel on the efficient management for wireless radio resources.

The rest of this thesis is divided into six chapters. Chapter 2 presents a literature review on the RT propagation prediction techniques. It also sheds light on a promising vision for utilizing the RT model in optimizing the management of the wireless network resources within various frameworks (*i.e.*, presented in the following chapters). In addition, a ray tracer MATLAB library is established, and validated using the Wireless Insite commercial propagation prediction tool [19], as part of our effort towards implementing a high speed RT engine. Chapter 3 presents the first attempt of utilizing the RT model in a throughput efficient channel-aware scheduler operating under a particular fairness criteria. In Chapter 4, a rather comprehensive investigation is introduced for the application of the RT model in the design of QoS-aware energy-efficient predictive scheduler in the LTE networks. Due to its vital role in determining the QoS levels for today's real-time applications, Chapter 5 delivers a rigorous study for delay jitter modelling and optimization in today's wireless networks. After presenting these jitter models, the study manifested a novel trade-off between optimizing the user equipment's (UE's) energy efficiency (EE) and delay jitter performance for VoIP traffic services over LTE networks (*i.e.*, recently portrayed as VoLTE). Furthermore, Chapter 6 extends the the vision of the EE and delay jitter trade-off of VoLTE, presented in Chapter 5, applying it in a heterogeneous traffic LTE network where various QoS requirements are needed to be concurrently addressed. Finally, Chapter 7 concludes the thesis and proposes future directions for extending the conducted research.

1.3 Contributions of the Thesis

The major contributions of the thesis are summarized as follows.

1.3.1 Contributions of Chapter 2

1. After conducting a thorough literature review on the RT approaches and the typical implementation architecture, a newly proposed correction method for the existing geodesic ray launching technique is presented. The proposed method shows better uniformity in the generated rays' angular separation compared to the state-of-the-art geodesic ray launching technique which directly impacts the prediction results.
2. A ray tracer MATLAB library is developed and cross-validated with the commercial Wireless Insite propagation prediction software. The ray tracer provides a seamless

user-interface that can easily read the Wireless Insite's maps (*i.e.*, in '.city' file format), and the environment's material and propagation characteristics.

1.3.2 Contributions of Chapter 3

1. A link adaptation analysis for studying the effect of the transmission rate adaptation time horizon on the channel outage probability is developed.
2. Based on the channel outage analysis, an optimal formulation for the maximum throughput (MT)-scheduling problem constrained by Max/Min fairness is provided. The proposed resource allocation scheme is assisted by long-term ray tracing channel predictions for maximizing the overall cell's throughput in TDMA uplink.
3. A heuristic algorithm is proposed to reduce the optimal scheme's computational requirements. In terms of complexity, the algorithm is of $O(kNlog(N)) + O(k^2N)$, where k is the number predicted channel frames and N is the number of users. The performance of the heuristic scheduler is evaluated and compared to the optimal scheduler, and to another existing scheduler.

1.3.3 Contributions of Chapter 4

1. An optimal framework which minimizes the energy consumption of the UE's receiver circuit while satisfying the effective bandwidth constraint is presented. The framework utilizes the ray tracing channel prediction model, and it considers both the modulation and coding scheme (MCS) and UE circuit operation time.
2. To assure feasible solutions, a second formulation for the optimization problem is proposed. This is done by relaxing the rate constraint using the penalty method to cope with the channel capacity limitations.
3. After investigating the dominant factors which affect the UE's power consumption budget in the downlink, a further modification for the optimization problem is conducted. That is allowing the scheduler to focus solely on optimizing the number of wake-up TTIs during which the UE's receiver circuit is decoding the data packets.
4. To address the complexity of the optimization problem, a low complexity heuristic

algorithm is designed to solve the scheduling problem with a comparable performance.

1.3.4 Contributions of Chapter 5

1. To take a forward step towards facilitating the design of powerful jitter control schemes, a rigorous mathematical model of the packet delay jitter in a simple queuing system with one traffic buffer of infinite length, one server and a single hop is developed.
2. A multi-objective optimization problem is formulated for the UE's EE and the delay jitter subject to delay constraints for VoLTE services.
3. Due to the inherent complexity in the optimal formulation, we propose two different heuristic algorithms for solving the resource allocation problem.
4. The obtained results showed a novel trade-off between the UE's EE and the packet delay jitter.

1.3.5 Contributions of Chapter 6

1. An optimal packet scheduling framework for optimizing the UE's EE and the delay jitter performance for real-time traffic connections in the downlink of heterogeneous traffic LTE networks is derived. The resource allocation problem is formulated as a binary integer programming (BIP) problem.
2. The proposed framework utilizes the widespread utility-based prioritization scheme to simultaneously deal with three different QoS classes denoted as best-effort, rate-constrained and delay-constrained classes.
3. A two stage paradigm for improving the packet delay jitter is proposed. In the first stage, a newly proposed metric function which captures the jitter requirement (side-by-side to the delay requirement) for real-time applications is used. In the second stage, two jitter-efficient resource allocation mechanisms are proposed for minimizing the packet delay jitter.
4. Based on the scheduling granularity, two different heuristic versions of our packet scheduler are proposed. The first version tackles the EE and delay jitter objectives

within the commonly employed time granularity of a single LTE frame. The second version provisions further potential improvement in the EE and delay jitter performances by utilizing our previously proposed cloud-based predictive scheduling model.

5. To enable the predictive version of our proposed scheduler, a window-based mechanism is proposed to alleviate the short-term BSI/long-term CSI imbalance problem.
6. To address the complexity of the optimal scheduler, and to be able to solve the problem, a total of four different heuristic algorithms are proposed based on the designed jitter control mechanisms for each version of our proposed scheduler.

Chapter 2

Ray Tracing Propagation Prediction: From Theory to Implementation

2.1 Introduction

The great demand for high QoS in today's broadband applications requires accurately designed wireless communication systems. To be able to implement efficient mobile network infrastructures, an accurate method of predicting the propagation of radio signals is always preferred over expensive and time consuming field measurement campaigns. Therefore, various radio frequency (RF) propagation prediction models, which fit both indoor and outdoor environments [20, 21, 22, 23], have been developed since the beginning of 1970's [24].

In the literature [25], propagation models are classified into three categories: empirical (*i.e.*, statistical in nature) models, site-specific (*i.e.*, deterministic in nature) models and semi-empirical models. These models are usually used to predict radio channel parameters including large-scale (*i.e.*, path loss) and small scale (*i.e.*, fading) effects, which are known to be a function of the propagation path distance and the mutlipath propagation, respectively. However, the approach for each category is different. Empirical models (*e.g.*, Okumura [26], Hata [27] and Walfisch-Ikegami [28]) use a set of equations extracted from the measured average of losses along typical radio links. They proved to be accurate as long as the environment characteristics are invariant [29]. Site-specific models predict channel parameters numerically based on either the finite-difference time-domain (FDTD) method [30] or the physical optics (PO) theory [31], both of which use physical laws of wave propagation. These models are able to predict the path loss and other channel parameters more accurately and reliably than empirical models. However, their drawback is the significant computational burden they experience due to their dependence on the detailed and accurate description of all objects in the propagation environment such as buildings, roofs, walls and material characteristics. On the other hand, semi-empirical models are considered to be in the mid-way between their empirical and site-specific counterparts. For example, the algorithm reported in [32] performs numerical calculations for the diffraction effect of the building structures present in the environment based on certain ideal assumptions (*i.e.*, uniform buildings heights and spacing). The results are more accurate than empirical models, however, the accuracy is far behind the site-specific models. The same is true for the complexity of the algorithm. As a result, these models were commonly perceived as an efficient alternative for the traditional field measurements.

The ray tracing (RT) propagation model is a deterministic approach which offers accurate modelling for predicting propagation effects in wireless communication channels based on the information provided by a geographic information systems (GIS) database. This database contains an accurate geometrical and morphological characterization of the objects existing in the propagation environment [18, 33]. The basic mechanism of any RT algorithm is to search for all possible radio paths connecting the transmitter and receiver locations. The searching process must account for all combinations of propagation effects, such as direct line-of-sight (LOS) as well as reflections and diffractions arising from the surrounding geographical environment objects. In the literature [33], two methods known as Shooting and Bouncing Rays (SBR) and Image Method are employed to determine the ray trajectory between the transmitter and the receiver. More details about the two methods will be provided below. Regardless of which method is employed in tracing the rays, a vector summation for the emitted field components associated with the received rays is then calculated to evaluate the total received power [34] (as will be shown later in Section 2.3.3).

It can be inferred from the previous paragraph that in complicated dense environments with many scattering objects, the ray tracing process becomes computationally intensive and time consuming since it requires a huge number of ray intersection tests with the surrounding obstacles. During the last decade [35, 36], and even recently [37, 38], many efforts were made to accelerate the ray tracing process, mainly by simplifying the geometry of the ray tracing environment. Use of these techniques leads to a smaller number of intersection tests and fast elimination of the redundant rays (*i.e.*, rays which miss the receiver location).

The rest of this chapter is organized as follows: Section 2.2 presents some of the potential applications in which the ray tracing model is of special interest, besides the motivation and the recent trends of the technique. The classification of algorithms employed in the ray tracing model is provided in Section 2.3. In Section 2.4, some of the reported ray tracing acceleration techniques, used to improve the computational efficiency (in terms of processing time and power consumption) of the ray tracing algorithm, will be discussed. The general architecture of a ray tracing engine is then provided in Section 2.5. Section 2.6 presents MATLAB implementation for the shooting and bouncing rays (SBR) ray tracing algorithm in conjunction with one of the acceleration techniques discussed in Section 2.4. The MATLAB ray tracer developed in Section 2.6 is then cross verified in Section 2.7 with the commercial Wireless Insite propagation prediction tool while considering different propagation scenarios. Finally, Section 2.8 concludes the chapter.

2.2 Applications, Motivation and Recent Trends

It has become evident that today's wireless communication technologies have created a broad spectrum of applications. Some of these applications are cellular networks, satellite networks, radar and marine communications, vehicular communications [39, 40] (*e.g.*, vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) and vehicular ad hoc networks (VANETs)), wirless local area networks (WLANs), personal access networks (PANs), wireless sensor networks (WSNs), and internet of things (IoT) [41]. Having these heterogeneous wireless networks in addition to the stochastic nature (in time and frequency) of the radio channel characteristics, makes the development of fast and efficient propagation prediction platforms, for use in the design of optimized wireless networks (in terms of capacity, data rate and power), an urgent matter. Ray tracing approaches have provided a promising solution with higher accuracy compared to statistical models. Ray tracing is not limited only to wireless systems. It inspired potential applications in the field of medical device technology. It is capable of simulating electromagnetic radiation propagation in biological tissue and light scattering materials which is used in the design of bio-medical devices [42]. Medical imaging and real time visualization of tissue cells and organs [43] also require high performance computing and rendering capabilities which could be provided by ray tracing based techniques.

Rapid and accurate estimation of radio propagation allows utilization of new optimization paradigms that would not be possible otherwise. For example, dynamic frequency planning allows the operator to dynamically assign more bandwidth to cells with more traffic. Although this is partially done in modern networks using some heuristics [44], full implementation of the optimum planning requires massive computations which is not practical with traditional tools and processing power margins.

Highly dynamic applications such as V2V and V2I communications can also benefit from fast (near real-time) channel estimation. In such systems, the direct result is saving the bandwidth, power and time required for excessively transmitting training signals that assist in the sensing of channel's quality. In urban environments, the channel between two moving vehicles or a vehicle and a fixed antenna can dramatically change in a very short time. Predicting these sudden changes is an effective solution for enabling reliable vehicular communications in such environments. The prominence of an efficient urban vehicular communication is coming from the growing need of today's wireless market for such type of communications, especially in modern cities [45].

A practical and efficient method to run the ray tracing algorithm fast enough for the mentioned applications is to utilize pipeline and parallel architectures of modern FPGA devices [46]. This chapter provides an insight on how to implement a complete ray tracing solution using MATLAB. Thus, this chapter serves as a guide to build a solid understanding of the implementation problem on the system level to further enable the practical implementation of the hardware solution. It is also worth mentioning that, to the best of our knowledge, no reported work in the open literature has considered the problem of accelerating the ray tracing engine (in the field of radio propagation prediction) in the hardware implementation domain. Hence, it is an attractive topic for future investigation.

2.3 Ray Tracing Algorithms

Ray tracing is a deterministic approach which offers an accurate modelling for predicting propagation effects in wireless communication channels based on the provided informa-

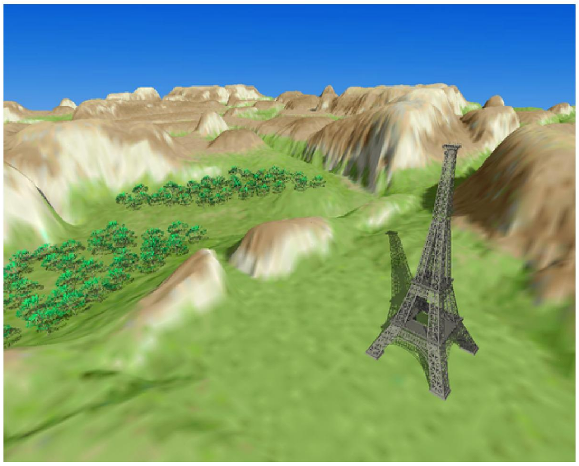


Figure 2.1: Three-dimensional scene rendered using real-time FPGA ray tracer [2]

tion in a Geographic Information Systems (GIS) database. It is an extension of its foremost illumination model used primarily in rendering high quality 3D graphics [18]. When appeared, the ray tracing model was used to simulate the propagation of light. An example of a generated graphic using the ray tracing model [2] is shown in Fig. 2.1.

Since radio signals and visible light are both electromagnetic (EM) signals, the propagation of radio signals can also be modeled using the same method. However, because of the vast difference in the operating frequency of each of these EM signals, some assumptions should be made in each case. For example the diffraction, which plays an insignificant role in a typical scenario for rendering graphics, accounts for a significant portion of the radio signal propagating in urban areas [47]. Consequently, important modifications on the mathematical formulation have been applied to the ray tracing model to consider the signal phase and polarization of each individual ray [33]. The objective of this RF-based model is to provide the wireless network designer with an accurate estimation of the signal propagation statistics such as: received signal strength (*i.e.*, important for measuring network geographical coverage), power delay profile, delay spread, and angles of arrival and departure. These parameters are widely known to enable efficient designs of wireless communication systems. Ray tracing algorithms are broadly classified into two categories: Shooting and Bouncing Rays (SBR) and Image Method [33]. More details about these two methods are provided in the following subsections.

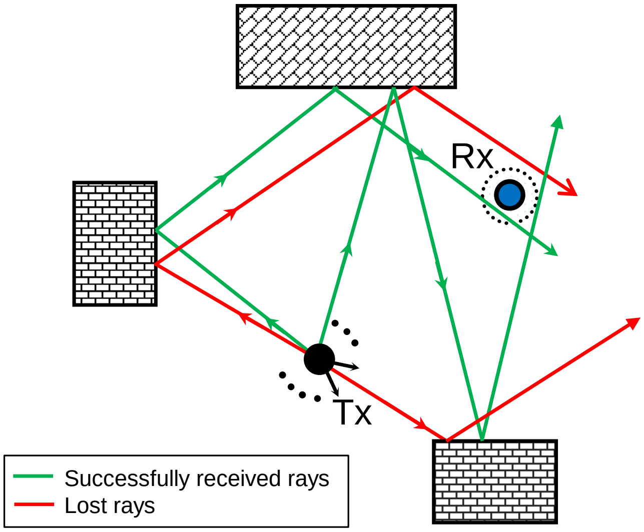


Figure 2.2: SBR method

2.3.1 Shooting and Bouncing Rays Method

The basic mechanism of SBR is that rays, which represent the radio signal wave-front, are launched from the location point of the transmitting antenna covering all directions in the 3D environment. The traversed path by every launched ray is then traced in the propagation environment geometry using the optical laws of reflection and geometrical theory of diffraction (GTD) [48]. Every object hit by a ray interacts with it and changes some of its characteristics (*e.g.*, polarization and strength) until the ray either reaches the receiver's target location or gets lost beyond the reception area. This is further illustrated in Fig. 2.2. A vector sum for the emitted field components associated with the successfully received rays is then calculated to determine the total received signal power. The propagation effects that are modeled in this process are the reflection, diffraction and transmission of the radio signal through the environment's physical obstacles (*e.g.*, buildings, cars, trees, ground surface, lamp posts, etc.). A ray might experience different combinations of these effects along its path from the transmitter to the receiver.

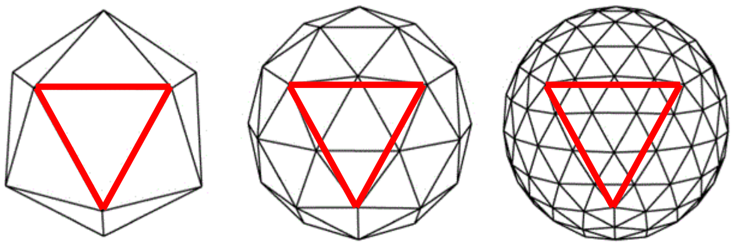
Generally speaking, the SBR ray tracing method is an iterative process based on testing all of the launched rays (in the order of several hundreds of thousands [33]) in the propagation environment for possible reception by the receiver antenna location. For each ray, the testing is done by applying a series of ray/object intersection tests [18] with all of the objects (*i.e.*, modeled by their facets) located in the environment. Due to the very large number of degrees of freedom, the process for testing all of the bounced rays is computationally expensive, especially for highly dense propagation scenarios. That complexity is known to be the main driver of a main stream research [35, 36, 37, 38, 49] devoted for developing efficient ray tracing acceleration techniques.

2.3.1.1 SBR launching method

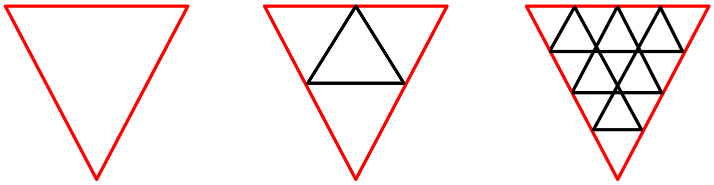
Two major criteria should be satisfied by any launching technique. Those criteria are large-scale uniformity and small-scale uniformity [50]. The first characteristic ensures that all directions of space are illuminated equally, which requires uniform distribution of the launching points along the surface of a sphere. The second characteristic defines the importance of having equal angles between all neighboring rays in order to achieve uniform distribution of rays along the wave-front. The only computer generated geometry found to satisfy the two aforementioned characteristics is the geodesic sphere [50]. A geodesic sphere can be approximated by subdividing (or tessellating) the faces of an icosahedron (*i.e.*, inscribed inside a unite sphere) and extrapolating the intersection points to the surface of a sphere. It should be noted that the tessellation of an icosahedron face has to be in a way such that no overlaps or gaps occur. Fig. 2.3 provides detailed illustration about tessellating a regular icosahesron with a frequency of 4 to produce a geodesic sphere. Also, a MATLAB generated geodesic sphere with a tessellation frequency of 4 is depicted in Fig. 2.4. The vertices of the hexagonal patches (*i.e.*, equal to $10{f_t}^2+2$, where f_t is the tessellation frequency) surrounding the sphere (*i.e.*, bold dots in Fig. 2.4) are then taken as the rays launching points. The approximated geodesic sphere has adequate uniformity. However, due to different sizes of hexagons generated by this technique, the generated rays have some discrepancies in their angular separation. That discrepancy will be discussed later in detail in Section 2.7.

2.3.1.2 SBR receiving model

An important factor that affects the estimation accuracy and computational time for a ray tracer is the reception criteria. Since radiation wave-front is modeled by several rays in a discrete manner, almost surely none of the rays hits the receiver point, but they pass



a) Tessellation of a regular icosahedron to produce a geodesic sphere (i.e., tessellation frequency =4)



b) Tessellation of a regular icosahedron face into 4 equal segments (i.e., tessellation frequency =4)

Figure 2.3: Geodesic sphere with tessellation frequency of 4

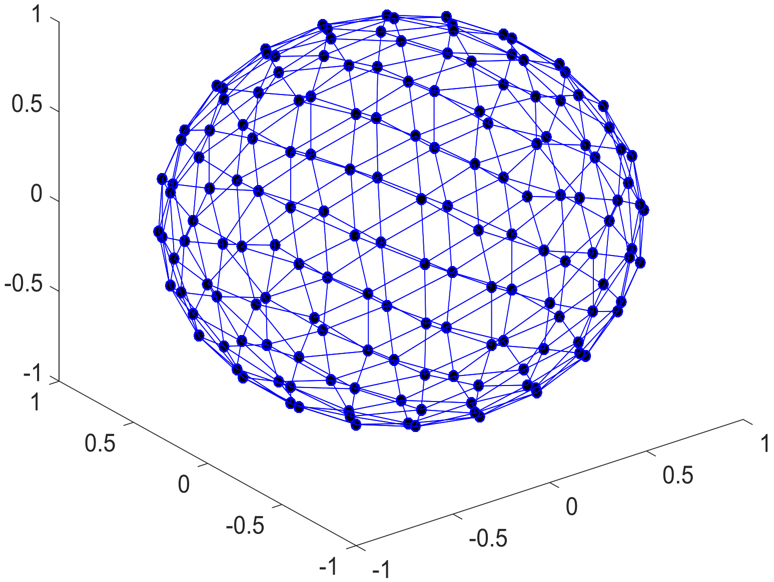


Figure 2.4: A MATLAB generated geodesic sphere with tessellation frequency of 4

very close to it. The problem of identifying which ones should be considered as received rays is not a straight forward problem and creats a potential trade-off between accuracy and complexity. It also relates to the ray launching technique. This problem has been rigorously investigated in [50, 51] where two different techniques have been proposed. The following paragraphs summarize the works reported in [50, 51] and discuss the implementation issues in terms of complexity and accuracy.

The ray launching technique is the first step in implementing an SBR-based ray tracing model. In any ray tracing algorithm for electromagnetic radiation modelling, the ray tracer's main function is to determine all the possible paths between two points; a transmitter and a receiver. In SBR, the transmitter launches a huge number of rays in all possible angles. In general, more rays translates to better accuracy. Moreover, since the rays spread more as they travel further from the transmitter, greater number of rays are required for studying large areas.

The question of receiving rays that pass close to the receiver point is addressed in [50] and [51]. In [51], the receiver is assumed to be a sphere centered at the receiver point. To account for the divergence of rays as they travel away from the transmitter, the radius of the receiving sphere is a function of the ray's length and the angular separation between adjacent rays. The ray is considered received if it intersects the sphere at any point. Although the ray/sphere intersection test [18] is known to be simple and fast, the major drawback of this technique is that received rays might have double counting errors, occuring when the same wave-front is counted twice, resulting in +3dB error as reported in [50]. Another disadvantage of this technique is the phase error, which is due to the fact that rays hitting different points of the sphere have different length and consequently different phases. If not accounted for, this phase error can cause severe power error when several rays are added together.

Another reception technique, proposed in [50], mitigates the reception errors that are inherent in the reception sphere model proposed in [51]. This technique, namely distributed wavefronts, assigns a weighting factor to each of the rays in the vicinity of the receiver. The approach utilizes the approximate uniformity of the geodesic rays to define a symmetric weighing function. In this technique, the contribution of each nearby ray to the total received field, is specified based on its radial distance from the receiver point along the

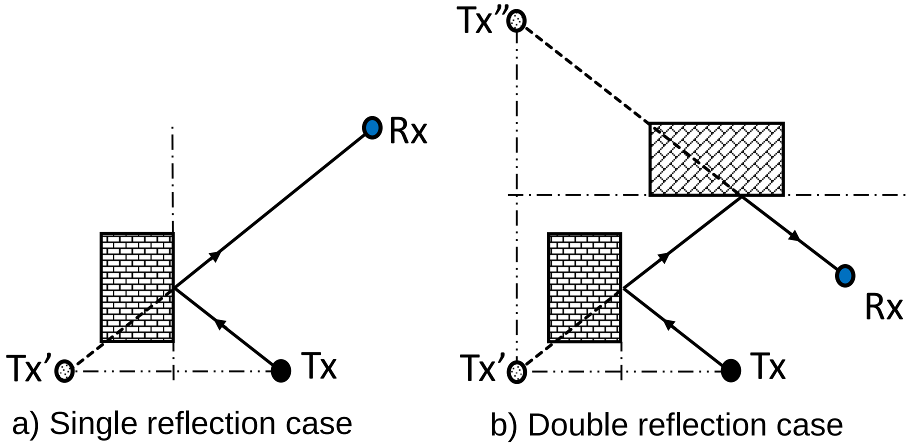


Figure 2.5: Ray reflections in image method

potential wave-front. As the radial distance becomes large, the ray contribution decreases.

2.3.2 Image Method

The image method finds the path between the transmitter and the receiver based on the image theory [33]. It creates virtual source points, called image sources, which are the mirror image of the original source with respect to the reflecting surface. The reflected rays are then assumed to be radiated directly from these image sources. The intersection between the line connecting the virtual source image and the observation point (*i.e.*, receiver point) with the reflecting surface, usually termed as facet, determines the ray reflection point. This approach is illustrated in Fig. 2.5 for the single and double reflection cases. A single transmitter image (*i.e.*, denoted as Tx') is created in the single reflection case, while two images (*i.e.*, denoted by Tx' and Tx") are created in the two reflections case. The maximum number of reflected rays (equal to the number of images created) that could possibly hit the receiver point is equal to the number of facets in the scenario. Similar to the SBR method, the calculated field at the receiver point is based on the radiation characteristics of the original source and the electrical properties of the reflecting surface. Generally speaking, the receiver point will be aimed by a reflected ray from a facet only if the receiver is located inside the facet's reflection space. The facet reflection space is illustrated in Fig. 2.6.

Despite the simplicity of the method, it can only be used for calculating field components due to reflected rays. However, tracing diffracted rays is more expensive in terms

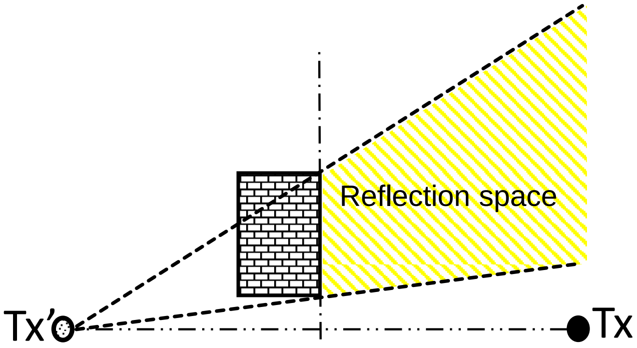


Figure 2.6: Facet reflection space

of computations due to the infinite degrees of freedom for the direction of diffracted rays. In addition, in complicated 3D urban scenarios (*e.g.*, city of Manhattan) that contain very large number of obstacles, the image method will not be an efficient solution in terms of the required CPU time and memory as the number of required virtual sources will be significantly high. Hence, the SBR is the method of choice for modelling complicated 3D scenes while the image method is known to be more efficient in simple 2D cases. From another perspective, the image method is highly accurate as it determines the exact propagation paths to the receiver's location without introducing phase errors. In the case of SBR, the accuracy is dependent on the angular separation of the launched rays. As a result, some implementations use the image method to augment SBR and correct the possible phase error caused by inaccuracies in the reception.

2.3.3 Electric Field Evaluation

In a 3D environment, the evaluation of the final electric field complex vector for each of the successfully received rays is an intricate process irrespective of which ray tracing method is employed. This is due to the fact that the electric field has two polarization components at each reflection boundary. One component is parallel (*i.e.*, vertical) to the plane of incidence and the other is perpendicular (*i.e.*, horizontal) [34]. In the case of a ray with only multiple reflections, the final received electric field complex polarized vector is calculated in an iterative process at each reflection point. To simplify the explanation of this iterative process, let us first consider the case of single reflected ray, and from there the generalization to N-reflected ray will become obvious. The complex polarized vector for the reflected ray electric field is given by:

$$\overrightarrow{E}_r = (\overrightarrow{E}_{\|}^{r} + \overrightarrow{E}_{\perp}^{r})\,\frac{e^{-j\beta d}}{d} \tag{2.1}$$

where $\overrightarrow{E}_{\|}^{r}$ and $\overrightarrow{E}_{\perp}^{r}$ are the parallel and perpendicular polarization vectors of the reflected electric field, respectively, β is the propagation constant, d is the total traversed distance by the ray (from the transmitter to the receiver).

It should be noted that the total traversed distance is given by $d = d_1 + d_2$, where d_1 is the distance traveled by the ray from the transmitter to the reflection point, while d_2 is the distance from the reflection point to the receiver. The importance of highlighting d_1 and d_2 will be seen shortly, when generalizing the field calculation process for an arbitrary number of reflections. The field polarization vectors in (2.1) is given by:

$$\overrightarrow{E}_{\|}^{r} = E_{\|}^{r}\hat{e}_{\|}^{r}\,,\ \overrightarrow{E}_{\perp}^{r} = E_{\perp}^{r}\hat{e}_{\perp}^{i} \tag{2.2}$$

where $E_{\|}^{r} = R_{\|}E_{\|}^{i}$ and $E_{\perp}^{r} = R_{\perp}E_{\perp}^{i}$ are the magnitudes of the parallel and perpendicular polarization field vectors, respectively, $E_{\|}^{i} = \hat{e}_{\|}^{i}.\overrightarrow{E}_i$, $E_{\perp}^{i} = \hat{e}_{\perp}^{i}.\overrightarrow{E}_i$, $\hat{e}_{\|}^{r} = \frac{k^r \times \hat{e}_{\perp}^{i}}{\left|k^r \times \hat{e}_{\perp}^{i}\right|}$ is the parallel polarized reflected field direction, $\hat{e}_{\perp}^{i} = \frac{k^i \times \hat{n}}{|k^i \times \hat{n}|}$ is the perpendicular polarized incident field direction, $R_{\|}$ and $R_{\perp}$ are the parallel and perpendicular polarized components of the Fresnel dyadic reflection [47], respectively, $\hat{e}_{\|}^{i} = \frac{k^i \times \hat{e}_{\perp}^{i}}{\left|k^i \times \hat{e}_{\perp}^{i}\right|}$ is the parallel polarized incident field direction, $\overrightarrow{E}_i = E_o e^{-j\beta d_1}$ is the incident field vector at the reflection point, E_o is the emitted electric field, k^i and k^r are the incident and reflected field directions, respectively.

As can be seen from (2.1) and (2.2), the complex polarized vector for the final received electric field is calculated in a single iteration since only single reflection point is considered. In general, for N-reflected ray (*i.e.*, N-reflection points), only the calculation of the polarized components shown in (2.2) is executed recursively at each reflection point having the $\overrightarrow{E}_i$ at the reflection point-(n) set to be equal to $\overrightarrow{E}_{\|}^{r} + \overrightarrow{E}_{\perp}^{r}$ that is previously calculated at reflection point-(n-1). The final complex received field vector will be calculated once at the end of the process using the formula in (2.1) after calculating the $\overrightarrow{E}_{\|}^{r} + \overrightarrow{E}_{\perp}^{r}$ at the final Nth reflection point. It is also worth mentioning that the calculation of $R_{\parallel}$ and $R_{\perp}$ in (2.2) depends mainly on the operating frequency and the material characteristics of the reflecting surface as follows [47]:

$$\begin{aligned} R_{\perp}(y) &= \frac{\sin(y) - \sqrt{\varepsilon_c - \cos^2(y)}}{\sin(y) + \sqrt{\varepsilon_c - \cos^2(y)}}, \\ R_{\parallel}(y) &= \frac{\varepsilon_c \sin(y) - \sqrt{\varepsilon_c - \cos^2(y)}}{\varepsilon_c \sin(y) + \sqrt{\varepsilon_c - \cos^2(y)}} \end{aligned} \tag{2.3}$$

where y is the angle of incidence, $\varepsilon_c = \varepsilon_r - j60\sigma\lambda$ is the relative dielectric constant, ε_r is the permittivity, σ is the special conductivity of the reflecting surface, and λ is the wavelength. Further details about the calculating the electric field vector in case of diffracted rays could be found in [33].

2.4 Ray Tracing Acceleration Techniques

A major ray tracing problem which has drawn the attention of many researchers is the high computational complexity, and hence, execution latency of the algorithm. These limitations become notable in complex scenarios composed of large capacity geographical databases. Therefore, the ray tracing algorithm should be improved for faster and more efficient processing. Most of the common acceleration techniques can be found in [25, 35, 36].

An acceleration technique can generally improve the ray tracing efficiency by utilizing one or more of the following strategies [18, 33]:

- *Using simple and faster ray/object intersection test*: simpler and faster tests implies decomposing a single primitive object into many simpler objects. The facet decomposition model, which uses polygonal plane facet as its primitive object, is known as the model with the simplest ray intersection test. Thus, most of the implementations adopt a variant of it.
- *Reducing the number of ray/object (or ray/facet) intersection tests*: a wide range of existing acceleration algorithms focus on reducing the number of ray/facet intersection tests (*i.e.*, known as shadowing tests) that need to be performed. This is due to the drastic increase in the number of tests required for complex environments.

Instead of testing the intersection of a ray with all the facets in the environment in a brute-force manner, the acceleration algorithm simplifies (*i.e.*, partitions) the environment to find out the potential intersecting facets and executes the test only on them. Different ways for partitioning the environment will be discussed in the following acceleration algorithms.

- *Reducing number of rays in the environment*: rays with a little contribution in the final received field can be safely discarded in order to minimize calculations and increase the speed. This is typically done by setting a minimum threshold value for the ray's field strength (or power). Consequently, if at any step of the tracing process the value of any ray's field strength falls below the defined threshold, the algorithm can safely discard (*i.e.*, stop tracing it) that ray.
- *Using generalized entities other than rays*: the main idea behind generalizing rays is to trace many rays simultaneously by replacing pencil rays, for instance, with cones having circular or polygonal cross sections. However, this methodology has not been practically implemented for electromagnetic modelling although being used extensively in computer graphics applications.

In the rest of this section, three famous acceleration techniques which belong to the second strategy of those discussed above.

2.4.1 Binary Space Partitioning

In this method [52], the space is partitioned into several sub-spaces, each of which has one facet inside. The sub-spaces are sorted into a tree graph called BSP tree. The tree contains the relative position (visibility) of each facet with the rest. At each step of building BSP tree, the space is divided into two half-spaces (branches), one contains facets in front of the root facet and the other contains the facets behind. The facets in the two half-spaces are designated with respect to the facet's surface normal vector. The root (starting) facet of the tree could be any facet in the environment. The subsequent tree branches then undergo the same partitioning procedure till each branch of the tree contains only one facet.

The BSP tree is referred to see if there is possible interaction between receiver and the transmitter. Hence, the ray is only tested with the facet (or the portion of it) that is collocated in the ray area. More details about the algorithm and the generation of the BSP tree could be found in [33].

2.4.2 Space Volumetric Partitioning

This algorithm works by dividing the space into equal size cells called voxels [53]. Voxels are cubes with sides parallel to the study area's (*i.e.*, geographical environment) coordinate system axis. Each facet in the study area will be registered inside a voxel whether it lies totally or partially within its boundaries. The registration information will be stored in a matrix which will be queried at each iteration of the tracing process. Similar to the BSP method, the SVP registration matrix generation does not depend on the transmitter and receiver locations.

To trace each ray, the algorithm first determines the voxels of interest in which the ray intersects. Those voxels are then sorted according to their distances from the ray origin, starting from the voxel that contains the source. It should be noted that ray origin is the transmitter location in case of direct ray or the last reflection point in case of reflected ray. In the next step, only the facets registered within the sorted voxels are tested for intersection with the ray in a sequential manner. The test stops with the first detected intersection. It is worth noting that the distance-based arrangement guarantees that decided intercepting facet, at the end of the testing process, is practically the right one (*i.e.*, closer to the ray origin) compared to those which might theoretically intercept the facet as well (*i.e.*, facets existing within subsequent distant voxels). As a result, the number of shadowing tests required will be dramatically reduced.

The voxel size is known to be the major parameter affecting the efficiency of the SVP algorithm. Large voxel size results in minimal elimination of the facets in the environment. Conversely, setting the voxel size to small value leads to creating large number of voxels which requires more processing and storage requirements. As a consequence, the author in [33] suggests to set the voxel size close to the size of the facets.

2.4.3 Angular Z-Buffer

The Angular Z-buffer (AZB) [35] algorithm is very close to the SVP algorithm. Like SVP, the AZB algorithm divides the space into equally sized angular partitions in the spherical coordinate system. In other words, the 3D space is divided into equal angular sections named anxels. All anxels are defined using the spherical coordinate system that assigns horizontal and vertical angles for each. The origin of the coordinate system will be taken at the ray origin. Therefore, the partitioning process depends on the transmitter location unlike previous techniques. The algorithm works similar to the SVP method. Each facet in the environment will be located totally or partially inside an anxel. The facets within the anxel containing the traced ray, are then arranged according to their distance from the ray origin. This information will be stored in the AZB matrix. Additional elimination for distant facets within the anxel can be achieved by removing those which are completely shadowed by others closer to the ray origin (*i.e.*, painter's algorithm [54]).

For a given observation point, the ray path will be tested sequentially for shadowing only with the sorted facets located in the anxel comprising the ray. Hence, the number shadowing tests conducted will be far less than the total number of facets existed in the environment.

It is very important to note that the efficiency of the AZB algorithm depends on the number of anxels used in partitioning with respect to the size of the environment. It means that for bigger scenarios, the number of anxels should be designed carefully to keep the number of facets in each anxel from growing remarkably. This is because anxels diverge with distance, which in turn increases the anxel volume, and thus, large number of facets will be registered in a single anxel. Consequently, the efficiency of the algorithm decreases. It should be noted that this problem does not exist in small scenarios (*e.g.*, micro and pico cells) in which the attained efficiency is usually high.

2.5 Ray Tracing Engine Architecture

The typical block diagram of a ray tracing system is depicted in Fig. 2.7. It is composed of three layers. The first is designated for initializing and arranging the engine input data

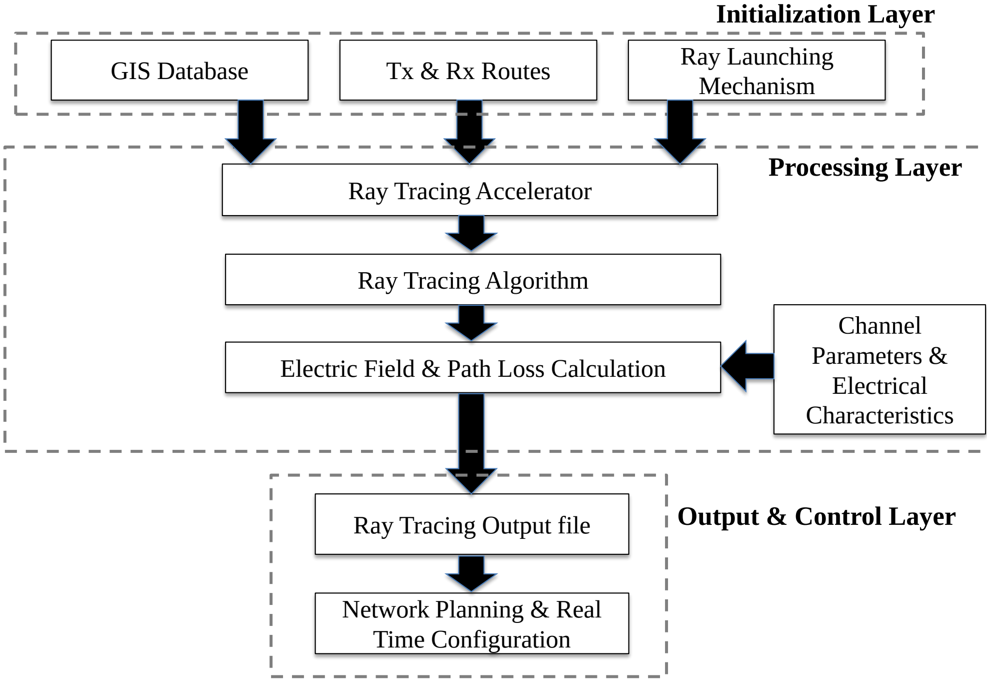


Figure 2.7: Ray tracing system architecture

which includes: the environment's geometrical and morphological data that is provided by a geographic information system (GIS) database, information of the generated rays from the launching mechanism, and transmitter and receiver locations.

The second layer starts by pre-processing the environment based on the data supplied by the preceding layer for accelerating the speed of the subsequent ray tracing algorithm. The pre-processed environment then becomes simplified and ready for tracing the launched rays and finding reflections, diffractions and transmissions with the surrounding obstacles using the selected ray tracing method (*i.e.*, either SBR or image). When implementing the ray tracing system in MATLAB, as will be discussed in the next section, the SBR method, geodesic ray launching and the distributed wave-fronts method [50] are employed as the ray tracing method, ray launching mechanism and receiver model, respectively. For each received ray detected by the ray tracing algorithm (*i.e.*, SBR), a field evaluation procedure (as discussed in Section 2.3.3) is executed for calculating the corresponding final complex field vector. All vectors corresponding to received rays are added together to determine the total received field intensity, and hence, the path loss at the receiver's location. The same process is subsequently repeated along the receiver route. All obstacles' (*e.g.*, building walls) electromagnetic properties such as relative permittivity and conductivity are predetermined before calculating the ray's field components. Among the resulting information is the path loss which defines radio coverage, a vital information for planning a cellular network.

The third layer will be responsible for storing the processing output data for further analysis and displaying purposes. The output data is stored in a file which contains information related to successfully received rays such as: ray coordinates, number of reflections, assigned weighting factor, reflection angle and point at each boundary, total propagation distance, field vector, path loss, and field intensity. This data can be used to calculate other important channel parameters, such as delay spread, power delay profile, direction of arrival or departure. Optimal design for the ray tracing engine, in terms of speed and accuracy, would enable efficient planning of wireless networks and configuring the network in real time to dynamically match the stochastic nature of the channel and traffic data streams.

2.6 MATLAB Ray Tracer

As the first step towards implementing a high speed ray tracing engine, a fast prototyping platform for the whole system is developed and tested in MATLAB. The main purpose is to establish solid understanding of the ray tracing problem mechanics and to provide a reference simulator (especially for MATLAB users) that could be used for verifying and facilitating future hardware designs. Thus, the technical hardware challenges of implementating the ray tracing engine in a cellular network is out of this chapter's scope. This section introduces the detailed structure of the MATLAB system model. In the next section, the MATLAB ray tracer will be presented and cross-verified with a commercial propagation prediction software. To avoid confusing the reader, it is imperative to point out that the MATLAB model presented in the following discussions represents a direct MATLAB realization for the components of the typical ray tracing system (*i.e.*, illustrated in Fig. 2.7) known in the literature [47].

For the sake of designing a scalable MATLAB model that could be seamlessly adjusted and configured based upon the target simulation scenario requirements, a modular MATLAB architecture, which consists of several interconnected subroutines, is established. The developed architecture and its operation can be simply described with the aid of the flowchart illustrated in Fig. 2.8. It starts by initializing the ray tracing input parameters such as: number of launched rays required for the ray tracing algorithm (*i.e.*, N), number of receiver locations along its route (*i.e.*, M), transmitter and receiver location coordinates in the Cartesian 3D space, the study area boundary limits of the propagation environment in the x-y plane and the vertices' coordinates for each facet within the study area boundaries (*i.e.*, typically 4 vertices per every facet). To reduce the time and effort spent in the initialization stage, especially in arranging the detailed and complex geometry of the propagation scenarios, a MATLAB interface (*i.e.*, script file) is built to seamlessly and directly read the geometrical information of all of the environment's obstacles from the GIS database. In our case, the MATLAB interface is designed to read the '.city' file which uses the facet model [33] to describe the environment's geometry. The '.city' file is supported by the Wireless Insite propagation prediction tool (*i.e.*, used later for verifying our MATLAB ray tracer) offered by Remcom Inc. [19]. The file can be easily created as the Wireless Insite tool encompasses a handy GUI-based editor. This is in addition to the Insite's built-in converters which read popular 3D computer-aided design (CAD) formats (*e.g.*, AutoCad's '.DXF' files, and environmental systems research institute (ESRI) '.shape' files) and convert them to the '.city' format. To further illustrate the MATLAB interface function, Fig. 2.9 shows a MATLAB plot for the directly extracted information from a '.city' file created on the Wireless Insite software for 3D urban scenario. As a result, the interface enables the migration from the Wireless Insite domain to the MATLAB domain in one step. It is also worth noting that the MATLAB architecture in Fig. 2.8 is designed for 3D ray tracing, and hence, all computations are carried out using the Cartesian coordinate system.

Following the initialization of the ray tracing parameters, the SVP partitioning algorithm takes place to have each facet fully or partially registered in a voxel, as explained in Section 2.4.2. The partitioning algorithm only requires the study area boundaries and the environment's geometry facets from the preceding initialization stage. Meanwhile, if not initially specified by the user, the voxel size is calculated based on the maximum facet size in the environment. Based on the selected value of N, the geodesic rays' generation function generates the tracing rays in such a way that the angular spacing between them is constant.

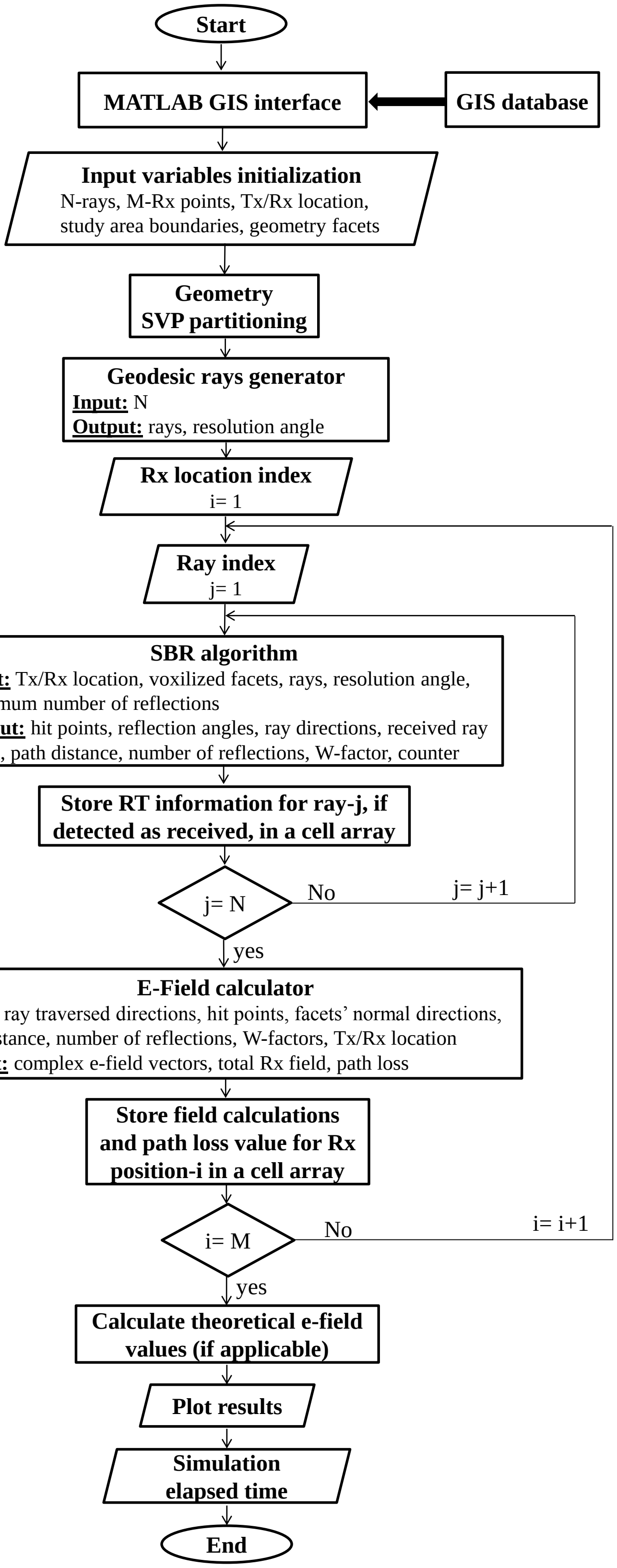
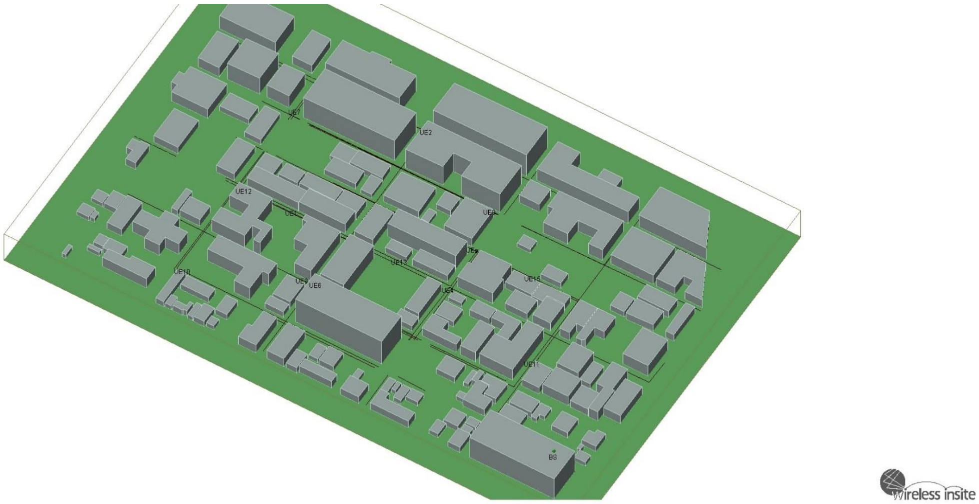
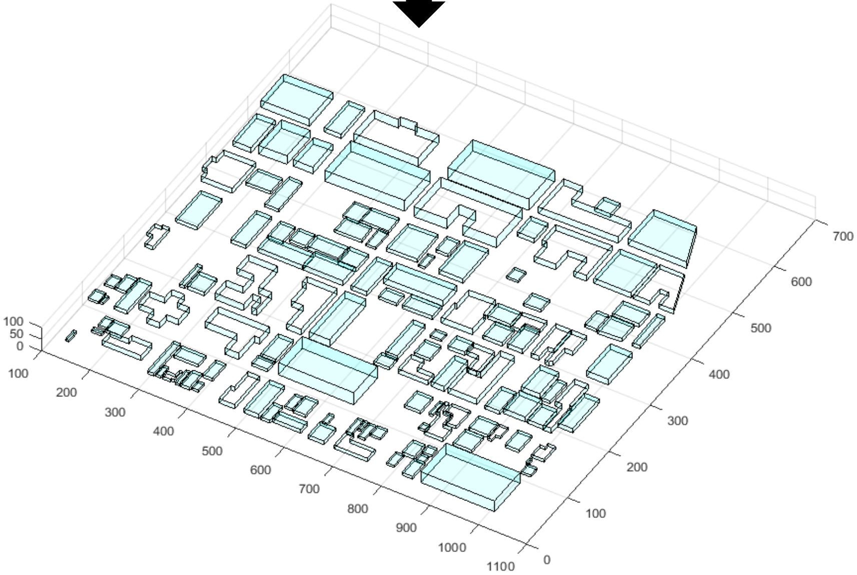

Start
MATLAB GIS interface
GIS database
Input variables initialization
N-rays, M-Rx points, Tx/Rx location, study area boundaries, geometry facets
Geometry SVP partitioning
Geodesic rays generator
Input: N
Output: rays, resolution angle
Rx location index
i= 1
Ray index
j= 1
SBR algorithm
Input: Tx/Rx location, voxilized facets, rays, resolution angle, maximum number of reflections
Output: hit points, reflection angles, ray directions, received ray index, path distance, number of reflections, W-factor, counter
Store RT information for ray-j, if detected as received, in a cell array
j= N
No
j= j+1
yes
E-Field calculator
Input: ray traversed directions, hit points, facets' normal directions, path distance, number of reflections, W-factors, Tx/Rx location
Output: complex e-field vectors, total Rx field, path loss
Store field calculations and path loss value for Rx position-i in a cell array
i= M
No
i= i+1
yes
Calculate theoretical e-field values (if applicable)
Plot results
Simulation elapsed time
End

Figure 2.8: Flow chart describing the MATLAB implementation of the ray tracer



Wireless Insite 3D urban scenario in '.city' format

MATLAB GIS interface



MATLAB 3D plot for the Wireless Insite scenario geometry

Figure 2.9: Reading the Wireless Insite 3D map in MATLAB

For the first receiver location point, the SBR algorithm starts tracing each of the N-generated rays with the voxels' registered facets in a fast and efficient manner as explained in Section 2.4.2. Tracing each ray continues until successfully received or reaches the maximum number of allowed reflections defined by the user. The receiver model employed in the SBR algorithm is the distributed wavefront model [50] highlighted in Section 2.3.1.2. The SBR algorithm is designed to consider direct and reflected rays. For each received ray, the SBR algorithm output contains information about: reflection (or hit) point at each facet, reflection angle at each facet, directions of incidence and reflection at each facet, facets' normal vector directions, total path distance, number of reflections, and the weighting factor. Besides, a counter variable tracks the number of conducted ray/facet intersection tests for further performance comparison between the SBR algorithm and its SVP accelerated version. This information is then stored for calculating the complex e-field for each ray. It should be noted that in all the considered propagation scenarios, the radio paths traced by the SBR algorithm are those due to reflections from static objects (*e.g.*, buildings) described by the GIS database of the propagation environment. Other scattering moving objects such as vehicles and people are not considered in our tests due to the difficulty of predicting their existence in a deterministic ray tracing environment.

After tracing all rays, the stored ray tracing information database will be used by the field calculation function to compute the complex polarized e-field vector associated with each received ray, as described in Section 2.3.3. After multiplying each e-field vector with its corresponding weighting factor, the weighted field vectors for all of the received rays are then added together to determine the final received field and path loss at the current receiver location.

The same process is repeated for all other receiver locations along the receiver's route. Based on the ray tracing scenario, and if applicable, the estimated path loss is then compared to theoretical calculations for measuring the accuracy of the ray tracing calculations. Further analysis for the processing time complexity of the ray tracing engine, which is not considered in this work, could be efficiently conducted using the MATLAB's built-in timer functions for speed comparison purposes.

2.7 Numerical Simulations and Verification with Commercial Software

A commercial propagation estimation tool, developed by Remcom Inc., named Wireless Insite [19], which is a popular product in the market and used extensively by researchers and engineers, has been selected to be running in parallel with the MATLAB ray tracer in order to enable comparison of results for verification purposes. It implements ray tracing models and EM calculations for predicting radio propagation characteristics in wireless environments. Insite has a user-friendly GUI which enables the designer to set the simulation parameters easily. It also provides useful interactive project view, various plots and 3D visualizations of the study area overlaid by the propagation paths and signal strength for each path. Moreover, it has the capability of creating user defined floor plans for studying indoor scenarios and buildings' structures.

To perform a comprehensive validation for the performance of the developed MATLAB ray tracer with the Wireless Insite, a complex simulation scenario is chosen. The selected scenario focuses on radio propagation in an urban wireless communication environment. However, as shown below, other prelminary tests have been conducted to verify the performance and integrity of the developed MATLAB ray tracer. These tests are: line-of-sight with fixed receiver distance, free space path loss and two-ray propagation [55]. The settings and results for each test are described in the following sections.

2.7.1 Line-of-Sight Test with Fixed Receiver Distance

The main purpose of this test is to validate the uniformity of the reception technique and highlight the effect of the geodesic rays aberration on the ray tracing accuracy. The test assumes that the transmitter is located at the center of a sphere, while the receiver is placed at different points along the sphere's surface. Ideally, all points on the surface of a sphere receive equal power. This test precedes the following two tests due to the importance of assuring that the generated tracing rays satisfy the properties explained in Section 2.3.1.1 before investigating a ray tracing scenario.

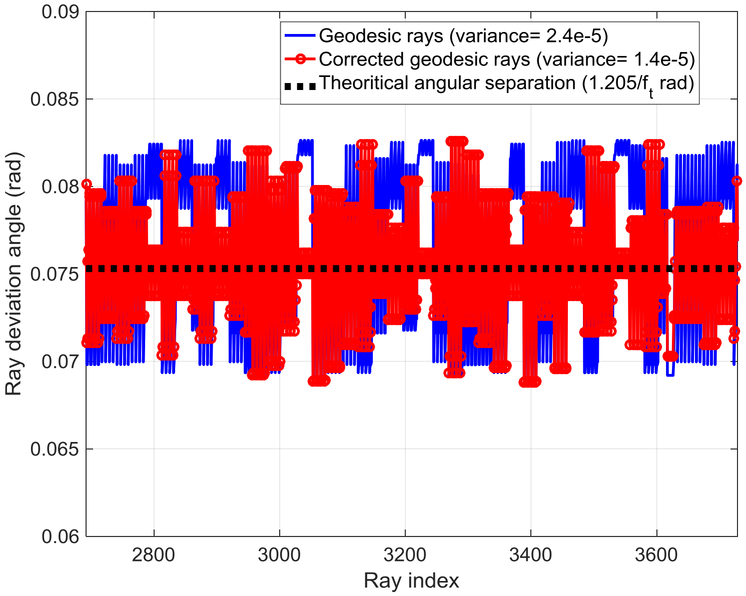


Figure 2.10: Snap shot for a sample of geodesic rays vs. corrected rays deviation angle

The calculation of the angular separation between rays shows that the geodesic ray launching method does not provide ideal uniform distribution to the rays as explained in [50]. To correct this, a simple heuristic approach is developed on MATLAB to finely adjust the transmission angle of the rays. The correction approach performs an iterative shifting process for the locations of the geodesic sphere vertices to establish equal angular distances between neighboring rays as much as possible. The idea is inspired by the movement of electron charges on the surface of a spherical conductor. By imagining the geodesic sphere vertices as electrons, the repulsion forces between points (*i.e.*, electrons) will keep pushing and displacing them along the sphere's surface until all points reach the equilibrium state. Consequently, a uniform spatial distribution of all points on the surface of the sphere is obtained. The equilibrium state occurs when all points on the geodesic sphere incur perfectly equal forces (*i.e.*, represented in MATLAB by the Euclidean distance) from the six neighboring points. It is worth noting that every point on the tessellated geodesic sphere, shown in Fig. 2.3a, has six hexagonal neighboring points, except for the original icosahedron vertices which have only five pentagonal shaped neighboring points instead.

The results of correcting a total of 2562 rays (*i.e.*, equivalent to a tessellation frequency of 16) are reported. Fig. 2.10 shows the aberration in the rays' angular separation

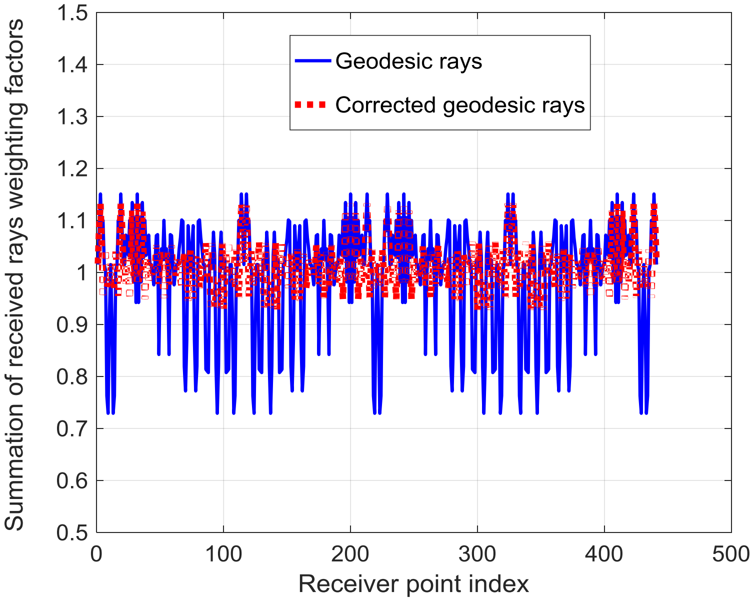


Figure 2.11: Geodesic rays vs. corrected rays weighting factors summation at different receiver points along the surface of a sphere

for both, the geodesic launching scheme and its corrected version (*i.e.*, explained in the previous paragraph), with respect to the constant theoretical value (*i.e.*, $1.205/f_t$ rad [50]). As expected, the observed variations confirm that geodesic rays are not 100 percent uniform. However, Fig. 2.10 shows that the developed correction method (used to correct the geodesic points' locations) improves the rays uniformity by decreasing the rays' angular separation variance by 41.6%. On the other hand, Fig. 2.11 shows the summation of the received rays' weighting factors at each receiver point along the sphere's surface. It is obvious that the weighting factors for both techniques do not exactly add up to one. However, the corrected geodesic rays show better accuracy compared to regular geodesic rays.

Further illustration for the improved geodesic uniformity by the implemented correction approach is depicted in figures 2.12 and 2.13, by looking at the weighting factors distribution along a geodesic sphere of receiving points for the geodesic and corrected geodesic rays, respectively. The geodesic aberration of rays is clearly shown in Fig. 2.12 as of the cyan colored triangular patterns of points on the sphere's surface. These points denote deviated rays from their target receiving points having weighting factors (*i.e.*, based on the ray deviation angle) less than one. On the other hand, the corrected geodesic rays show better uniform distribution of weights (*i.e.*, less pattering of colored points) along the surface of the sphere as shown in Fig. 2.13.

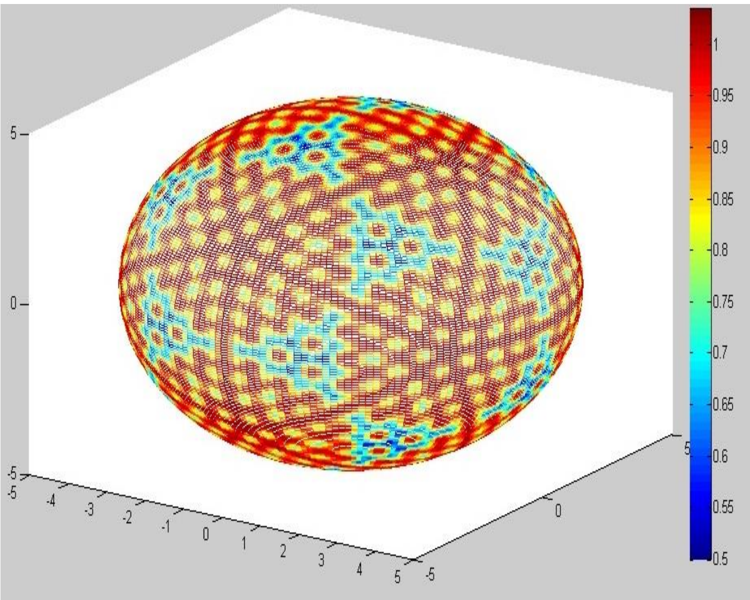


Figure 2.12: The distribution of weighting factors for the geodesic rays on a geodesic sphere centered at the transmitter

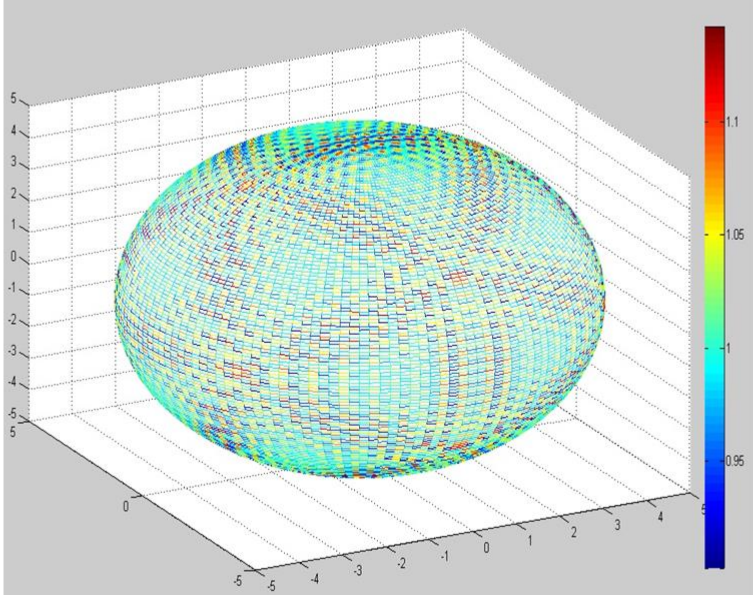
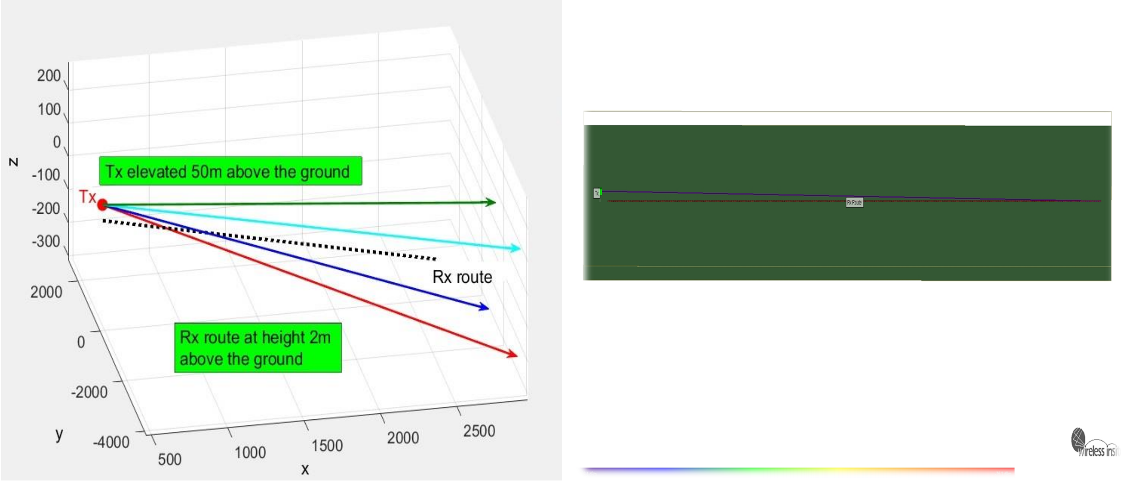


Figure 2.13: The distribution of weighting factors for the corrected geodesic rays on a geodesic sphere centered at the transmitter


200
100
0
z
-100
-200
-300
Tx elevated 50m above the ground
Tx
Rx route
2000
0
-2000
y
-4000
Rx route at height 2m above the ground
500
1000
1500
2000
2500
x

(a) MATLAB ray tracer 3D view for the FSPL test

(b) Wireless Insite 3D view for the FSPL test

Figure 2.14: 3D view for the Matlab/Wireless Insite FSPL test

2.7.2 Free Space Path Loss Test

This test is designed to compare the theoretical free space path loss (FSPL) with that estimated by the developed MATLAB ray tracer and the Wireless Insite software. Thus, it is the first test to measure the prediction accuracy of the developed MATLAB ray tracer against the Insite tool with respect to theory. The theoritical FSPL is given by the following formula [55]:

$$\mathrm{FSPL} = 10\log_{10}\left(\left(\frac{4\,\pi\,d\,f}{c}\right)^{2}\right)\ \mathrm{dB} \tag{2.4}$$

where d is the Rx-Tx separation distance in meters, f is the signal frequency in Hz and c is the speed of light in m/sec.

A three dimensional view of the test set-up used for both the MATLAB and Insite environments, is shown in Fig. 2.14. The transmitter and receiver antennas are elevated to 50m and 2m above the ground, respectively. The receiver is initially located 48m apart from the transmitter. Then it starts moving away from the transmitter in a straight path with a total route distance of 3.3km in 1.64m steps. At each step the path loss is calculated using MATLAB, Wireless Insite and theory. To match the theoretical calculations with the simulation environments, only the LOS ray is considered in the MATLAB and Insite ray tracing domains, while discarding ground reflections. The path loss predictions and calculations are as shown in Fig. 2.15. Generally speaking, the results show that both the MATLAB and the Insite ray tracing predictions for the path loss precisely match theoretical calculations. However, the MATLAB accuracy is higher compared to the Wireless Insite tool at small separation distances. This could be due to the unknown ray launching mechanism used by the Wireless Insite tool which misses the receiver location at short distances. In the case of MATLAB, the corrected geodesic mechanism explained in the previous test was able to hit the receiver location at close proximity.

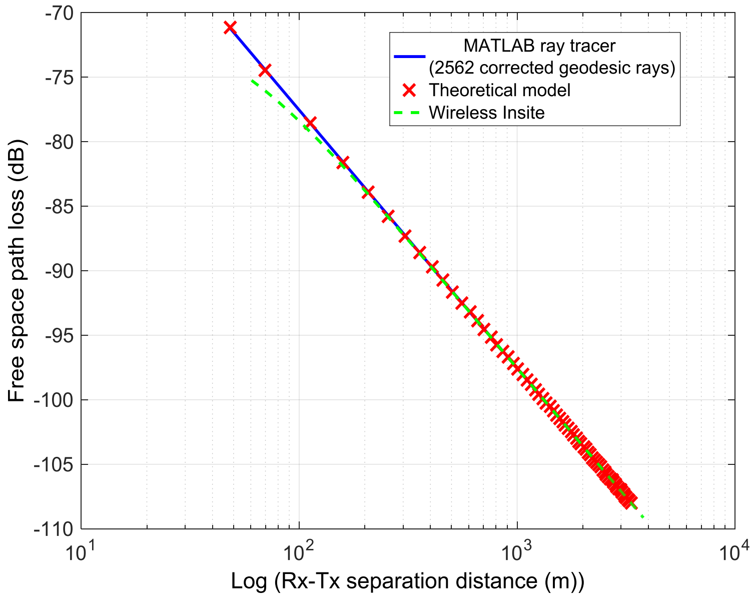


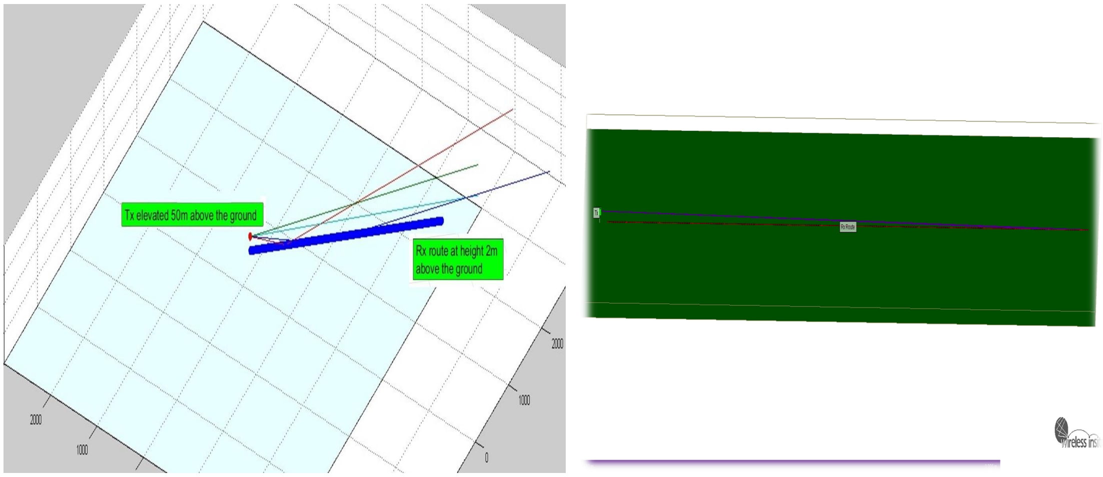
Figure 2.15: FSPL prediction using MATLAB ray tracer vs. Wireless Insite with respect to the theoretical model

2.7.3 Two Ray Model Test

This test is intended to investigate the channel fading effect, the phenomenon that is widely known to be one of the major factors contributing to the degradation of the wireless channel's quality (*i.e.*, lower capacity), which takes place due to the natural multipath propagation of wireless signals. The multipath propagation creates two receiving paths of the signal, one coming directly from the transmitter (*i.e.*, LOS ray) and the other is reflected from the ground. This is a standard and widely studied model in the literature and known to be very accurate. The model is represented by the following closed-form equation [55]:

$$\begin{aligned} \mathrm{PL}_{2ray} &= \; P_{tx}|_{\mathrm{dBm}} \; - \; P_{rx}|_{\mathrm{dBm}} \\ &= \; 40\log_{10}(d) - 20\log_{10}(h_t\, h_r) - 10\log_{10}(G_l) \qquad \mathrm{dB} \end{aligned} \tag{2.5}$$

where $P_{tx}|_{\mathrm{dBm}}$ is the transmit power in dBm, $P_{rx}|_{\mathrm{dBm}}$ is the receive power in dBm, h_t is the transmitter's elevation height above the ground in meters, h_r is the receiver's elevation height above the ground in meters and G_l is the product of the transmit and receive antenna field radiation patterns in the LOS direction.


Tx elevated 50m above the ground
Rx route at height 2m above the ground
2000
1000
2000
1000
0
Tx
Rx Route

(a) MATLAB ray tracer 3D view for the 2 ray test

(b) Wireless Insite 3D view for the 2 ray test

Figure 2.16: 3D view for the Matlab/Wireless Insite two ray model test

Snapshots from the Matlab and Wireless Insite simulation environments are depicted in Fig. 2.16. The heights for the transmitter and receiver antennas are assumed to have the same values with those used in the FSPL test. The ground has been assumed to be a perfectly conducting reflecting surface in both the MATALB and Insite domains. The results presented in Fig. 2.17 show very good agreement between both the MATLAB and Insite ray tracing engines and the theoretical calculations.

With multipath fading, the signal experiences consecutive constructive and destructive interferences of the two rays with the distance due to the difference in the received signal phase from each path. Thus, the signal envelope shows continuous rising and falling behavior. The free space path loss curve is added to the plot to show the signal attenuation trend. At a certain distance from the transmitter, which is proportional to the product of the antennas' heights, the oscillations disappear and the signal power falls off rapidly, proportionaly to $1/d^4$ [55].

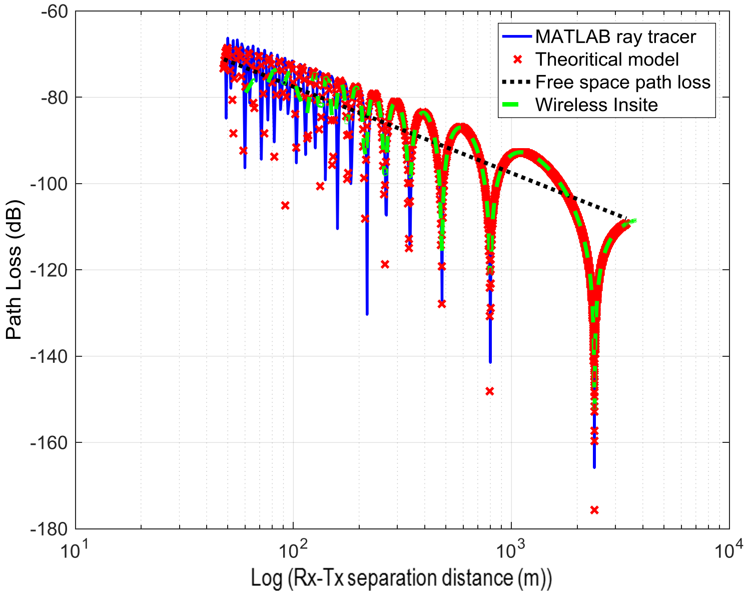


Figure 2.17: Path loss prediction for two ray model using MATLAB ray tracer vs. Wireless Insite with respect to the theoretical model

2.7.4 Urban Scenario Propagation Test

In this test, we compare the prediction accuracy of the developed MATLAB ray tracing system illustrated in Fig. 2.8 with respect to the Insite prediction results. Due to the lack of an accurate theoretical model to characterize the propagation in such scenario, which is primarily dependent on the environment's geometry and geographical description in general (*i.e.*, site specific), the MATLAB ray tracing prediction results is only compared with that of the Wireless Insite tool. In the same context, it is worth mentioning that an even more comprehensive model like the Ten-Ray model (Dielectric Canyon) [56] would not be adequate for this test as the number of rays received at each receiver location is changing based on the geometry and obstacles surrounding the receiver location.

The propagation scenario in this test is taking place in north of the centretown of the city of Ottawa, Canada. This scenario is pictured by the Google map view shown

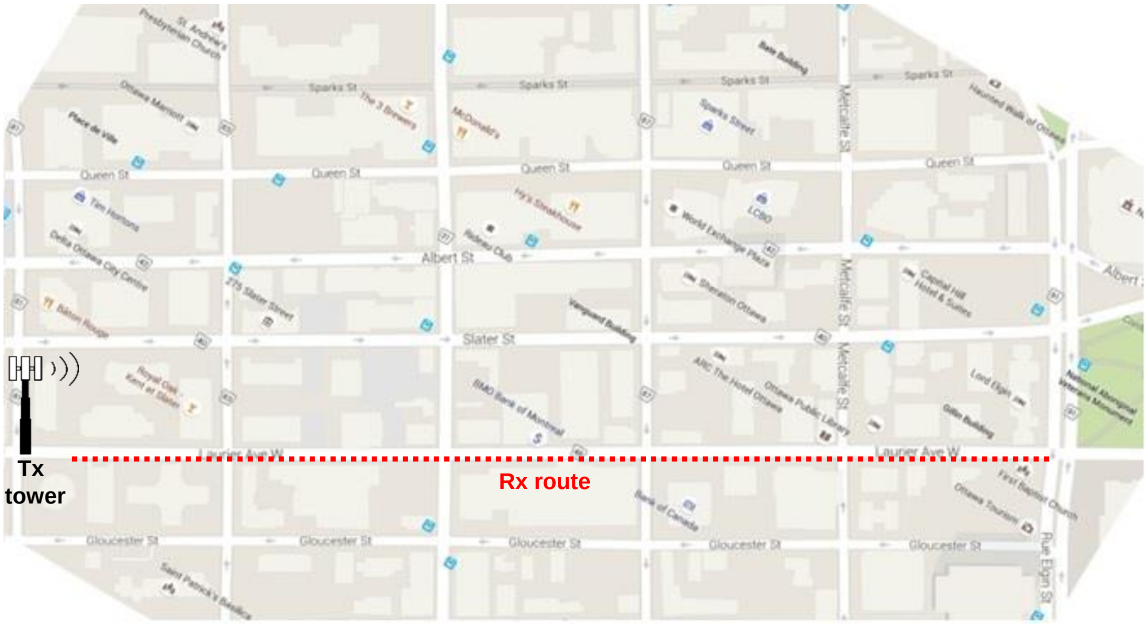
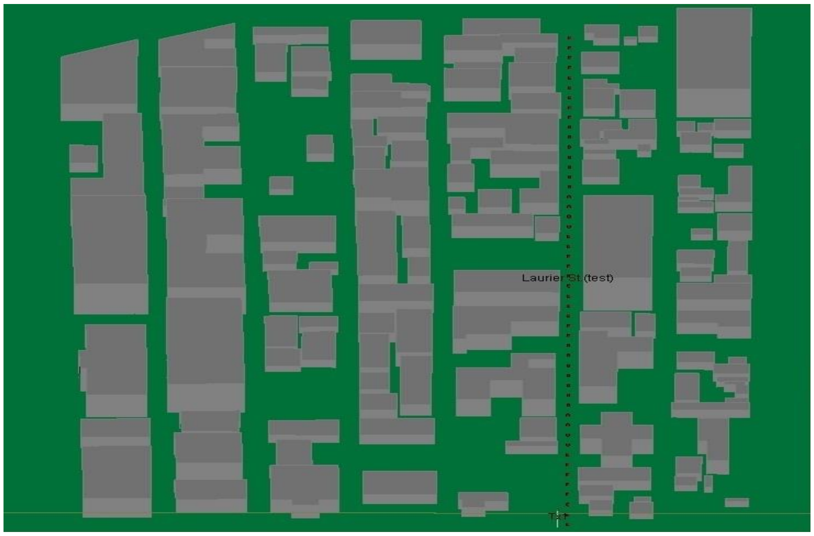


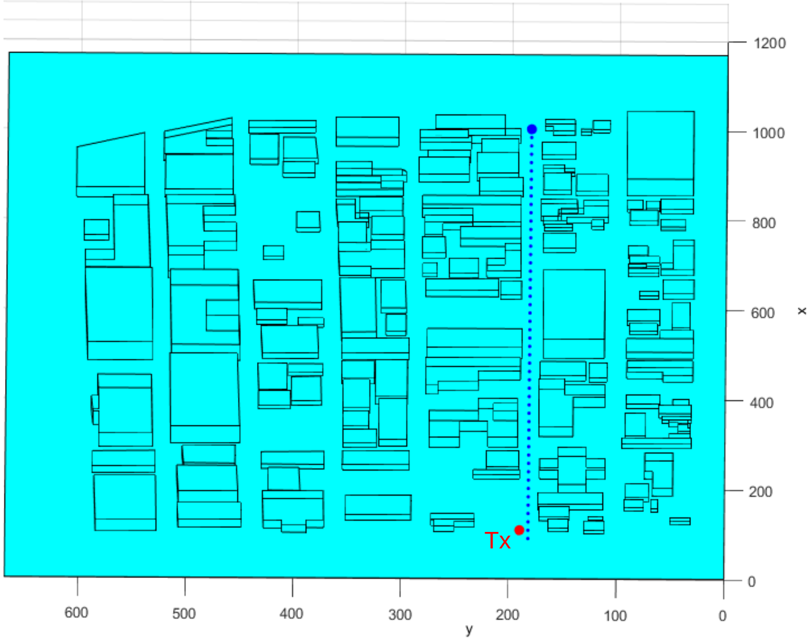
Figure 2.18: Google map view for the urban scenario environment

in Fig. 2.18. The transmitter tower is located at the intersection of Laurier Ave W with Lyon St N at a height of 28m above the ground. The receiver traverses Laurier Ave W till the intersection with Elgin St at a constant speed of 50km/h. The total distance covered by the receiver route is 913m. The receiver was initially 26.18m apart from the transmit tower location, and elevated at 3.5m above the ground. The path loss prediction in both, the MATLAB and Insite environments, is evaluated every 1.34sec (*i.e.*, 18.63m separation distance between two successive receiver points) during the receiver trip, resulting in a total of 50 points. The simulation environments for both the MATLAB and Wireless Insite, for the Google map view of Fig. 2.18, is depicted in Fig. 2.19.

The path loss prediction results for the MATLAB and Insite ray tracers are demonstrated in Fig. 2.20. The results show that the MATLAB calculations follow very closely to the Wireless Insite points except of at few receiver points. In particular, the two receiver points at distances of 208.6m and 245.6m from the transmitter show a remarkable prediction difference between the MATLAB and Wireless Insite. However, after debugging both of the MATLAB and Wireless Insite prediction results it has been revealed that the source of this discrepancy is solely coming from the Insite. In other words, when increasing the resolution of the Insite results (*i.e.*, increasing the number of receiver points at which the



(a) Wireless Insite urban propagation test



(b) MATLAB urban propagation test

Figure 2.19: Wireless Insite/MATLAB urban scenario simulation environments

path loss is calculated), the path loss values at the previously highlighted Rx-Tx separation distances have changed to match those calculated by MATLAB as illustrated in Fig. 2.21. This behavior is attributed to the fact that the Wireless Insite, based on a certain unknown criteria, neglects some of the possible received ray paths based on the number of calculation points (*i.e.*, number of receiver points at which the path loss is calculated). Consequently, the change in the number of traced rays at a receiver point results in a variation in the total received field, total received power, and hence the path loss. As a result of these findings, the developed MATLAB ray tracer shows high accuracy and consistency with respect to the Wireless Insite commercial software. Therefore, the MATLAB system could be reliably used for later verification with other software and hardware implementations. Finally, more insight on the identical ray tracing operation for both the MATLAB and the Wirless Insite ray tracers can be deduced from the results demonstrated in Fig. 2.22.

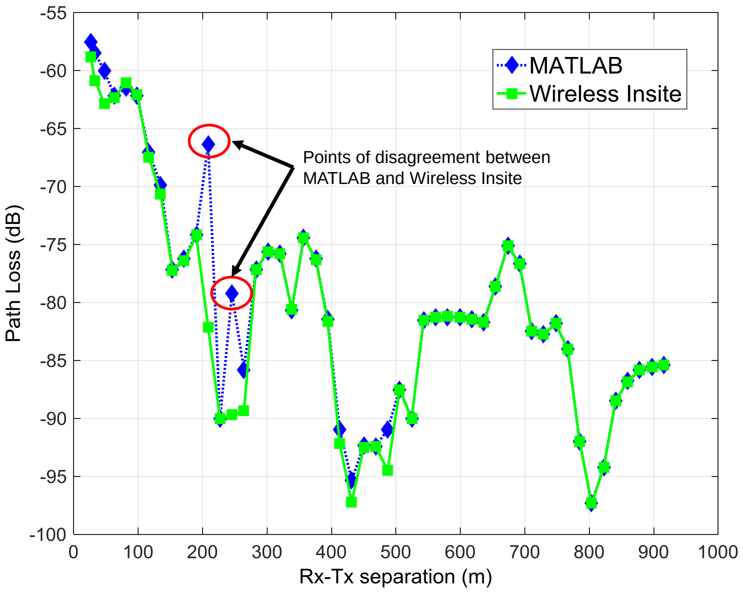


Figure 2.20: The urban scenario path loss prediction results for MATLAB vs. Wireless Insite

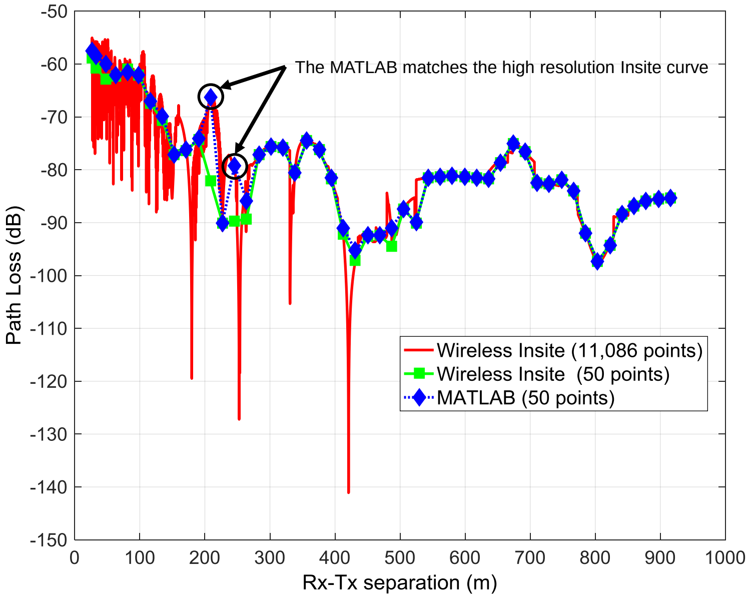
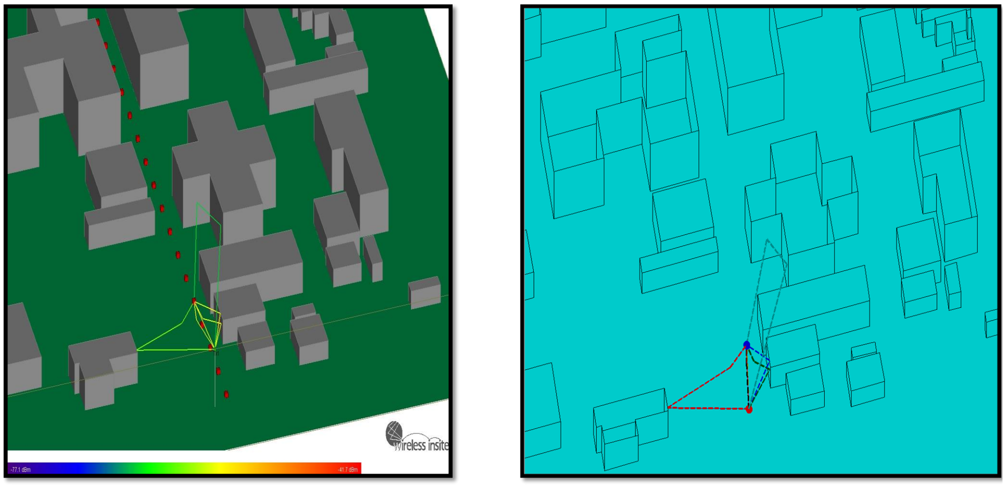


Figure 2.21: Debugging the Wireless Insite resolution problem



(a) Wireless Insite received ray paths at Rx #5 (b) MATLAB received ray paths at Rx #5

Figure 2.22: Received ray paths traced by the MATLAB and Wireless Insite at receiver point number 5

2.8 Chapter Summary

In this chapter, a comprehensive survey on the ray tracing model for radio propagation prediction is presented. The survey is provided in two parts. Part one covers the motivation, applications, recent trends, theory, and most prominent algorithms of the ray tracing propagation model. On the other hand, part two focuses on the proposed ray tracing system architecture, system level implementation, and numerical evaluation and testing. The chapter delivers solid understanding of the ray tracing problem and the software implementation challenges. Finally, the MATLAB ray tracing library developed at the end of this chapter provides the reader deep and detailed insight on the ray tracing theory of operation especially after being verified with the commercial Wireless Insite tool. Besides, it serves as a solid software implementation for validating different software implementations (*e.g.*, CPU based) and even enabling future efficient hardware counterparts (*e.g.*, FPGA and GPU based).

Chapter 3

On a Throughput-Efficient Look-Forward Channel-Aware Scheduling

3.1 Introduction

The continuing growth of mobile users and exigent demands for high QoS data communications requires designing high-speed and efficient wireless networks. Obviously, this creates a two-faced problem for researchers to solve. The first face is the capacity limitation of wireless channels, which weakens the ability of wireless devices to send and receive digital information reliably with high data rates. In this context, many advanced transmission techniques have been proposed to mitigate channel impairments, such as: wideband code division multiple access (WCDMA), multiple-input multiple-output (MIMO) and orthogonal frequency-division multiplexing (OFDM). The second face of the problem is related to the scarcity of the available spectrum, which is shared by multiple users. To deal with this issue, channel-aware scheduling strategies [58] were designed to adaptively adjust transmission parameters (*e.g.*, modulation and coding schemes) with temporal channel conditions and dynamically manage network resources (*e.g.*, power, bandwidth) among competing users. This could only be done by having periodic knowledge about a user's channel state information (CSI). The goal is always defined to maintain a challenging balance between throughput, delay and fairness.

To acquire knowledge about channel, different propagation models have been developed for both indoor and outdoor environments [25]. In this work, special attention

---

A version of this chapter has been published in [57].

is given to site-specific models for outdoor applications. More specifically, we consider the ray tracing (RT) model to be a potential approach which could be utilized to assist in allocating network resources more effectively. The rationale for using the RT channel prediction model is discussed later in Section 3.3.

Extensive research has been conducted on channel-aware scheduling algorithms that utilize periodic link adaptation to enhance transmission efficiency. In this chapter, we highlight only some of the recent studies. In [1], the authors developed two power-efficient schedulers for mixed streaming services in LTE uplink systems which offered a remarkable transmission power reduction compared to the proportional fair (PF) and the energy aware resource allocation (EARA) [59] schedulers constrained by QoS requirements. In [60], the authors of [1] have further enhanced the power efficiency of the user equipment (UE) by controlling the maximum allowable transmit power (MATP) with respect to the user's buffer queue length. In a related context, the authors in [61] utilized the prediction of incoming traffic for building bandwidth-efficient scheduling algorithms in hybrid wireless optical networks. It has been noted that, from the channel point of view, most scheduling algorithms usually adjust their scheduling decisions according to the user's channel quality for, at most, a single transmission frame at a time. However, more information about the future of the channel could be extracted by tracing mobile radio paths in known environments.

To the best of our knowledge, no published work considers the impact of utilizing multiple future channel frames for each user, when scheduling radio resources, on the network's spectral efficiency. Despite the fact that the framework presented in [62] introduced an idea about the predictive scheduling over wireless links, the considered scheduling model did not clearly indicate how the future channel conditions could be practically obtained. Also, the assumption made was for limited prediction time horizon of 10msec. Therefore, this chapter investigates the effect of considering future channel states in making decisions for efficiently allocating network resources. The results reported in this chapter are presented in two parts: First, we study the temporal channel outage probability and derive an approximate closed-form for it as a function of the adaptation horizon. Second, we build on the above mentioned outage analysis to investigate the performance of the conventional maximum throughput (MT) scheduler constrained by the Max/Min fairness criteria. The key strategy of the proposed scheduler is based on utilizing long-term channel prediction provided by a fast RT engine for increasing the system's average throughput and alleviating the prospective high channel outage when scheduling users over long time intervals.

The rest of this chapter is organized as follows. Section 3.2 presents an analytical formulation of the channel outage probability. The ray tracing prediction model and the rationale for using it with the proposed scheduler is explained in Section 3.3. In Section 3.4, the difference between the conventional channel-aware scheduler and the proposed scheduler is depicted. In addition, the optimal formulation and a low complexity heuritstic algorithm for the proposed scheduler are presented. Simulation results are reported in Section 3.5, and finally Section 3.6 summarizes this chapter.

3.2 Link Adaptation Analysis

Multipath propagation is known to cause signal fading and to affect the overall performance of wireless communication systems, especially for moving user terminals. In these situations the temporal random variation of the channel impulse response that is due to rapid geometrical variation of the environment surrounding the transmitter and receiver, significantly affects the link rate reliability. Therefore, in this analysis we investigate the effect of changing the adaptation time horizon of the signal transmission rate on the channel outage probability. Motivated by the RT-based model proposed in the following sections, which is known to predict the channel characteristics due to the Multipath propagation, the only channel impairment considered in this analysis is the Mutipath fading.

In our analysis, the propagation channel between transmitter and receiver is modeled as a Gaussian random variable [14]. Random variations are due to relative changes in the geometric location for any of them which creates different propagation paths from the surrounding objects (*e.g.*, buildings). This results in different received power levels and signal-to-noise ratio (SNR) at the receiver terminal. Consequently, transmission data rate should not remain constant: it should be adaptively controlled to match the channel capacity within a suitable observation period. Considering single carrier system, the instantaneous channel capacity based on the instantaneous receiver's SNR could be expressed by

Shannon's channel capacity theorem [14] as follows:

$$C(t) = \log_2[1 + \gamma(t)] \tag{3.1}$$

where $C(t)$ is the instantaneous capacity per unit bandwidth, and $\gamma(t)$ is the instantaneous SNR. In order to ensure a low information loss rate, the instantaneous link rate $R(t)$ must be always kept lower than $C(t)$. Hence, the channel outage probability at a given reference SNR γ_o is defined as:

$$P_{out}(\gamma_o) = \text{Prob}\left\{R_o \, > \, C(t)\right\} = \int\limits_0^{\gamma_o} p_\gamma(\gamma)\, d\gamma = P(\gamma_o) \tag{3.2}$$

where $R_o = \log_2(1 + \gamma_o)$ is the desired transmission rate, $p_\gamma(\gamma)$ is the probability density function (PDF) of the instantaneous channel SNR (*e.g.*, exponential distribution for rayleigh fading channel), and $P(\gamma_o)$ is the corresponding cumulative distribution function (CDF). Ideally, we can say that the outage probability will approximately reach zero based on having complete information about the instantaneous channel SNR $\gamma_n = \gamma(t_n)$ at every time instant $t_n = nT_s$. Hence, the rate R_n will converge to $\log_2(1+\gamma_n)$ which theoretically matches the instantaneous channel capacity. Thus, we can say that the average transmission rate over a long period of time will be:

$$\overline{R} \, = \, \overline{C} \, = \int\limits_0^{\infty} \log_2\left(1 + \gamma\right)\, p_\gamma(\gamma)\, d\gamma \tag{3.3}$$

In practice, the instantaneous adaptation is impossible to implement. Therefore, slow adaptation based on a partially known channel will be a more realistic approach. In this case the rate $R_q = \log_2[1 + \gamma(qNT_s)]$ is adjusted at fixed time intervals $t_{qN} = qNT_s$, where N is the number of transmissions within each adaptation block, and q is the number of blocks considered in the adaptation horizon. Usually the rate is selected at the beginning of each block, and then kept constant for the subsequent N transmissions. Therefore, the adaptation horizon must be carefully chosen to make sure that the channel will experience little or no outage. In general, the outage probability will increase as the adaptation horizon increases. For example, letting $T = NT_s$, increasing T to a value much greater than the channel coherence time τ_{coh} will probably lead to observing many variations of the channel within the same horizon. In this context and from (3.2), the upper bound for the probability of losing data packets within the chosen adaptation horizon will be:

$$\lim_{T \to \infty} P_{out}\left(T|\gamma_o\right) \;=\; P(\gamma_o) \tag{3.4}$$

The result of (3.4) represents the probability of having the instantaneous SNR falling below the specified value γ_o taken at the beginning of the adaptation block when $T >> \tau_{coh}$. On the other hand, if $T << \tau_{coh}$, then P_{out} will be given by equation 8.163 in [63] as follows:

$$P_{out}(T|\gamma_o) \sim \; LCR(\gamma_o)\,T \;=\; \sqrt{\frac{-\rho''(\tau=0)}{2\pi}}\,p_\gamma(\gamma_o)\,T \tag{3.5}$$

where LCR is the level crossing rate at a reference channel SNR γ_o within T, and $\rho''(\tau)$ is the second derivative of the SNR correlation coefficient (that measures the coherence of the channel in an observed interval). After doing some simplifications (*i.e.*, $LCR(\gamma_o)T << 1$), equation (3.5) could be approximated as:

$$P_{out}(T|\gamma_o) = \; P(\gamma_o)\,\left[1 - e^{-LCR(\gamma_o)T}\right] \tag{3.6}$$

The result obtained in (3.6) will lead us to measure the amount of information which has been successfully transmitted. Thus, for one block interval $[qNT_s : (q+1)NT_s]$, the total amount of information sent will be:

$$I_{sent}(T|\gamma_o) = \; \log_2(1+\gamma_o)\,T \tag{3.7}$$

whereas, the effective amount of information transmitted reliably is:

$$I^R_{sent}(T|\gamma_o) = \; \left[1 - P_{out}(T|\gamma_o)\right] I_{sent}(T|\gamma_o) \tag{3.8}$$

Then from (3.3), (3.6) and (3.8), the average reliable transmission rate over long transmission session will be:

$$\overline{R}^{R}(T) \,=\, \frac{1}{T}\int\limits_{0}^{\infty} I_{sent}^{R}(T|\gamma_o)\, p_\gamma(\gamma_o)\, d\gamma_o \tag{3.9}$$

since $\overline{R}^{R}(0) \,=\, \overline{C}$ and $0 < \overline{R}^{R}(\infty) \leq \overline{C}$, then

$$\overline{R}^{R}(T) \,\approx\, \overline{R}^{R}(\infty) \,+\, [\overline{C} \,-\, \overline{R}^{R}(\infty)]\, e^{-\alpha T} \tag{3.10}$$

To find α we substitute (3.5) and (3.8) in (3.9) which gives:

$$\overline{R}^{R}(T) \,=\, \overline{C} \,-\, \alpha T \tag{3.11}$$

where, $\alpha = \sqrt{\frac{-\rho''(0)}{2\pi}}\ \int\limits_{0}^{\infty} \log_2(1+\gamma_o)\, p_\gamma^2(\gamma_o)\, d\gamma_o$

(3.6) and (3.11) provide us with closed form expressions of the variation of the outage probability and the average reliable transmission rate with respect to the adaptation horizon T, respectively.

For the purpose of numerically evaluating the previous results, a MATLAB simulation has been conducted using the standard COST207 multipath channel model [64] to investigate the variation of the outage probability versus γ_o and T. The simulation was done twice at different speeds (25Kph and 50Kph) for the mobile terminal to account for the Doppler effect as well. The curves shown in Fig. 3.1 illustrate the effect of increasing the adaptation horizon T from 10msec to 100msec at the two aforementioned speeds. The results show that increasing T dominates the speed increase as having the outage probability nearly the same at both speeds when T= 100msec. Also the intersection of the 10msec curves (*i.e.*, marked with square and circle) for small values of γ_o is due to shortage in adaptation time recovering from the outage especially when the channel is slowly varying at low mobile speeds. However, the saturation levels for the outage probability is generally noticed to be higher at larger T and speed values. In conclusion, the results reported in Fig. 3.1 provide an insight of the adaptation horizon for optimization in the network layer.

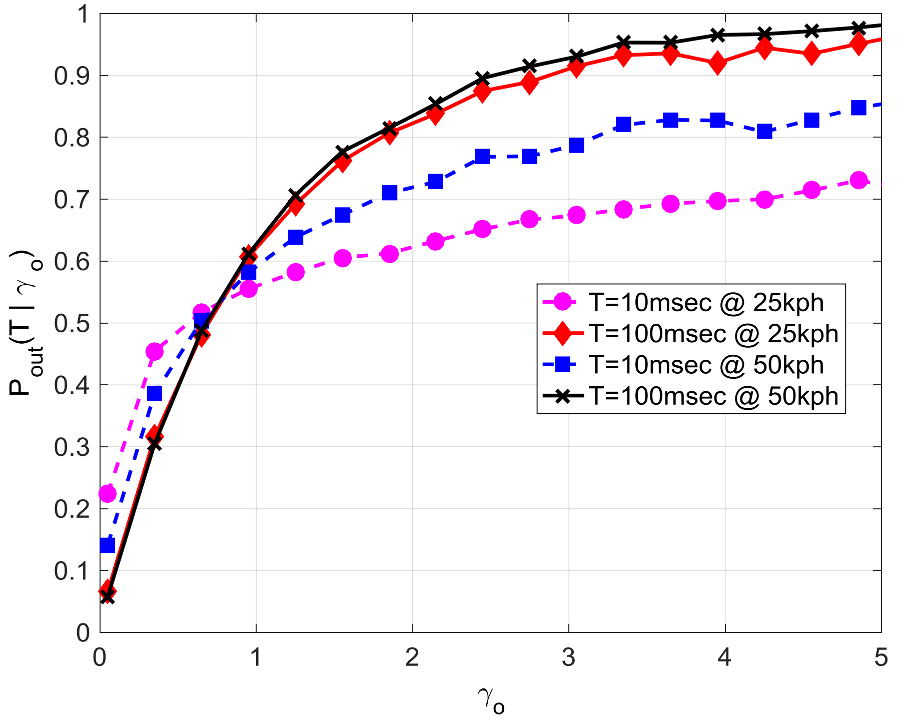


Figure 3.1: Outage probability at different mobile velocities and T values

3.3 Ray Tracing Based Prediction

Ray tracing is a site-specific approach which falls under the category of deterministic modeling. It has been under investigation for the last four decades in much of the research done for efficient utilization in radio propagation prediction [36]. Initial work was mainly focused on the feasibility of applying RT in the field of electromagnetic propagation. Following the verification stage, the research then steered towards optimizing the speed and accuracy of different ray tracing algorithms [37].

The major advantage behind using a RT model for providing the scheduler with the required CSI of the mobile users is being a promising propagation model for real-time applications. It could be efficiently implemented using today's high performance computing platforms such as field programmable gate arrays (FPGAs), graphics processing units (GPUs) and advanced digital signal processors (DSPs), or even a combination in order to meet the required computation power. The details on how to implement such a system is beyond the scope of this chapter. However, by having the RT engine implemented on a suitable hybrid platform at the base-station (BS) site, together with advanced localization systems, the BS would be capable of pre-estimating the mobile users' propagation channel behavior for specific sections of their trip routes during their access periods. The calculations can be done offline as well. In this case a database of the prediction results is required to provide information about the channel, thus removing the processing burden from the transceiver. Hence, the scheduler could gain from future information about the channel, provided by the RT engine, for each user to be efficiently assigned to their optimum time-slots and frequencies. This effectively ensures reliable transmission sessions with high throughput for longer time intervals. Although these time intervals are fraction of a second (*i.e.*, multiples of 10msec radio frame) which is short enough to ensure fixed channel conditions as predicted by the RT engine, yet predicting it notably affects the scheduling performance (*i.e.*, system's throughput) as will be highlighted later in Section 3.5. Furthermore, despite the RT prediction error with respect to the accuracy of locating the mobile terminal in the environment is beyond the scope of this thesis, the following numerical example would give the reader general idea about it. Suppose that a mobile user is moving in an outdoor environment with an average speed of 50kph. Having the widely used assumption in the literature [65] of block fading channel within a single radio frame of 10msec, the RT prediction is error free for up to 0.138m error in the device location. Thus, increasing the mobile speed consequently increases the tolerable error for the device location.

3.4 Channel-Aware Scheduling

Channel-aware scheduling strategies [12] are proposed to adaptively match the transmission parameters (*e.g.*, power, modulation and coding schemes) and the resource allocation scheme to the CSI. For instance, by setting the system's spectral efficiency as an objective, the scheduler gives higher allocation priority to users experiencing good channel conditions (*i.e.*, users that can achieve high throughput) to transmit their data packets. In this case, the channel-aware scheduler certainly makes use of the independent channel variations across users (*i.e.*, multi-user diversity) to improve the system's spectral efficiency. This property will be quantitatively illustrated in Section 3.5.

3.4.1 Conventional Channel-Aware Scheduler

In order to comparatively test the proposed RT-based scheduler performance which will be discussed in the next section, we first built the conventional scheduler that is based on having the user's CSI available for single radio frame. The conventional scheduler strategy aims at maximizing the throughput while maintaining Max/Min fairness [66]. This means that the scheduler will allocate the available resources in each time slot to the user with maximum achievable throughput constrained by granting equal allocations for all users in a periodic manner. Hence, we denote this scheduler as MT-Max/Min throughout the rest of the chapter. The MT-Max/Min scheduler prioritizes users according to their CSI within only one frame for each transmission cycle (*i.e.*, a transmission cycle is equivalent to the number of frames such that each user will utilize an equal number of resources compared to others for one time). The scheduler keeps only the remaining unassigned users from previous frames for next frame assignments till the end of the cycle is reached. For example, consider a simple scenario of 20 users and a single carrier channel (*i.e.*, defined as a frequency resource till the end of this chapter) which is time-shared over 10 time-slots of a frame (*i.e.*, 10 resources are available per frame). The MT-Max/Min scheduler will start by assigning the best 10 high channel quality users to the first frame slots, while the remaining 10 users will be assigned accordingly in the next frame based on their updated CSI. At this point the scheduler finishes one complete transmission cycle (*i.e.*, 2 frames in this example) in which all users have been equally granted one frequency resource for one time.

3.4.2 The Proposed Optimal RT-Assisted Scheduler

Based on the channel outage analysis highlighted in Section 3.2, we propose an RT-based predictive scheduling scheme to avoid the channel outage appeared when scheduling multiple future frames. Unlike the MT-Max/Min scheduler, the proposed RT version will perform the scheduling operation on a larger time-scale based on the predicted channel information in the future frames for each user, that is provided by the RT engine as described in Section 3.3. It is worth mentioning that the RT prediction provides comprehensive information about the propagation channel such as the received signal power, propagation paths,

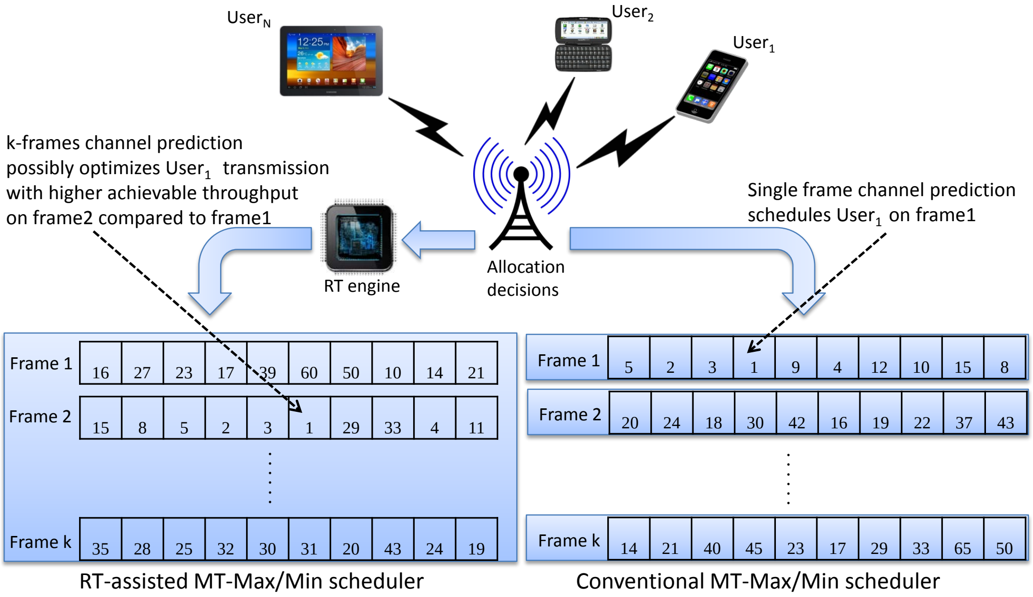

User$_N$
User$_2$
User$_1$
k-frames channel prediction
possibly optimizes User$_1$ transmission
with higher achievable throughput
on frame2 compared to frame1
RT engine
Allocation
decisions
Single frame channel prediction
schedules User$_1$ on frame1
Frame 1 16 27 23 17 39 60 50 10 14 21
Frame 2 15 8 5 2 3 1 29 33 4 11
Frame k 35 28 25 32 30 31 20 43 24 19
Frame 1 5 2 3 1 9 4 12 10 15 8
Frame 2 20 24 18 30 42 16 19 22 37 43
Frame k 14 21 40 45 23 17 29 33 65 50
RT-assisted MT-Max/Min scheduler
Conventional MT-Max/Min scheduler

Figure 3.2: Proposed RT-assisted vs conventional MT-Max/Min scheduler

time of arrival, delay spread, electric field magnitude and phase, directions of arrival and departure, carrier-to-interferer ratio, and the channel's impulse and frequency responses. However, without loss of generality, only the received signal power is the main channel characteristic considered in all simulations executed throughout this thesis due its direct impact on the channel capacity, and thus, the scheduling decisions in a single cell scenario (*i.e.*, exclusively considered in this thesis). This can be illustrated with the aid of Fig. 3.2. It shows that the proposed scheduler has a bigger solution space over k frames compared to the conventional single frame scheduler. The variable k denotes the channel prediction time range (*i.e.*, awareness) of the ray tracing engine which provides the BS's central scheduler with the users' CSI. Therefore, instead of just a single frame, the RT-based scheduler will be able to process k frames at a time until the end of each transmission cycle. This enables the scheduler to optimize the user's transmission rate by searching for the best time slot with the respect to the channel quality on a bigger time range compared to single frame based optimization. Increasing the prediction range (k-value) will allow the scheduler to process more frames at once and perform better long-term allocation of resources. It should be noted that the CSI is updated every frame in all cases.

Here, we assume that both of the MT-Max/Min and RT-based schedulers are seeking to optimize the system's average throughput per frame which is defined as:

Maximize

$$TP = \frac{1}{M} \sum_{j=1}^{M} \sum_{i=1}^{N} \Psi_{ij} R_{ij} \tag{3.12a}$$

Subject to

$$\begin{aligned} &\sum_{j=j_o}^{j_o-1+N/L} \Psi_{1j} = \sum_{j=j_o}^{j_o-1+N/L} \Psi_{2j} = \text{.........} = \sum_{j=j_o}^{j_o-1+N/L} \Psi_{Nj} = r\,, \\ &\forall\, j_o \in \left\{1, \tfrac{N}{L}+1, \tfrac{2N}{L}+1, \tfrac{3N}{L}+1, ..., M-\tfrac{N}{L}+1\right\} \end{aligned} \tag{3.12b}$$

where TP is the system's average throughput per frame, R_{ij} is the maximum throughput that user-i could achieve when transmitting over frame-j in order to maximize the system's average throughput per frame, N is the total number of users (N is assumed to be an integer multiple of L), M is the total number of observed frames (*i.e.*, simulation time), $\Psi_{ij} \in \{0, 1\}$ is a binary decision variable denoting whether user-i transmits on frame-j or not, L is the frame length in time slots (we set L to be equal to 10), and r is the number of frequency resources available in each time slot. For simplicity, we assume that $r = 1$ until the rest of the chapter.

The constraint defined by (3.12b) corresponds to the Max/Min fairness criteria which ensures that all users will be assigned an equal number of resources in each transmission cycle. It should also be noted that the transmission cycle length in frames is equal to $\left\lceil \frac{N}{L} \right\rceil$.

The solution of (3.12) will be straight forward in the case of MT-Max/Min scheduler because, as mentioned earlier, the scheduler will process a single frame at each step to allocate the best users among those remaining in the queue buffer. However, solving (3.12) in the case of the RT-based scheduler is more complex as the scheduler has to simultaneously deal with k-frames at every step within the transmission cycle. Thus, the solution will require solving $\frac{N}{kL}$ partial multi-dimensional binary integer programming (BIP) optimization problems within each transmission cycle. The solution of each of these problems will find the optimal allocations in the observed k frames at a time. The formulation for each of these problems will be as follows:

Maximize

$$TP_{j_o}{}^{t} = \sum_{j=j_o}^{j_o+k-1} \sum_{i \in N_b} \Psi_{ij}{}^{t} R_{ij}{}^{t} \tag{3.13a}$$

Subject to

$$\sum_{i \in N_b} \Psi_{ij}{}^{t} \leq L \quad , \forall j \in \{j_o, j_o+1, ..., j_o+k-1\} \tag{3.13b}$$

$$\sum_{j=j_o}^{j_o+k-1} \Psi_{ij}{}^{t} \leq r \quad , \forall i \in N_b \tag{3.13c}$$

$$\Psi_{ij}{}^{t} \in \{0, 1\} \tag{3.13d}$$

where: $TP_{j_o}{}^{t}$ is the total throughput attained at transmission cycle-t (t $\in \{1, 2, ..., \frac{LM}{N}\}$) for the frames index $[j_o, j_o+1, j_o+2, ..., j_o+k-1]$, N_b is a set containing indexes of the remaining unscheduled users waiting in the queue buffer, $R_{ij}{}^{t}$ is the maximum attainable throughput for user-i when transmitting over frame-j within transmission cycle-t, $\Psi_{ij}{}^{t} \in \{0, 1\}$ is a binary decision variable denoting whether user-i will be transmitting over frame-j within cycle-t or not.

The first constraint in (3.13b) ensures that the number of assignments for each frame matches its available resources, while the second constraint in (3.13c) is similar to that in (3.12b) which maintains equal number of resource allocations for all users within the transmission cycle (*i.e.*, Max/Min fairness).

3.4.3 Heuristic RT-Assisted Scheduler

Although the optimal formulation in Section 3.4.2 yields the best results our proposed RT-based scheduler could achieve, an alternative heuristic approach was investigated. The approach aims to relax the computational complexity (*i.e.*, highlighted later in Section 3.5) required in solving the $\frac{N}{kL}$ multi-dimensional BIP optimization problems in each transmission cycle, especially for large values of N. Table 3.1 shows the pseudo-code for our heuristic algorithm. It can be explained as follows: In each new transmission cycle, starting at Line 3, the algorithm targets allocating the available resources on a total of

Table 3.1: Heuristic Algorithm For RT-Based Scheduler

```
**Require:** $R_{ij}, i = 1, 2, \ldots, N, \;\; \forall j \in \{1, 2, \ldots, M\}$
$TP = 0$
**for** $j_o = 1$ **to** $M - \frac{N}{L} + 1$ **increment by** $\frac{N}{L}$ **do**
    **for** $j = j_o$ **to** $j_o + \frac{N}{L} - k$ **increment by** $k$ **do**
        **if** $j = j_o$ **then**
            $ReqBuffer = 1 : N$
            $[S_j, Id_j] = Sort(R_{ij}), \; \forall j \in \{j_o, j_o + 1, \ldots, j_o + k - 1\}, \; \forall i$
            $[\Psi_{ij}, TP_{ij}] = \Gamma(S_j, Id_j)$
            $TP = TP + \sum_{j=j_o}^{j_o+k-1} \sum_{i=1}^{N} \Psi_{ij} TP_{ij}$
            **Update** $ReqBuffer$
        **else**
            $[S_J, Id_J] = Sort(R_{iJ}),$
                    $\forall J \in \{j, j + 1, \ldots, j + k - 1\}, \; i \in ReqBuffer$
            $[\Psi_{iJ}, TP_{iJ}] = \Gamma(S_J, Id_J)$
            $TP = TP + \sum_{J=j}^{j+k-1} \sum_{i \in N_b} \Psi_{iJ} TP_{iJ}$
            **Update** $ReqBuffer$
        **end if**
    **end for**
**end for**
$TP = \frac{TP}{M}$
```

$\frac{N}{L}$-frames to N-users in k-steps (*i.e.*, the proposed scheduler works on k-frames at a time, as previously discussed). In order to process the first set of k-frames (Line 5) in the transmission cycle, the requests buffer (i.e, $ReqBuffer$) is first initialized with the indexes of all users requesting access (Line 6). All users are then sorted in descending order according to their MT-metric on each of the k-frames (Line 7). The Γ-algorithm (Line 8) then takes the users' sorted indexes (Id) and their corresponding throughput values (S) and outputs two $k \times N$ matrices, Ψ_{ij} and TP_{ij}, which represents the binary allocation decisions and scheduled throughput values, respectively. The Γ-algorithm is based on the greedy strategy where, for each user, it keeps searching for the best frame (*i.e.*, to transmit over) among k-frames using simple comparison operations for the predicted throughput values. In particular the algorithm starts by populating each of the k-frames with users transmitting with the maximum throughput compared to other frames using the users' sorted order (Id). Then it keeps distributing excess users in overpopulated frames (*i.e.*, frames having users exceeding its available L resources) in the same fashion to frames with empty allocations until all of the k-frames are completely allocated. The aggregate throughput over each frame is then directly calculated (Line 9) from the two output matrices, as in (3.13a), and summed with the previous frames. The users' requests buffer is then updated (Line 10) leaving only the remaining unscheduled users. The same process will be continuously repeated (Lines 12-15) through the loop of Line 4 but only on the remaining unscheduled users (*i.e.*, N_b) stored in the buffer until the final set of frames in the transmission cycle is completely assigned. The average throughput per frame is then calculated (Line 19) at the end by dividing the summation of the aggregate throughput over all frames by their number M.

As can be understood from the previous discussion, the heuristic algorithm depends mainly on two functions: Sorting (Lines 7 and 12) and Finding a maximum value (Lines 8 and 13). The most complex sorting operation occurs at the start of each new transmission cycle (Line 7), where N users are sorted on k-frames, and is equal to O(kNlog(N)) (*i.e.*, the $ReqBuffer$ is full of all users). On the other hand, finding a maximum, that is the major operation of the Γ-algorithm has a complexity of O(k) for each user. The worst case for this operation is when all users seek to transmit on the same frame. The first round of populating N users on k-frames will require O(kN) operations. The second round will accordingly require fewer operations which is equal to O($(k-1)(N-L)$). This regression repeats (k-1) times until all users are distributed over the empty frames. Therefore, this function will require operations $\ll$ O((k-1)(kN)) which can be approximated by O(k^2N). As a result, the worst case complexity of the algorithm is O(kNlog(N)) + O(k^2N).

3.5 Numerical Results

In this section, we present the simulation results of the proposed optimal and heuristic versions of the RT-based MT-Max/Min scheduler in comparison with its conventional version to investigate the trade-off between performance and complexity when implementing our proposed predictive scheduling approach. For simplicity, we assume that all users' buffers are saturated (*i.e.*, there are always packets in the queue ready to be sent) and all the sent packets will be received correctly by the receiver side. Only one frequency resource is available ($r = 1$), with a bandwidth of 180kHz, and time shared over 10 time-slots ($L = 10$) of a 10msec frame. This means that 10 resources are available in each frame. To evaluate the network behavior under high traffic load where the competition on available resources is high, we select the number of access users N= $\{40, 80, 120, 160, 200, 240\}$. N= 10 and 20 are also taken into account for a reason that will be discussed later in the next paragraph. In our system, all users are assumed to be initially distributed uniformly within a circular cell with a radius of 500m and moving with an average speed of 50kph. All users want to send their data to the BS that is located at the center of the cell. The adopted channel model is similar to the one used in [67] which accounts for path loss, log-normal shadowing and Rayleigh fading. The simulation is executed for 5000 frames. It is essential to indicate that increasing the simulation time (measured in frames) had a marginal effect on the obtained results, however, significantly increased the computation time especially in the case of the optimal scheduler.

The throughput performance of the optimal and heuristic implementations for the proposed predictive scheduler versus its conventional version is depicted in Fig. 3.3. All curves have a similar rising trend when increasing the number of users. This is expected given the multi-user diversity gain. More specifically, increasing the number of users in a cell will increase the probability that the scheduler will find candidates experiencing good

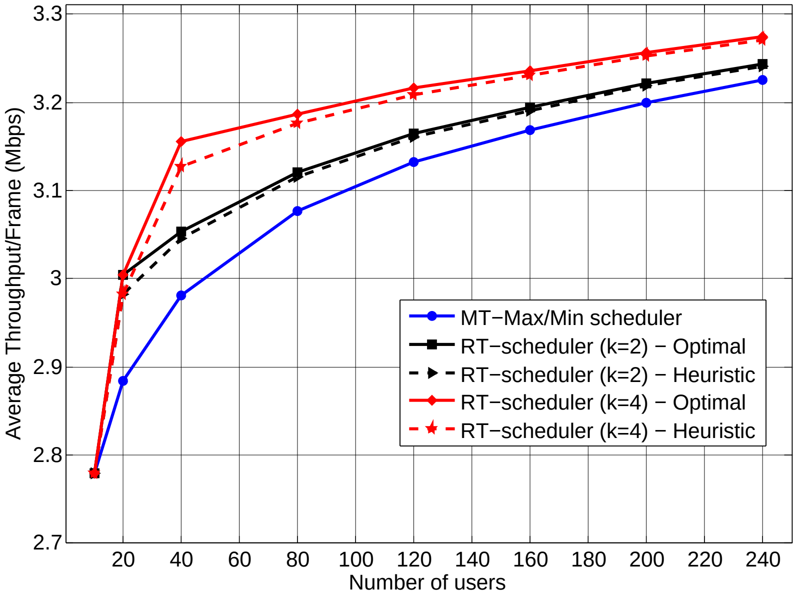


Figure 3.3: Throughput performance comparison

channel conditions at a given time and frequency. In the meantime, the increase in the throughput is bounded by the system's capacity. The throughput improvement noticed in Fig. 3.3 for the optimal and heuristic RT-based schedulers is obvious as their awareness degree k increases from 2 to 4. It should be emphasized that this improvement is achieved while the system is highly congested (*i.e.*, for N between 40 and 240). The heuristic scheduler was also consistent with its optimal counterpart. However, in our assumed scenario (*i.e.*, r= 1 and k= 2 and 4), decreasing the number of users might lead to different observations. For instance, considering the scenarios where N= 10 and 20, the throughput performance shown in Fig. 3.3 was noticed to be as follows: When N= 10, the throughput performance for both RT (*i.e.*, k= 2 and 4) and conventional schedulers are exactly the same because the 10 users will be scheduled in only one frame. It follows that knowing future frames of the channel will not add any new information to the scheduler. However, when N= 20, k= 4 RT-scheduler will not improve the system's throughput performance compared to k= 2 scheduler due to the fact that the scheduler will take advantage of knowing only the first and second frames of the channel, just as k= 2. The remaining third and fourth frames (which belong to the next transmission cycle) will therefore be of no use.

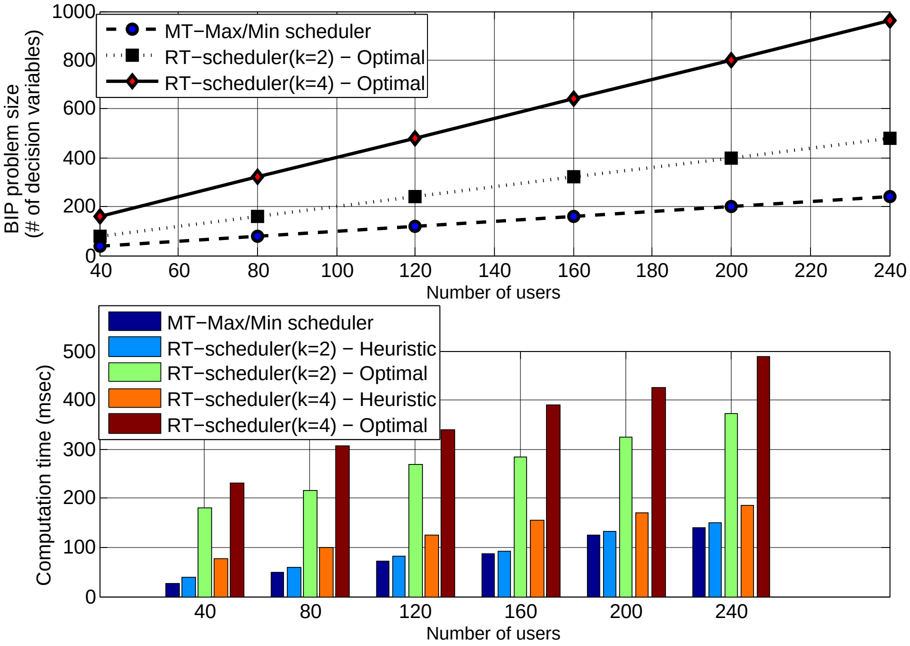


Figure 3.4: Computational complexity analysis

In sum, the results obtained in Fig. 3.3 confirm that the RT-scheduler either improves the system's throughput performance or keeps it as it is, depending on the situation. Moreover, the heuristic scheduler performs as well as the optimal one.

Fig. 3.4 shows the computational complexity of implementing the proposed RT predictive scheduler in its optimal and heuristic forms when compared to the conventional one. In the case of the optimal scheduler, we considered the number of decision variables (*i.e.*, $N \times r \times k$ and $N \times r$ in cases of RT and conventional schedulers, respectively) and the computation time required to solve the first BIP problem of each transmission cycle as the metrics for comparing the complexity of the proposed scheduler with the conventional one. This is due to the fact that the first BIP problem is the most complex problem solved by the optimal RT-based scheduler. However, only the computation time is used to compare the heuristic scheduler with the optimal and conventional ones. The problem size difference between the two schedulers when r= 1 is depicted in the top plot of Fig. 3.4. The computation time is then calculated using MATLAB Profiler on Intel Xeon CPU W3670 with 6 core processors running at 3.2 GHz and 16-GB RAM. The bottom plot results of Fig. 3.4 clearly show a comparable computational cost for the heuristic scheduler compared to the conventional scheduler and significant speed up compared to the optimal one. Hence, the modest throughput improvement (*i.e.*, approximately 5%) obtained in Fig. 3.3 for the proposed heuristic scheduler justifies its effectiveness compared to the conventional MT-scheduler. It is worth noting that the ability of our proposed schedulers to improve the system's throughput is highly dependent on how fast the channel changes in time (*i.e.*, coherence time) and the chosen value of k. Thus, utilizing faster fading channel model than the one used in our simulations and using large values of k is highly expected to positively affect the results of Fig. 3.3 in the favor of our proposed schedulers. This investigation is carried out in the next chapter simulations.

3.6 Chapter Summary

In this chapter, we proposed a framework for implementing a Throughput-efficient predictive scheduling approach which utilizes a long-term ray tracing channel prediction method. First, we conducted a detailed analysis for studying the effect of changing the adaptation horizon on the channel outage probability which limits the system from achieving high reliable data rates. Second, to avoid the high outage observed in the first part, we proposed a predictive scheduler scheme which profits from this analysis by acquiring periodic information about the channel from a fast ray tracing engine. The proposed heuristic RT-assisted MT-Max/Min scheduler showed a comparable throughput performance to its optimal version, which already outperforms the conventional scheduler, with considerably lower complexity.

The extended version of the proposed scheduler which better addresses existing wireless systems (*e.g.*, LTE, 5G)[68] is studied in the next chapter. Finally, while the scheduler is employed at the BS which is capable of delivering high computational power, the hardware implementation challenges will be addressed in the future in a sequel to this work.

Chapter 4

QoS-Aware Energy-Efficient Downlink Predictive Scheduling for OFDMA-Based Cellular Devices

4.1 Introduction

A rapid growth of cellular system designs and standards in the last 10 years has significantly enlarged the wireless market volume. Today's statistics show that over 1 billion users worldwide are connected to the social networking media like Facebook and YouTube [70], and that approximately 40% of them are mobile users [71]. However, analysts predict that these numbers will continue to grow dramatically over the coming years [70] due to two major factors. The first is the unabated advancements in the mobile devices industry, particularly with smart cell phones. Their high computing capabilities allow them to replace many other important devices such as GPS receivers, cameras, and laptop computers. The other factor is the increasing popularity of multimedia services such as VoIP, video streaming, social networking, interactive gaming, web browsing, etc.

From a technical point of view, the emergence of the aforementioned services over the currently deployed 4G networks (*i.e.*, long-term evolution (LTE)) has introduced various challenges for the system design from both the network and user equipment (UE) sides. From the network side, most operators seek to maximize capacity (*i.e.*, spectral efficiency) and reduce the operation cost including the energy efficiency. These goals present

---

A version of this chapter has been published in [69].

challenges, especially in situations of potentially increasing numbers of users and heavy load traffic connections, while having to maintain stringent QoS requirements. Whereas from the UE side, the intensive and complex circuitry of a 4G device is quite rigorous on the current smart cell phone battery technology. This results either in a fast depletion of the battery energy, or it may limit the implementation of a fully functional 4G device. Therefore, a main stream of research has recently been established and devoted for enabling green communications (*i.e.*, Energy-efficient or Energy-aware communication systems) [72]. Moreover, the future generation of mobile communications, known as 5G [5], will address the *energy efficiency* (EE) as a fundamental aspect of the system.

4.1.1 Related Work

An energy efficient design for wireless systems should encompass both the network and the UE sides. Although the majority of the system's energy consumption resides in the network side [73], most recent studies were focusing on optimizing the UE energy consumption either in the uplink [1, 59, 60] or in the downlink [74, 75, 76, 77, 78]. This is due to the need for increasing the UE's battery lifetime per charge. Consequently, in this chapter, we focus on minimizing the UE's energy consumption in the downlink, a subject less studied in the literature.

Gupta *et al.* [74] showed that optimizing the UE power consumption inherently requires optimization of the base-station (BS) downlink transmit power. Hence, the optimization formulation was designed to improve the EE for both of the BS and UE. The idea was based on buffering BS downlink traffic for some transmission time intervals (TTIs) and then transmitting this data in the minimum possible number of time slots constrained by a fixed bit rate constraint. However, the implemented heuristic didn't consistently fulfill the data rate constraint. Unlike Gupta's approach, Xiong *et al.* [75] considered the EE only from the base station side. The authors' objective was to design an optimal energy efficient resource allocation scheme with delay provisioning for delay-sensitive traffic in downlink OFDMA based wireless access networks. The authors' model of the scheduling problem used the effective capacity concept to provide the statistical delay provisioning. Thus, the problem was modeled as maximizing the effective capacity-based EE under statistical delay constraints. Utilizing the effective capacity method, like the model in [75], Tang *et al.* [76] proposed an adaptive resource allocation scheme for downlink heterogeneous mobile wireless networks. The scheme dynamically assigns power-levels and time-slots, and derives the admission control conditions for different real-time mobile users to satisfy various statistical delay-bound QoS requirements. The authors took into consideration the channel-state information (CSI), which was estimated at the receiver and sent back to the transmitter, for adaptive modulation and adaptive power-control. In a different context, Wang *et al.* [77] considered the problem of improving the EE in the downlink of an OFDM-based cognitive radio (CR) network. The objective was to design an energy efficient resource allocation scheme which maximizes the overall EE of the CR system while considering proportional fairness and rate requirements among the secondary users (SUs). This is in addition to keeping the interference to the primary users (PUs) below their tolerable thresholds.

Unlike the studies mentioned above, Chu *et al.* [78] has proposed a green resource allocation (GRA) scheme as an alternative approach to the well known 3GPP LTE discontinuous reception (DRX) power management scheme [79]. To minimize the UE energy consumption in the downlink, the authors optimized the scheduling of the BS downlink transmissions to the UE in a fewer time slots while turning off the receiver circuit in the unused slots. The scheduling was formulated as a nonlinear integer programming problem. It is worth noting that, in contrast to this chapter which focuses on the energy efficiency problem, the previous chapter in the same context of the predictive scheduling has focused on maximizing the network's average throughput (*i.e.*, spectral efficiency) subject to fairness constraints in TDMA-based systems.

4.1.2 Scope and Contribution

In this work, we consider the LTE frequency division duplexing (FDD) mode systems with a frame structure type 1, where two time slots make one subframe (*i.e.*, of duration 1msec) [11]. Combining 10 subframes (*i.e.*, used for scheduling) makes one frame with a length of 10msec. We noted that the work in [78], like many studies in both downlink and uplink, depends on scheduling time granularity of one subframe or at most one frame. In this work, we further expand the solution space of the scheduling problem for optimizing the UE's

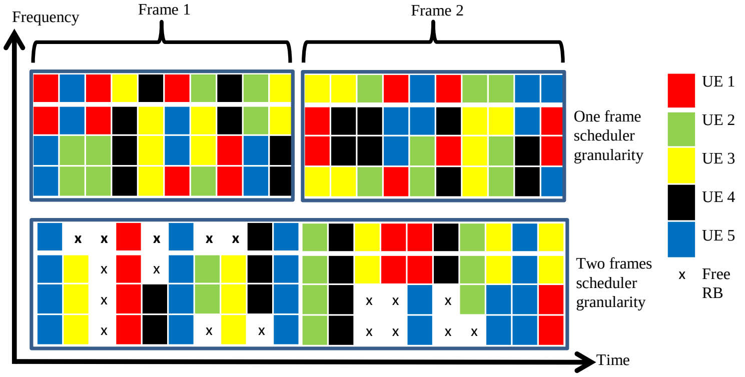

Frequency
Frame 1
Frame 2
One frame scheduler granularity
Two frames scheduler granularity
Time
UE 1
UE 2
UE 3
UE 4
UE 5
x Free RB

Figure 4.1: Energy efficiency for downlink predictive scheduling

energy consumption in the downlink while maintaining users' QoS requirements. The key strategy, as used in [78], is minimizing the number of wake-up TTIs for the UE's receiver circuit, but in a longer time scale spanning multiple future frames (*i.e.*, 10msec LTE radio frames) of the UE's channel.

The problem's time expansion is supported by pre-estimating the users' propagation channel over multiple future frames. This is done using an advanced ray tracing (RT)-based central downlink scheduler system implemented at the BS site. The direct result of increasing the knowledge about the user's channel state information (CSI) is that the scheduler's capability of increasing the UE's EE and meeting the QoS requirements becomes higher compared to previously implemented schedulers. In other words, the UE opportunistically consumes less energy for the same amount of data received based on the future statistics of the propagation channel. This idea is illustrated in Fig. 4.1. It shows that the predictive scheduler with two frames of time granularity would be able to rearrange downlink transmissions to UEs into fewer TTIs compared to that of the traditional single frame scheduler. This results in better EE and possibly offloading spectral resources that help in admitting more UEs. However, the underlying increase in the solution space of the optimization problem results in a substantial growth of the computational complexity. We then address this complexity by designing a less complex heuristic algorithm which approximates the optimal scheduler performance. More details about the optimization problem and its relaxation is provided in Sections 4.4 and 4.5.

The contributions of this work are summarized as follows:

- We propose an optimal framework which minimizes the energy consumption of the UE receiver circuit while satisfying a constant rate (*i.e.*, effective bandwidth) constraint. The framework utilizes the ray tracing channel prediction model, and it considers both the modulation and coding scheme (MCS) and UE circuit operation time.
- To assure feasible solutions, we propose a second formulation for the optimization problem through relaxing the rate constraint using the penalty method to cope with the channel capacity limitations.
- After investigating the dominant factors which affects the UE's power consumption budget in the downlink, we further modify the optimization problem by allowing the scheduler to focus solely on optimizing the number of wake-up TTIs for the UE.
- In order to address the complexity of the optimization problem, we deduce a heuristic algorithm to solve the scheduling problem in the final formulation with a comparable performance but significantly lower complexity.

The rest of this chapter is organized as follows: Section 4.2 presents the system model and design objectives. The motivation for utilizing the ray tracing channel prediction model with the proposed RT-based scheduler system is discussed in Section 4.3. The optimal formulation and the iterative algorithm of the proposed scheduler are described in Sections 4.4 and 4.5, respectively. Simulation results are provided in Section 4.6. Finally, Section 4.7 summarizes the chapter.

4.2 System Description

4.2.1 System Model

In this work, we consider a single cell of a mobile cloud computing (MCC) LTE downlink multiuser system. It is based on the evolving concept of cloud radio access networks (C-RAN) [80]. On-line computational resources can be used for the computationally demanding RT prediction of the eNB-UE channel, as explained below within the MCC framework. The arrangement which allows the transfer of the prediction cost from the eNB to the C-RAN. This is illustrated with the aid of Fig 4.2. A C-RAN architecture is based on centralizing multiple baseband processing units (*i.e.*, traditionally located at every BS site)

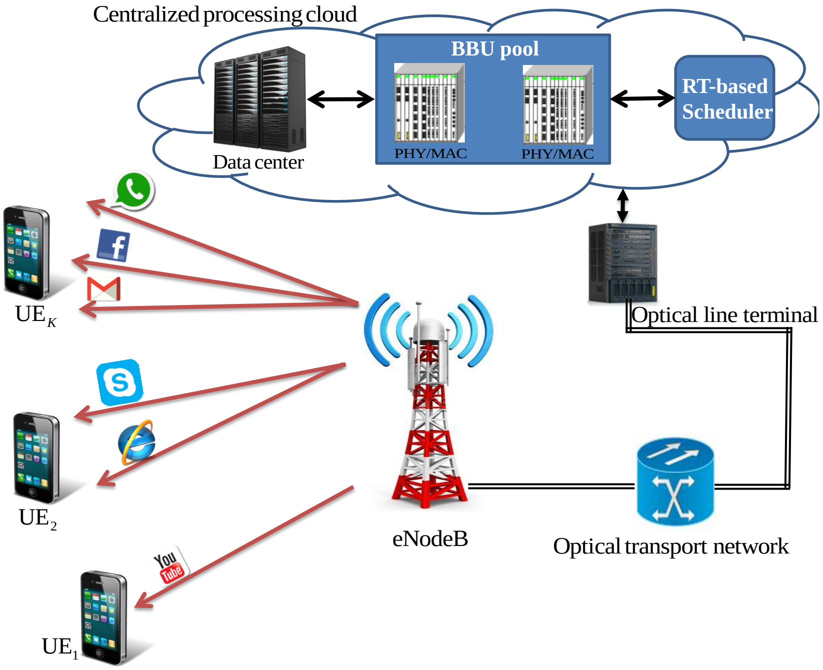


Figure 4.2: C-RAN based model

which forms a pool of shared wireless resources within a centralized processing cloud [80]. In addition to the baseband units (BBU) pool, the cloud also integrates a data center and an RT-based downlink scheduling system. The data center is responsible for establishing users' traffic connections based on standard QoS requirements. The RT-based scheduling system mainly integrates an RT engine (*i.e.*, for predicting the downlink CSI) and a central scheduler. The system's detailed structure and operation will be discussed in Section 4.3. The evolved Node B (*i.e.*, eNB) tower is connected to the central processing cloud via an optical transport network.

We assume a single eNB located at the center of cell. The overall cell bandwidth is divided equally into N 180kHz resource blocks (RBs) consisting of 12 adjacent subcarriers. The FDD LTE frame type 1 duration is 10msec and is composed of 10 1msec subframes (*i.e.*, each subframe represents a TTI) [11]. When the normal cyclic prefix is used, each subframe consists of 14 OFDMA symbols, each with a duration of 66.67μsec.

The eNB transmits H traffic connections (*i.e.*, bearers) to each one of a K UEs. As our model addresses the UE's energy efficiency in the downlink, we use $P_t^{(k)}$ to denote the total downlink power consumed by the receiver circuit of UE k. More details about the calculation of the components of $P_t^{(k)}$ are provided in Section 4.2.2. Each user connection is associated with different QoS requirements, depending on the traffic type (*e.g.*, VoIP, video streaming, FTP, etc). However, without loss of generality, we assume that the QoS requirements for all traffic connections (of each UE) are pre-processed by the data center. Then the data center translates them into a single connection request (or reservation) with a target average transmission rate $\bar{R}_D$. That rate is calculated based on the effective bandwidth theory (*i.e.*, the dual concept of the effective capacity [75]) and will simultaneously meet all of the user's individual connections requirements. From the UE side, multiple traffic connections with different QoS parameters are further prioritized (*i.e.*, intra-scheduling) according to their QoS class identifier (QCI) priority (Table 13.1 in [11]). It is known that intra-scheduling user's connections is preceded by UE's inter-scheduling. This two step process is vital especially when the system's capacity prevents the scheduler from allocating enough resources to accommodate the user's target rate (*i.e.*, $\bar{R}_D$).

To satisfy the users' QoS requirements, we assume that the central processing cloud requests data connections between eNB and users with a total time duration of T and a target average bit rate for each UE of $\bar{R}_D^{(k)}$. Starting from this assumption, the eNB ultimately aims to schedule the transmission of the data for each user's requested connection in a way that satisfies two major goals. The first is to maintain the average connection's downlink rate for each UE by adequate allocation of resources. That rate should be preserved throughout the requested connection duration T at the data center's designated value $\bar{R}_D^{(k)}$. The second goal is to minimize the energy consumed by the UE's hardware to receive and decode the eNB's downlink traffic. The key strategy behind reducing the UE receiver's energy consumption is minimizing the number of TTIs where the UE receiver circuit is scheduled to be in active mode. More details about this strategy are provided Sections 4.2.2 and 4.4. Ideally, in an OFDMA-based system, for the eNB to satisfy the user's requested connection rate requirement, the following constraint must hold all the time:

$$\frac{1}{T}\sum_{m=1}^{M}\sum_{n=1}^{N} B_k(m,n) \;\geq\; \bar{R}_D^k,\;\; \forall\, k \tag{4.1}$$

where M is the total number of TTIs within a requested connection of total duration T

such that $T = M\, T_{TTI}$ (*i.e.*, T_{TTI} is equal to 1 msec), $B_k(m,n)$ is the number of received bits by user k during TTI m over RB n, and $\bar{R}_D^{(k)}$ is the data center's requested effective rate for user k within the connection time T. It should be noted that the requested average connection rate $\bar{R}_D^{(k)}$ is selected to accommodate the QoS requirements of multiple traffic buffers for UE k (*i.e.*, $\bar{R}_D^{(k)} = \sum_{h=1}^{H} \bar{R}_D^{(k)}(h)$, where h is the UE's connection index).

The importance of the constraint (4.1) lies in carefully selecting a suitable time horizon (*i.e.*, further explained later in Section 4.4 when defining τ) during which the scheduler successively allocates resources throughout the requested connection time T. This can be illustrated by looking at the following sub-constraint:

$$\frac{1}{\tau} \sum_{m=m_o}^{m_o+G-1} \sum_{n=1}^{N} B_k(m,n) \;\geq\; R_D^{(k)}, \;\; \forall\, k, m_o \tag{4.2}$$

where G is the number of TTIs considered in the time horizon τ (*i.e.*, equal to G x 1 msec) during which the RT engine predicts the channel, $m_o \in \{1,\, 1+G,\, 1+2G,\, ...,\, 1+M-G\}$ is the initial TTI index in the observed horizon within the connection time T, $R_D^{(k)}$ is the quasi-instantaneous target rate of user k within the horizon τ.

The key idea behind the sub-constraint (4.2) is that changing the value of τ provides the network with a two fold control on the UE's energy efficiency and network's QoS requirements (including the packet delay). For instance, we consider a user receiving a delay sensitive VoIP connection with a total duration of 50sec (*i.e.*, T= 50sec) at a standard instantaneous rate of 13.6Kbps (*i.e.*, $R_D^{(k)}$= 13.6Kbps). The 13.6Kbps corresponds to generating a voice packet of 244 bits every 20msec plus an extra 28 bits for the compressed IP/UDP/RTP header (as discussed in [81]). Satisfying the 13.6Kbps for T= 50sec with the aid of (4.2) having τ set to any value less or equal to 100msec (*i.e.*, 10 LTE frames) will ensure a packet delay bounded by 100msec (*i.e.*, VoIP packet delay budget) throughout the 50sec call time. In this case, both of the connection delay and rate requirements are met. In addition to meeting the QoS requirements, utilizing accurate future predictions of the user's CSI by an RT-based mechanism (*i.e.*, explained in Section 4.3) in the 100msec horizon better optimizes the UE's EE compared to traditional shorter term scheduling.

A major trade-off could be seen when selecting the value of τ. Shortening the scheduling time horizon to a small value enhances the scheduler's packet delay performance, yet it will exploit short-term channel statistics for optimizing the UE's energy consumption. Thus, τ is left to be designed based on the target application QoS requirements. For guaranteed bit rate (GBR) connections (*e.g.*, VoIP, video), increasing τ within the range of the packet expiration time increases the energy efficiency and the average packet delay as well. For Non-GBR (NGBR) connections (*e.g.*, buffered video and web browsing), increasing τ arbitrarily to any value (less than or equal T) will increase the energy efficiency. However, this will be associated with a substantial computational growth because of the enlarged optimization problem and even creates challenges from the RT engine side in predicting the CSI for prolonged time intervals. In sum, setting τ to a proper value, allows the designer to achieve an optimal trade-off between the bit rate, delay and energy efficiency for the radio link.

It should be stressed that satisfying the constraint in (4.1) instantaneously with the aid of (4.2) at the minimum level of energy consumption, might not be guaranteed all of the time for all users. Two major factors take place: first, is the system's limited capacity (*i.e.*, frequency resources); second, is the continuous variations in time and frequency domains for the user's channel conditions. Moreover, in this work, increasing the transmit power will not help to guarantee the satisfaction of (4.2). This is supported by the assumption that the eNB (*i.e.*, the transmitter in our analysis) is operating at the maximum allowable transmit power. This assumption is based on the fact that our work is considering a single cell scenario and thus, no inter-cell interference constraints exist. Besides, our main objective in this work is to only optimize the UE's energy efficiency in the receive mode while disregarding that from the eNB side. A potential solution and relaxation for this hard constrained problem (*i.e.*, sometimes unfeasible) will be later provided in Section 4.4.

In the same context, it is also worth noting that the scheduler is always protected by an admission control system which helps the scheduler avoid admitting users' connections over reaching the network's capacity. Thus, after adapting the value of τ accordingly, the eNB's scheduler can safely utilize (4.2) to support both admitted GBR and NGBR connections.

In this work, and to simplify the analysis, we only investigate improving the UE's EE constrained by the GBR requirement (*i.e.*, effective bandwidth) on an instantaneous basis.

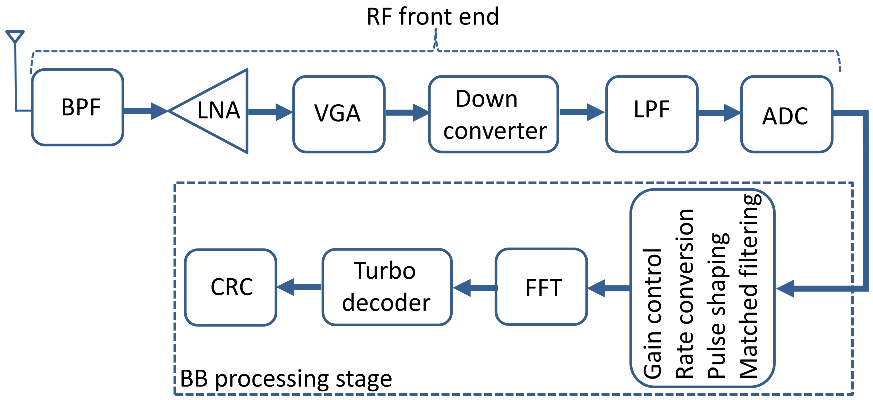


Figure 4.3: Simplified block diagram for LTE UE downlink physical layer processing chain

This framework is supported by selecting the scheduler's granularity (*i.e.*, τ) less than or equal to 100msec. Thus, the delay analysis is not considered in this work.

4.2.2 UE Circuit Power Consumption

The UE transceiver circuit could be seen as a composition of basedband (BB) and radio frequency (RF) stages. A simplified block diagram for those stages can be seen in Fig. 4.3. The components of those stages are the major source of energy consumption inside any cellular device. In order to investigate the EE of our scheduling scheme, the LTE UE power consumption model developed in [82] is utilized in our analysis to measure the UE energy consumption while in the receive mode (*i.e.*, downlink). The model originally accounts for the power consumption of both the transmit and receive processing paths. However, in this work we only consider the power model of the receiver section. This model was defined as follows [82]:

$$P_t^{(k)} = \underbrace{m_{idle} \underbrace{P_{idle}}_{\text{constant}}}_{\text{UE OFF}} + \underbrace{\overline{m_{idle}}(\underbrace{P_{on} + P_{rx}}_{\text{constant}} + \underbrace{P_{BB} + P_{RF}}_{\text{variable}})}_{\text{UE ON}} \quad \text{watts} \tag{4.3}$$

where $P_t^{(k)}$ is the total power consumption of the UE's k receiver circuit, m_{idle} is a logical variable which determines whether the UE is OFF (*i.e.*, idle state) or ON (*i.e.*, wake-up state), P_{idle} is the power consumed when the UE is in the idle state and is equal to 0.5w,

P_{on} is the power consumed when the UE is awake from the idle state and is equal to 1.53w, P_{rx} is the base power consumed by the receiver circuit while in the operation state and is equal to 0.42w, P_{BB} is the power consumed by the baseband stage of the receiver circuit, and P_{RF} is the same for the RF stage.

More details about the power components in (4.3) can be found in [82]. For simplicity, and for the rest of the chapter, we set P_c to denote the constant power term in (4.3) that is either equal to P_{idle} (when the UE is OFF) or $P_{on} + P_{rx}$ (when the UE is ON), while P_k denotes the variable power consumed by UE k when UE is ON and is equal to $P_{BB} + P_{RF}$. In [82], both P_{BB} and P_{RF} are modeled by fitting a first order polynomial to experimental circuit power measurements employing the least mean square error criteria as follows:

$$P_{BB} = 1.923 + (2.89 \times 10^{-3} \times B_r) \text{ watts} \tag{4.4}$$

$$P_{RF} = 1.889 - (1.11 \times 10^{-3} \times P_r) \text{ watts} \tag{4.5}$$

where B_r is the downlink bit rate in Mbit/sec, and P_r is the received signal power in dBm.

4.2.3 Channel Model

The multipath fading downlink channel between the eNB and each UE is modeled using the deterministic RT approach [33] with the aid of the RT engine residing in the central processing cloud of Fig. 4.2. The question of how often (or for how long) the channel is modeled during the user's connection is going to be answered in the next section. The received signal power at the UE side is calculated by squaring the vector summation of all complex polarized electric fields components arriving at the UE antenna. Each polarized field vector, which differs in magnitude and phase, corresponds to a separate received radio ray scattered from objects in the surrounding environment such as buildings, trees and ground. The total received signal power in the far zone of the transmitting antenna is, therefore, as described in Chapter 2 of [34]:

$$P_r^{(k)} = \frac{\lambda^2 \beta}{8\,\pi\,\eta_o} \left| \sum_{i=1}^{I} \left[\begin{array}{c} E_{\theta,i}^{k} \sqrt{\left|G_{\theta}^{k}(\theta_i,\phi_i)\right|}\, e^{j\psi_\theta} \\ +\, E_{\phi,i}^{k} \sqrt{\left|G_{\phi}^{k}(\theta_i,\phi_i)\right|}\, e^{j\psi_\phi} \end{array} \right] \right|^2 \tag{4.6}$$

where $P_r^{(k)}$ the total received signal power by the antenna of UE k, η_o is the free space wave impedance, β is the propagation constant, I is the total number of radio paths received by UE k, $E_{\theta,i}^{k}$ and $E_{\phi,i}^{k}$ are the theta and phi components of the electric field associated with the i^{th} radio path received by UE k, $G_{\theta}^{k}(\theta_i,\phi_i)$ and $G_{\phi}^{k}(\theta_i,\phi_i)$ are the theta and phi components of UE k receiver antenna's gain for the i^{th} path with a direction of arrival of θ_i and ϕ_i, ψ_θ and ψ_ϕ are the relative phases of the theta and phi components of the far zone electric field.

Each of the E_θ and E_ϕ electric field components in (4.6) is further resolved into an appropriate pair of polarization components. One component is parallel (*i.e.*, vertically polarized) to the plane of incidence at the reflection (or diffraction) point on an obstacle's surface intercepting the signal path. The other component is perpendicular (*i.e.*, horizontally polarized). Each of the I paths might contain multiple reflection and diffraction points, or even a combination of them, throughout the radio path trip from the eNB antenna to the UE receiver's antenna. More details about the calculation of the polarization components at the reflection and diffraction points can be found in [33]. The RT engine which is capable of tracing the radio signal paths and evaluating their associated fields will be further illustrated in the next section in regards to the proposed downlink scheduling system.

The signal power prediction, provided by the RT engine, is then utilized by the eNB's central scheduler to optimize the users' reception schedule in time (*i.e.*, TTI) and frequency (*i.e.*, RB) in terms of EE while meeting target QoS requirements. That is to say, knowing the received signal power over each RB during all TTIs within a single frame or across multiple frames will allow the scheduler to determine the received block size by each UE. That information then becomes available on the physical downlink shared channel (PDSCH) in every TTI after setting the UE to a specific MCS. The supported MCSs with their corresponding spectral efficiency defined by the LTE physical layer, and targeting a 10% block error rate (BLER), are as shown in Table 4.1 [1]. Based on the received block size and the power consumed by the UE's receiver circuits to receive that block, the scheduler efficiently commands the UE, to turning its circuits ON or OFF during all of the observed TTIs. This control scheme is formulated and explained in detail in Section 4.4.

Table 4.1: List of MCS Indices [1]

Index s	Modulation	Coding Rate	Spectral Efficiency ζ	Effective SNR (dB) γ_k
0	—	—	0 bits	> -6.7536
1	QPSK	78/1024	0.15237	$-6.7536 : -4.9620$
2	QPSK	120/1024	0.2344	$-4.9620 : -2.9601$
3	QPSK	193/1024	0.3770	$-2.9601 : -1.0135$
4	QPSK	308/1024	0.6016	$-1.0135 : +0.9638$
5	QPSK	449/1024	0.8770	$+0.9638 : +2.8801$
6	QPSK	602/1024	1.1758	$+2.8801 : +4.9185$
7	16QAM	378/1024	1.4766	$+4.9185 : +6.7005$
8	16QAM	490/1024	1.9141	$+6.7005 : +8.7198$
9	16QAM	616/1024	2.4063	$+8.7198 : +10.515$
10	64QAM	466/1024	2.7305	$+10.515 : +12.450$
11	64QAM	567/1024	3.3223	$+12.450 : +14.348$
12	64QAM	666/1024	3.9023	$+14.348 : +16.074$
13	64QAM	772/1024	4.5234	$+16.074 : +17.877$
14	64QAM	873/1024	5.1152	$+17.877 : +19.968$
15	64QAM	948/1024	5.5547	$> +19.968$

The MCS is decided by the eNB's scheduler based on the effective SNR (γ_k) at the UE side in order to maintain a target BLER performance (*i.e.*, default value is 10% in LTE [10]). Given certain modulation order, the code rate is also selected based on the channel condition (*i.e.*, low code rate is used in poor channel conditions). The selected code rate [14] for an MCS with index s directly affects the spectral efficiency per transmitted symbol (ζ_s) according to the following formula:

$$\zeta_s = \log_2(\mu_s).\, C_s, \tag{4.7}$$

where μ_s and C_s denote the modulation order and the coding rate for an MCS of index s, respectively. If we assume that a user k is assigned to a set of RBs j during TTI m (*i.e.*,

$\mathcal{N}_{j,k}(m)$), then the total received block size, in bits, by user k during TTI m is given by:

$$\Re_k(m) = \lfloor S_{RB} \times \zeta_s \times \mathcal{N}_{j,k}(m) \rfloor \tag{4.8}$$

where S_{RB} denotes the number of data symbols transmitted per single RB (*i.e.*, 14 symbols in the case of normal cyclic prefix), $\lfloor x \rfloor$ denotes the largest integer less than or equal x.

It is also important to highlight that in LTE, the UE is configured to report the channel quality indicator (CQI) feedback over the physical uplink shared channel (PUSCH) [10] to assist the eNB in selecting an appropriate MCS to adopt for the downlink transmissions. However in our model, we assign this task to the RT engine, located in the central processing cloud (as explained in Section 4.2.1), which predicts the downlink channel quality by tracing the radio signal paths to the user's geographical location. The study of how efficient the RT channel prediction replaces the traditional CQI reporting, in terms of offloading frequency resources and reducing the number of UE transmissions (*i.e.*, feedback time), is beyond the scope of this thesis.

4.3 Proposed Predictive Scheduling System

Advancements in today's mobile data services and applications continue to emerge and grow. The main challenge is that this growth is existing in a highly dynamic radio propagation environment. Consequently, the development of faster and more efficient propagation prediction platforms needed for designing optimized wireless networks in terms of spectral and energy efficiency is becoming more critical. In this context, the RT prediction model have provided a promising agile solution with higher accuracy compared to traditional statistical models [25]. Due to its interactive nature which simulates the influence of the surrounding geographical environment on the propagation of radio waves, the RT model has been envisioned to enable real-time applications (*e.g.*, vehicle-to-vehicle (V2V) communication) which usually experience fast and dramatic change in the channel impulse response.

In this section, we visualize the ray tracing technique (*i.e.*, discussed in Chapter 2) as being a core part of an integrated cellular eNB platform that could be utilized either in

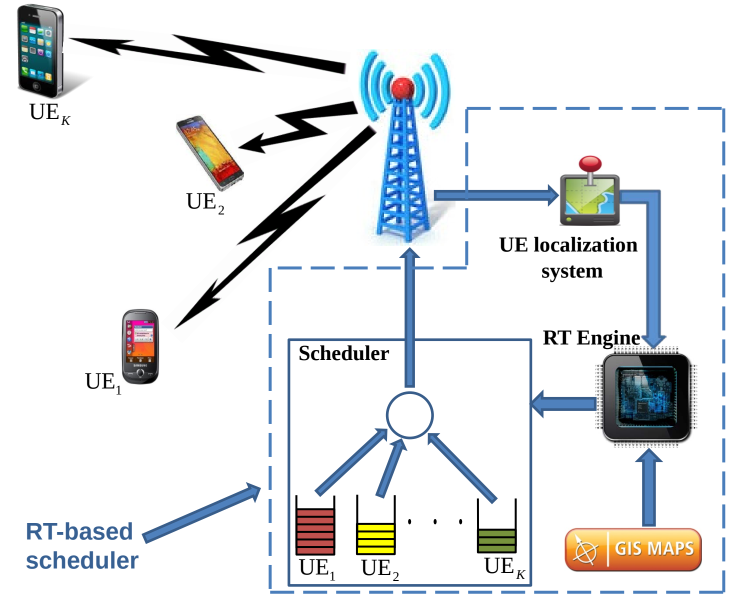


Figure 4.4: Ray tracing based downlink scheduler system

the current 4G or tomorrow's 5G networks. This platform is depicted in Fig. 4.4. This promising platform is motivated by two important factors. As illustrated earlier, the first factor is the intensive research conducted (and still active) in the area of accelerating the ray tracing method for efficient radio propagation modelling [37, 38]. These efforts have produced numerous efficient acceleration techniques that make the implementation of ray tracers an attractive solution for modelling wireless channels. The second factor is the fast and continuing evolution of today's high performance computing (HPC) platforms such as field programmable gate arrays (FPGAs), graphics processing units (GPUs) and advanced digital signal processors (ADSPs). These platforms have offered powerful solutions to build high speed ray tracing engines [83].

The predictive downlink scheduler system shown in Fig. 4.4 depends mainly on the channel's future information provided by the RT engine. The engine is designated for predicting the CSI for all UEs connected to the eNB for longer time intervals compared to traditional sounding of pilot signals [84] that provide short-term measurements. The long-term channel prediction for mobile users is assisted by an accurate localization system and GIS maps database. The function of the localization system is to interactively determine the geographical location of each UE within the cell's coverage GIS map. This ensures that the RT engine calculates an accurate channel SNR based on a real location of the UE within the cell's propagation environment. Whether the localization strategy is UE-based, UE-assisted or network-based [85], we consider the fact that eNB is capable of acquiring the geographical information of the UE along its trip route for a certain time interval in a periodic manner. This information is then utilized by the RT engine to predict the UE's CSI along its registered route (or route section). Hence, smart and seamless integration between the UE localization system and the GIS map with the RT engine is fundamental for building our predictive scheduling system.

To further elicit the relation between Fig. 4.4 and Fig. 4.2, it should be noted that the RT-based scheduler block highlighted in Fig. 4.4 represents the detailed structure of the RT-based scheduler located inside the centralized processing cloud of Fig. 4.2. Thus, the RT engine which predicts the CSI for each UE is a part of the shared architecture explained in Fig. 4.2. However, just like the BBU pool of the C-RAN model in Fig. 4.2, a pool of RT processors will also be available within the cloud to be efficiently shared between different cell towers. This way, there will be no need for a dedicated RT engine at each cell tower in the large-scale network.

In light of the previous discussion, the eNB is capable of pre-estimating the mobile user's propagation channel for complete sections of their trip routes during their access periods. Such long-term UEs channel estimation enables the eNB's central scheduler to optimally allocate RBs for the downlink traffic of each user more efficiently in terms of energy consumption. The key advantage here is that the scheduler has a better long-term information about the channel capacity for each user compared to conventional channel-aware schedulers with less CSI information. Consequently, the proposed RT-assisted scheduler becomes capable of scheduling users on the available RBs in fewer TTIs with a greater degree of freedom compared to traditional energy/channel-aware schedulers [78]. This enables a more energy efficient operation for the UE's receiver circuit in the downlink, while maintaining target QoS levels. However, this comes at the expense of computational complexity, discussed in Section 4.6. It also should be noted that the channel prediction time horizon (*i.e.*, scheduler's granularity) we consider here is a few multiples of the LTE radio frame (*i.e.*, 10 msec duration). Since this time horizon is as short as a fraction of a second, the predicted channel conditions by the RT engine will stay almost invariant. However, predicting the CSI within this time horizon notably affects the energy consumption of the UE as will be highlighted later in Section 4.6.

It is important to highlight that one of the major challenges facing the C-RAN architecture which might affect the decision accuracy of the proposed predictive scheduler system shown in Fig. 4.4 is the front-haul latency. The optical link between the eNB tower and the BBU, shown in Fig. 4.2 (known as the front-haul) introduces a transport network latency that did not originally exist in traditional RAN architecture which has both the BBU and the radio tower co-located. Such latency is due to three major sources which are transmission, queuing and processing of data. Therefore, to maintain the scheduling decision accuracy within one LTE frame of 10msec duration, given that light travels approximately 1km in 5μsec through fiber, the maximum fiber distance allowed between the BBU and eNB tower should be less than 1000km (typically less than or equal 20km [80]) in order to have a round trip transmission delay less than 10msec. Moreover, various promising solutions (*e.g.*, compression techniques, single fiber bi-direction and wave-length division multiplexing) have been addressed in [80] to reduce the traffic volume over the fiber links, and hence, the queuing delay. In addition, optimizing the front-haul queuing delay and its impact on the information flows has been recently addressed in [86]. Finally, the field trials carried out in [80] have showed that the processing delay could be practically less than 1μsec.

4.4 Optimal Scheduler

4.4.1 General Formulation

In this section, the optimal scheduling problem is formulated. The problem's objective as mentioned earlier is to minimize the UE's receiver energy consumption while maintaining the quasi-instantaneous rate for multiple connections per user terminal at a target value. As explained in Section 4.2, the quasi-instantaneous rate constraint for each user connection, that has been designated by the system's central cloud, exclusively accounts for its QoS requirement(s). The problem constraints are divided into three sets as follows:

1. GBR constraint: each connection for each user must accomplish fixed quasi instantaneous transmission rate (*i.e.*, effective bandwidth) throughout the requested connection duration T in time steps of τ (*i.e.*, RT prediction range).
2. Interference constraint: in order to avoid intra-cell interference between users, a single RB must be exclusively allocated to a single user every TTI.
3. UE's circuits operation time constraint: in order to optimize the overall energy consumption for each UE, the scheduler is devoted to finding the optimal allocations, with respect to the energy consumed, in the minimum possible number of TTIs. This will ensure minimal base power consumption (*i.e.*, $P_{on} + P_{rx}$) for the user's receiver circuit.

The optimal energy allocation is obtained by solving the following constrained sum-utility minimization:

Min

$$E_{tot} = T_s \sum_{k=1}^{K} w_k \sum_{h=1}^{H} \sum_{m=m_o}^{m_o+G-1} \left(\begin{array}{l} P_k(m, \mathcal{N}_{j,k}(m))\, \Psi^h_{\mathcal{N}_{j,k}(m)}(m) \\ +P_c\, \Phi_k(m) \end{array} \right) \tag{4.9a}$$

Subject to

$$\sum_{m=m_o}^{m_o+G-1} B^h_k(m, \mathcal{N}_{j,k}(m))\, \Psi^h_{\mathcal{N}_{j,k}(m)}(m) \geq \tau\ R^{(k)}_D(h),\ \ \forall k, h \tag{4.9b}$$

$$\bigcap\nolimits_{k=1}^{K} \Psi^h_{\mathcal{N}_{j,k}(m)}(m) = \phi,\ \ \forall m, h \tag{4.9c}$$

$$\Psi^h_{\mathcal{N}_{j,k}(m)}(m) - \Phi_k(m) \leq 0\ , \forall k, m, h \tag{4.9d}$$

where E_{tot} is the total energy consumed for all users over an observation period of G TTIs, w_k is a weighting factor for UE k, T_s is the TTI duration (*i.e.*, 1 msec), $P_k(m, \mathcal{N}_{j,k}(m))$ is the total power consumption of the baseband and RF receiver circuits for UE k during TTI m over the set of RBs $\mathcal{N}_{j,k}(m)$, $\mathcal{N}_{j,k}(m)$ is the set of RBs j assigned to UE k during TTI m, $\Psi^h_{\mathcal{N}_{j,k}(m)}(m)$ is a binary decision variable which indicates whether the set of RBs j is allocated to connection h of UE k during TTI m or not, P_c is the UE's receiver constant power consumption during each TTI which depends on the UE's operation state (*i.e.*, P_{idle}

for OFF state and $P_{on} + P_{rx}$ for ON state), $\Phi_k(m)$ is a binary indicator which determines whether UE k receiver circuit is in an active state during TTI m or not, $B_k^h(m, \mathcal{N}_{j,k}(m))$ is the number of scheduled transmitted bits in the downlink for connection h of user k over RBs set j during TTI m, τ is the scheduler's time granularity (or time step) that is related to the RT engine prediction range over G TTIs, and $R_D^{(k)}(h)$ is the data center's equivalent instantaneous transmission rate for connection h of user k.

The first decision variable $\Psi^h_{\mathcal{N}_{j,k}(m)}(m)$ in the cost function of (4.9a) allows the scheduler to optimally adjust the MCS of the RBs set allocated to each user in each TTI in order to minimize the UE's baseband and RF circuits energy consumption (as explained in Section 4.2.2). The second variable $\Phi_k(m)$ is devoted to minimizing the number of scheduled wake-up TTIs for each UE to receive its designated data bits. As could be inferred from (4.9a), this is achieved through penalizing the cost function by the UE's constant power consumption value P_c for each scheduled (*i.e.*, wake-up) TTI (per each user) irrespective of the number of RBs allocated within each TTI.

The constraint defined in (4.9b) resembles the quasi-instantaneous rate constraint for each user connection that is assumed to exclusively satisfy its QoS requirements as described in Section 4.2. The second constraint in (4.9c) ensures that each user is assigned to a unique set of RBs (*i.e.*, not intersecting with other users' sets) during each TTI, and hence, avoid intra-cell interference between users. For a TTI m having N available RBs, the number of possible RB sets with sizes of 1, 2, ..., N, from which the scheduler searches for each user, is equal to $\sum\limits_{q=1}^{N-1} {}^N c_q + 1$. The constraint in (4.9d) is the well known if-then constraint that is designed to ensure that the binary variable $\Phi_k(m)$ penalizes the cost function by P_c if a user is assigned to any set of RBs within TTI m.

4.4.2 Penalty Method-Based Formulation

In practice, the scheduler may not able to satisfy the rate constraint in (4.9b) for all users all the time, especially in situations of deep fading (or outage) channel conditions. In other words, as a result of the time varying nature (*i.e.*, due to the multipath fading) of the users' channel, which temporarily limits its capacity to accommodate their rate constraints, the optimization problem in (4.9) could be unfeasible in different time intervals. Since we do not consider admission control procedures in this work to handle the issue of unbalanced demand versus available resources within the duration of an admitted connection, the penalty method [87] is utilized to ensure feasible solutions for the optimization problem in (4.9) by relaxing the constraint (4.9b).

The penalty method is known to approximate the solution of constrained optimization problem by iteratively solving a series of dependent unconstrained problems whose solutions ideally converge to the original constrained problem. The dependency implies that the solution of each unconstrained problem in each iteration affects the following one. More specifically, each unconstrained problem adds a penalty term, also known as penalty function, to the objective function of the following problem in a successive manner till the penalty function converges to zero. The penalty function value represents how far the current solution is from that of the original constrained problem. In our problem, the penalty method is used to relax the constraint of (4.9b) by reversing the inequality sign and adding a new term in the objective function. The new term role is to push the new unconstrained formulation to converge to the solution of the original constrained formulation as much as possible. The new unconstrained formulation can be illustrated as follows:

Min

$$\begin{aligned} Z_1 \;=\; & E_{tot} + \\ & \sum_{k=1}^{K}\sum_{h=1}^{H}\alpha_{k,h}\left(\begin{array}{c}\left(\tau R_D^{(k)}(h)+\ell(k,h)|_{m_o-G}^{m_o-1}\right)\Omega_k \;\; - \\ \sum\limits_{m=m_o}^{m_o+G-1} B_k^h(m,\mathcal{N}_{j,k}(m))\,\Psi^h_{\mathcal{N}_{j,k}(m)}(m)\end{array}\right) \end{aligned} \tag{4.10a}$$

Subject to

$$\sum_{m=m_o}^{m_o+G-1} B_k^h(m,\mathcal{N}_{j,k}(m))\,\Psi^h_{\mathcal{N}_{j,k}(m)}(m) \leq \;\tau\, R_D^{(k)}(h)+\ell(k,h)|_{m_o-G}^{m_o-1} \;\;,\quad \forall\, k,h \tag{4.10b}$$

$$(4.9c),(4.9d) \tag{4.10c}$$

$$\Omega_k \;>\; 0\,,\;\forall\, k \tag{4.10d}$$

where Z_1 is the new unconstrained objective function, E_{tot} is the original constrained objective function that was given in (4.9a), $\alpha_{k,h}$ is a QoS optimization weighting factor for prioritizing traffic connection h of user k, $\ell(k,h)|_{m_o-G}^{m_o-1}$ is the left over unscheduled bits from the previous G TTIs, and Ω_k is a new binary decision variable accounting for the total number of bits that user k requires to receive within the current G TTIs.

It is obvious in (4.10a) that the penalty function added to the constrained problem presented in (4.9) is the whole term added to E_{tot}. The key idea of the new formulation highlighted in (4.10) can be understood as follows: any remaining bits for a certain user connection that has not been transmitted within the previous G TTIs will be accumulated as leftover bits (*i.e.*, $\ell(k,h)|_{m_o-G}^{m_o-1}$) to the original bits (*i.e.*, $\tau R_D^{(k)}(h)$) awaiting transmission in the current G TTIs. After a few iterations (that corresponds to a certain time delay), the leftover bits for the same connection are converged to zero. This is clearly understood from (4.10a) as minimizing the difference for each user k between what is demanded (*i.e.*, $\tau R_D^{(k)}(h) + \ell(k,h)|_{m_o-G}^{m_o-1}$) and what is available and granted (*i.e.*, $\sum_{m=m_o}^{m_o+G-1} B_k^h(m, \mathcal{N}_{j,k}(m))\, \Psi_{\mathcal{N}_{j,k}(m)}^h(m)$) will lead to a close solution of the constrained problem, albeit with a certain amount of delay. In addition, dynamically configuring the weighting factor $\alpha_{k,h}$ for each UE and each connection per UE enables class-based packet scheduling [81]. Thus, continuously prioritizing users (*i.e.*, inter-scheduling) based on a certain fairness criteria, and prioritizing different traffic buffers (*i.e.*, intra-scheduling) for the same user based on their QCI priority index could be easily attained. From another perspective, the ratio of the weighting factors α to w for each user determines the amount of effort the scheduler spends to satisfy user's connection rate constraint (*i.e.*, QoS) to that spent to minimize the amount of energy consumed (*i.e.*, E_{tot}), respectively. Also, it has to be noted from (4.10d) that the decision variable Ω_k is constantly set to 1 to force its term - which resembles the total number of bits awaiting transmission for each user - to always appear in the cost function in (4.10a).

4.4.3 Practical ON-OFF Formulation

Looking back to equations (4.4) and (4.5), we found that changing the scheduled MCS, for any user, within a single TTI (*i.e.*, 1msec) has a marginal effect on the energy consumption compared to deciding whether to turn UE's circuits ON or OFF. To illustrate this comparison, we present the following numerical example. Assume a single cell operating with the full LTE bandwidth of 20MHz with a total of 100 RBs available in each TTI. The 100 RBs are assumed to be allocated to a single user within a single TTI. The user's effective SNR is assumed to be 20dB. With the aid of equations (4.4) and (4.5) and Table 4.1 in [1], the total power consumption $P_k(m, \mathcal{N}_{j,k}(m))$ of the UE within the observed TTI when all RBs are configured to MCS index 1 is equal to 3.92W. Whereas, in the case of MCS index 15, the total power is 4.11W. Thus, there is a maximum of 4.62% reduction in the UE's power consumption if the scheduler coarsely changes the MCS index over the allocated RBs from index 15 to index 1. On the other hand, turning UE's circuits ON and OFF in each TTI affects the UE's power consumption budget by a value 1.45W (*i.e.*, $P_{on} + P_{rx} - P_{idle}$) that is almost equivalent to 35.28% of the total power consumed by the baseband and RF circuits.

Based on the finding of the previous example, and to simplify the scheduler's formulation, we further modify the formulation in (4.10) by removing the term $P_k(m, \mathcal{N}_{j,k}(m))$ from the cost function and allowing the scheduler to focus only on minimizing the number of wake-up TTIs as it offers a remarkable reduction in the UE's energy consumption budget. Thus, the formulation in (4.10) can be re-written as follows:

Min

$$\begin{aligned} Z_2 = T_s \sum_{k=1}^{K} \sum_{m=m_o}^{m_o+G-1} w_k P_c \, \Phi_k(m) + \\ \sum_{k=1}^{K} \sum_{h=1}^{H} \alpha_{k,h} \left(\begin{array}{l} \left(\tau R_D^{(k)}(h) + \ell(k,h)|_{m_o-G}^{m_o-1}\right) \Omega_k \\ - \sum_{m=m_o}^{m_o+G-1} \sum_{n=1}^{N} B_k^h(m,n) \Psi_k^h(m,n) \end{array} \right) \end{aligned} \tag{4.11a}$$

Subject to

$$\sum_{m=m_o}^{m_o+G-1} \sum_{n=1}^{N} B_k^h(m,n)\, \Psi_k^h(m,n) \leq \; \tau\, R_D^{(k)}(h) + \ell(k,h)|_{m_o-G}^{m_o-1} \;, \;\; \forall\, k, h \tag{4.11b}$$

$$\sum_{k=1}^{K} \Psi_k^h(m,n) \;\leq\; 1, \quad \forall m, n, h \tag{4.11c}$$

$$\Psi_k^h(m,n) \; - \; \Phi_k(m) \;\leq\; 0\,, \;\; \forall k,\, h,\, m,\, n \tag{4.11d}$$

$$(4.10d) \tag{4.11e}$$

As explained in the previous two paragraphs, the modified cost function in (4.11a) focuses mainly on minimizing the number of scheduled wake-up TTIs for each UE while maintaining the connection's target transmission rate throughout the connection duration T. The reader can also notice another difference between (4.11) and (4.10). That is, changing the notations of the variables $\Psi^h_{\mathcal{N}_{j,k}(m)}(m)$ and $B^h_k(m, \mathcal{N}_{j,k}(m))$ that appeared in (4.10) to $\Psi^h_k(m,n)$ and $B^h_k(m,n)$ in (4.11). In particular, the scheduler's simplified formulation in (4.11) does not care about the sets of RBs allocated in every TTI for each user to optimize the UE's baseband and RF power consumption (*i.e.*, $P_k(m, \mathcal{N}_{j,k}(m))$) as was the case in (4.10). This is due to the fact that the practical design for LTE scheduler suggests that for each UE all RBs within the same sub-frame are preferably adjusted to a fixed MCS [10]. Therefore, the new decision variable $\Psi^h_k(m,n)$ introduced in (4.11), which substantially reduces the solution space, is designated only to identify the number of scheduled transmitted bits over each individual allocated RB for each user (*i.e.*, $B^h_k(m,n)$) to meet the user's connection target rate. Hence, we define $\Psi^h_k(m,n)$ as a binary decision variable which indicates whether connection h of user k has been assigned to RB n during TTI m or not, and $B^h_k(m,n)$ as the number of allocated bits for connection h of user k over RB n during TTI m.

4.5 Heuristic Scheduler

In this section we design a heuristic algorithm for simplifying the solution of the optimization problem defined in (4.11). This is derived by the high complexity inherent in the optimal model especially for large values of τ which strictly determines the problem's solution space. In order to provide an approximate figure about the complexity of solving the problem in (4.11), we provide the following timing measurement for a small scale problem. Consider three users each with a single connection and three available RBs at each TTI. In an attempt to solve problem (4.11) with a value of τ= 10msec (*i.e.*, 1 frame of scheduling granularity) over a total number of frames M= 5000 (*i.e.*, T= 50sec), the MATLAB Profiler recorded a total elapsed time of 565.45hrs (*i.e.*, 23.5 days). The MATLAB was running on an Intel Xeon CPU W3670 with 6 core processors running at 3.2GHz and 16-GB of RAM. On the other hand, our proposed heuristic algorithm (*i.e.*, discussed below) was able to solve the same problem in just a few seconds.

4.5.1 Heuristic Algorithm

The proposed heuristic algorithm as depicted in Fig. 4.5 is fed by an initialization part, labeled by 1, which is created to set the parameters of the considered scenario. These parameters include: number of users (K), number of connections for each user (H), total number of TTIs considered for the users connections (M), number of available RBs per TTI (N), scheduler's granularity in TTIs (G) (*i.e.*, taken in multiples of a frame), quasi-instateneous target rate for each user connection $R_D^{(k)}(h)$. Furthermore, to calculate the total power consumed by each user (*i.e.*, $P_t^{(k)}$) every TTI using the model described in (4.3). First P_r (*i.e.*, in (4.5)) is calculated each TTI for each user by employing a reference channel model (*i.e.*, discussed in detail in the next section). Second, using Table 4.1 and based on the user's generated channel SNR, B_r (*i.e.*, in (4.4)) is calculated accordingly each TTI. It should be noted that part 1 of Fig. 4.5 will be also used when solving the optimal problem in (4.11).

The second part of Fig. 4.5 labeled by 2 represents the core heuristic algorithm. The algorithm is designed to run in a sequential manner on the requested users connections. In other words, the algorithm allocates resources for one user connection at a time. The connections scheduling sequence order is determined by the parameter $\alpha_{k,h}$ for each connection. In our heuristic, the parameter $\alpha_{k,h}$ is set to be proportional to the user's traffic queue length which reflects the priority of scheduling that queue. Obviously, the length for each user queue is continuously changing in time based on the number of allocated resources and their respective capacities during each scheduling period τ. So large queue length is equivalent to large value of $\alpha_{k,h}$, and hence, higher priority is allotted compared to smaller length queue.

The algorithm works as follows. For each G TTIs, that is equivalent to τ sec from the total connection time T, all users connections are sorted according to their corresponding queue lengths. Then, for each connection according to the sorted order, the algorithm

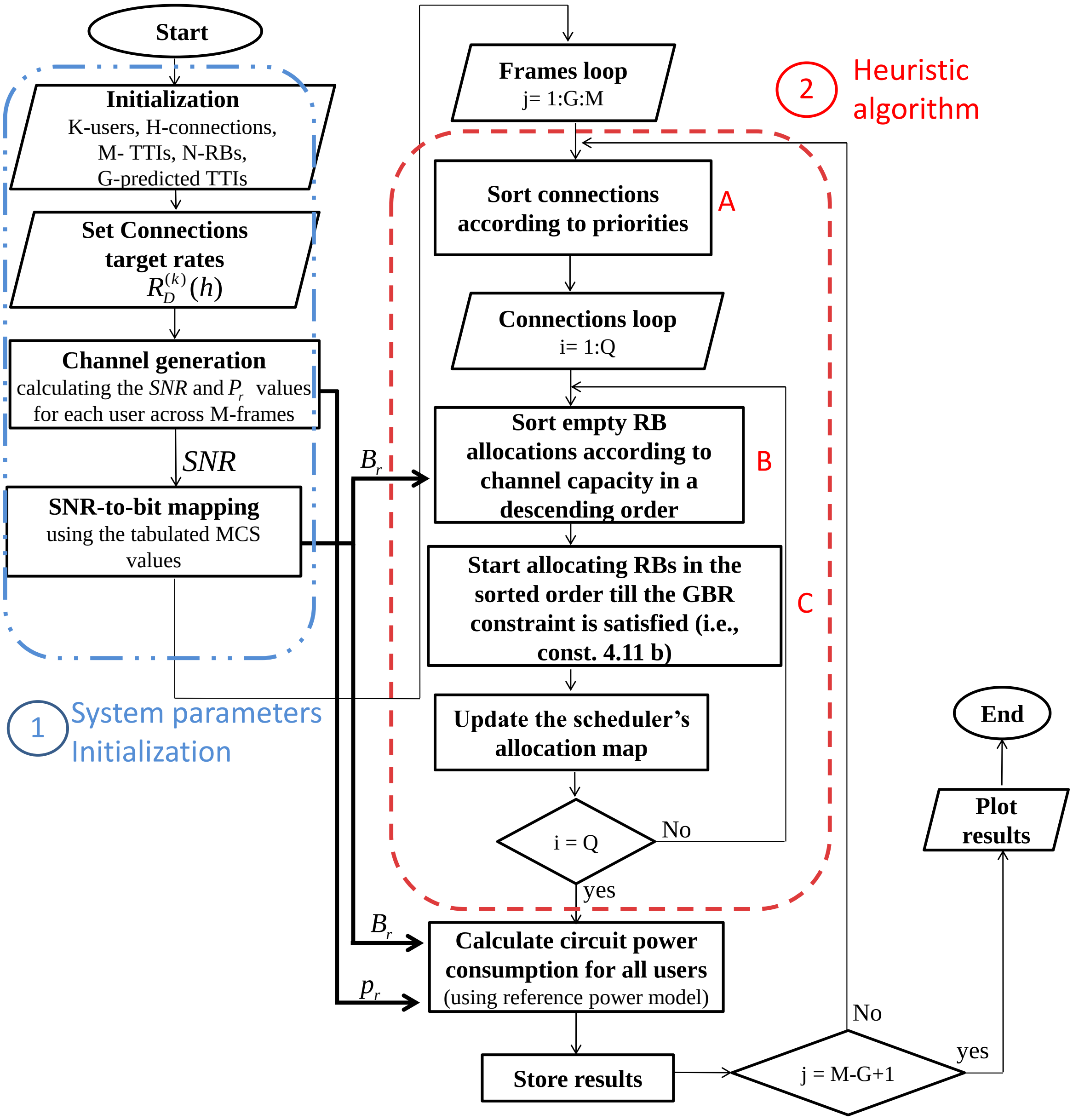

Start
Initialization
K-users, H-connections,
M- TTIs, N-RBs,
G-predicted TTIs
Set Connections target rates
$R_D^{(k)}(h)$
Channel generation
calculating the SNR and P_r values for each user across M-frames
SNR
SNR-to-bit mapping
using the tabulated MCS values
1 System parameters Initialization
Frames loop
j= 1:G:M
2 Heuristic algorithm
Sort connections according to priorities
A
Connections loop
i= 1:Q
Sort empty RB allocations according to channel capacity in a descending order
B
Start allocating RBs in the sorted order till the GBR constraint is satisfied (i.e., const. 4.11 b)
C
Update the scheduler's allocation map
i = Q
No
yes
B_r
B_r
p_r
Calculate circuit power consumption for all users
(using reference power model)
Store results
j = M-G+1
No
yes
Plot results
End

Figure 4.5: Heuristic algorithm flowchart for the proposed scheduler

aims to allocate the minimum number of RBs within the smallest possible number of TTIs which gives a total number of scheduled bits less than or equal to the target bits (*i.e.*, equivalent to constraint (4.11b)). This is done by sorting the unallocated RBs within the observed TTIs in descending order according to their capacities (*i.e.*, number of received bits per RB). Sorting the RBs in this way results in both minimum number of allocations and TTIs. This is due to the fact that, for each UE, all RBs capacities (which correspond to different MCSs) are set to be equal to the lowest RB capacity (*i.e.*, MCS index) within the same TTI. In the LTE standard, it is known to be highly efficient to reduce the signaling overhead by making the UE's receiver (or transmitter) circuit adjusted to a fixed MCS rather than using frequency-dependent MCS at a given subframe (Section 10.2 in [10]). In this case, the scheduler is strictly enforced to fully allocate RBs in each TTI before moving to another TTI, and hence, minimizing the overall number of wake-up TTIs and circuit energy consumption. The scheduler then updates the allocation map for all RBs across the observed time slots with the current connection allocations before proceeding with the next connection in the sorted listed. The algorithm continues until all connections in the sorted list are served. Based on the RB allocations for each user, the receiver's circuit power consumption is calculated as in (4.3) during the current scheduling period and then stored for later analysis. The whole algorithm continues in the same fashion for the next scheduling period (*i.e.*, next G TTIs) until the last frame in the established connection time.

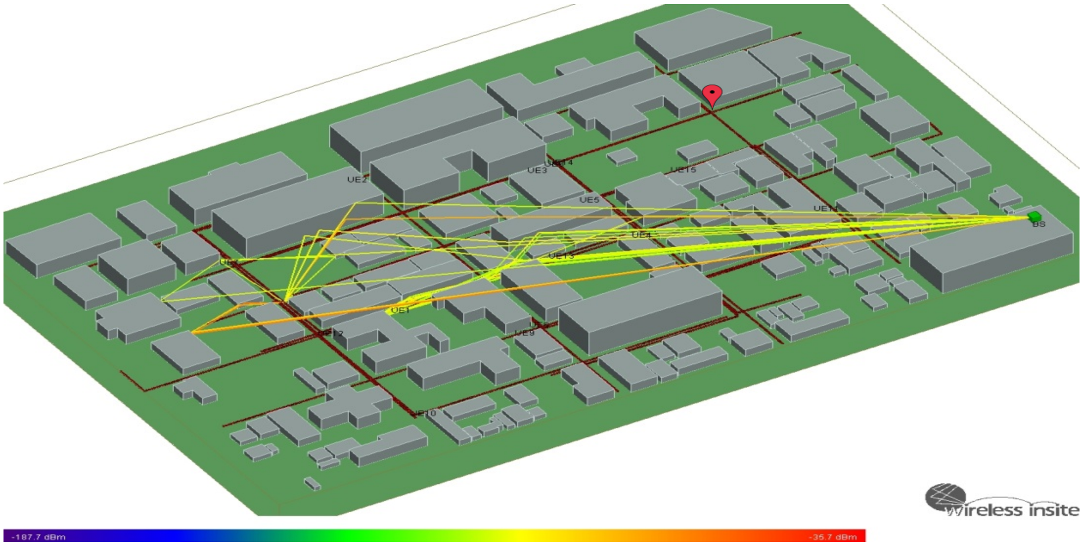
4.5.2 Complexity Evaluation

For the sake of assessing the algorithm complexity, we divide it into three major processing components labeled as: A, B and C as depicted in Fig. 4.5. Those components hold the main operations constituting the algorithm. Component A, as previously explained, is responsible for sorting the admitted users' connections based on their queue lengths. Hence, the complexity of component A is equal to O($Q \log(Q)$), where Q is the total number of connections for all admitted users. Component B sorts empty RB allocations to rank potential assignments to the observed connection. This requires first searching the indexes of unallocated RBs during the observed TTIs (*i.e.*, of number G) then sorting them according to their capacities. Therefore, the worst case complexity for component

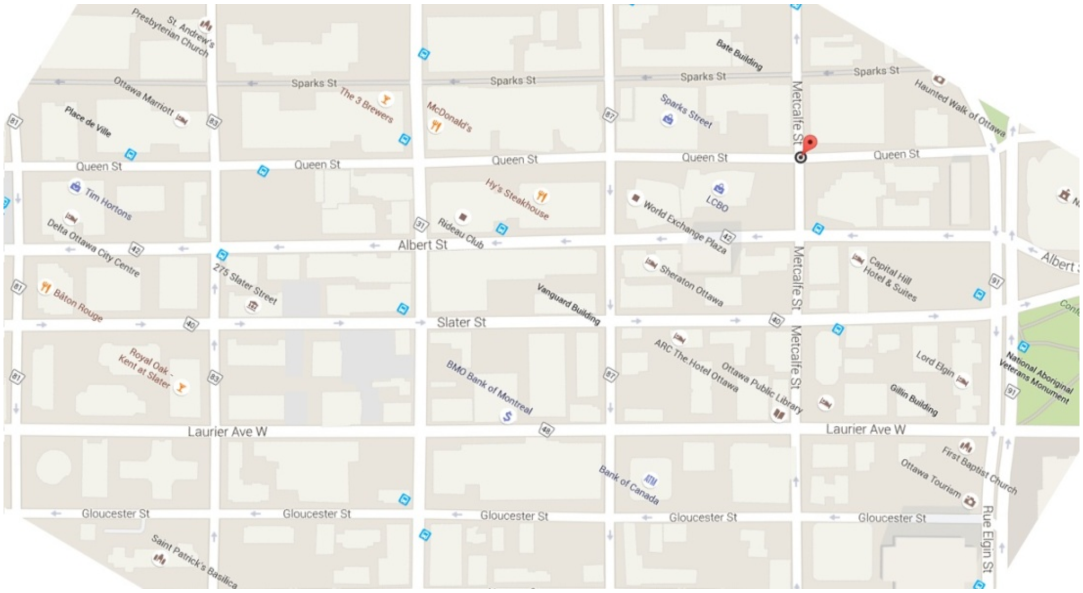
B, when allocating resources for the first connection in the sorted list, is equal to O(NG) + O($NG\log(NG)$), where NG is the total number of RB allocations in G TTIs. Finally, component C keeps checking capacities for all unallocated RBs within the observed G TTIs to make final decisions about RB allocations which satisfy the user's connection buffer requirement (*i.e.*, equivalent to constraint (4.11b)). This results in an upper bound complexity of O($NG\log(NG)$). In sum, the asymptotic upper limit for the algorithm complexity when allocating resources for a total of Q connections can be approximated by O($Q\log(Q)$) when $Q \gg NG$, or O($NG\log(NG)$) when $NG \gg Q$.

4.6 Numerical Results

In this section, the performance of the proposed predictive scheduling scheme is compared with the green resource allocation (GRA) scheme proposed in [78] through MATLAB numerical simulations. To the best of our knowledge, the GRA is the only reported scheme that has been designed for optimizing the UE EE in downlink OFDMA systems, and hence, is exclusively considered in our comparison. The numerical simulations are conducted in two different scenarios both of which exclusively focus on an outdoor radio propagation scenario. In the first scenario, the channel is modeled as a quasi-static block Rayleigh fading (QSBR) channel [14]. The channel is assumed to be constant within the 180kHz band of each RB during each TTI. However, it changes randomly and independently from one sub-band to another (*i.e.*, frequency selectivity) and from one TTI to another (*i.e.*, time selectivity). Each UE is also assumed to experience independent fading. Considering the Rayleigh fading in this scenario, the distribution of the instantaneous (*i.e.*, taken every subframe) received channel SNR over each RB follows the exponential distribution [63]. The second scenario, as shown in Fig. 4.6, utilizes a real ray tracing measurements for the propagation channel of mobile UEs located in the north of centretown of Ottawa city, Canada. This location represents an area in downtown Ottawa which includes Gloucester St., Laurier Ave W, Slater St., Albert St., Queen St., Sparks St., Lyon St. N, Kent St., Bank St., O'Connor St., Metcalfe St. and Rue Elgin St. The picture in Fig. 4.6a shows the actual ray tracing experiment carried out using the Remcom's Wireless Insite tool [19].



(a) Ray tracing 3D view



(b) Google map top view

Figure 4.6: 3D ray tracing experiment for part of Ottawa city

The MATLAB model for both scenarios is implemented just as illustrated in the flowchart of Fig. 4.5. The only two variable parts are the channel generation block located inside the initialization part labeled by 1 and the heuristic algorithm labeled by 2. The channel generation block is based on the simulation scenario (*i.e.*, channel model) being considered while the second part is based on the scheduler type (*i.e.*, proposed optimal, proposed heuristic, GRA optimal and GRA heuristic). In other words, in case of scenario 1, the RT-based scheduler highlighted in Fig. 4.4 is assumed to have the same channel measurements as those generated by the QSBR model. In the case of scenario 2, the RT-based scheduler is fed by real ray tracing measurements conducted by the Wireless Insite tool experiment illustrated in Fig. 4.6a. As explained in Section 2.6, the MATLAB ray tracing environment is directly interfaced to the Insite tool via a piece of MATLAB script.

For both scenarios of the simulation, all users are assumed to be located within a single cell coverage and receiving downlink connections from its serving eNB. In scenario 1, two kinds of investigation were undertaken. The first focuses on the performance and complexity comparison of the optimal and heuristic versions of the proposed scheduler - described in formulation (4.11) and Section 4.5, respectively - to that of the GRA scheduler. For this and due to the high computational burden and latency of the optimal scheduler, a relatively small number of UEs (*i.e.*, set to be equal or less to the number of available RBs) are admitted to request downlink connections from the eNB for a small number of frames (*i.e.*, the simulation time). Having the proposed heuristic algorithm benchmarked, the following investigation is devoted to study the system capacity variations and users' buffers queue stability only for the heuristic versions of the proposed and the GRA schedulers. Thus, the number of admitted UEs, as well as their requested connections duration, are allowed to be increased while keeping the number of available resources constant. This increase can be easily handled due to the dramatic speed-up and simplicity of the heuristic scheduler compared to its optimal counterpart. The evaluation of the proposed scheduler takes place at different ray tracing prediction ranges (or scheduling granularity) to show its impact on the scheduler's overall performance. For the sake of simplicity, and due to the limited available computing resources, only the second investigation has been carried out in scenario 2.

4.6.1 Scenario 1: QSBR Channel

As described earlier, scenario 1 utilizes the QSBR channel model for setting the UEs' propagation statistics. In MATLAB, we generate independent and identically distributed (iid) random variables. Each random variable, which models the downlink channel SNR values for a certain UE over a single RB across the selected M frames, has an exponential distribution with an assumed average of 10dB. Since the adopted QSBR channel is known to model scattering environments with multiple path propagation. We use the Rayleigh channel model with the covariance function in the form of [14]:

$$R_{\xi}(t_s) = J_0(2\pi f_d t_s) \tag{4.12}$$

where ξ is the channel Gaussian process, t_s is the channel sampling time, J_0 is the Bessel function of the first kind with order 0 and f_d is the Doppler frequency. In order to determine a proper value for the channel coherence time (τ_{coh}), equation (4.12) needs to be evaluated at different speeds of the mobile terminal and the operating carrier frequency (*i.e.*, 2.6GHz). Based on the curves shown in Fig. 4.7, taking 0.5 as threshold value to determine the channel coherence time, it is obvious that any speed that is greater than or equal to 100km/h leads to $\tau_{coh} <= 1$msec. As a result, 1msec is assumed to be the sampling time for the generated random variables.

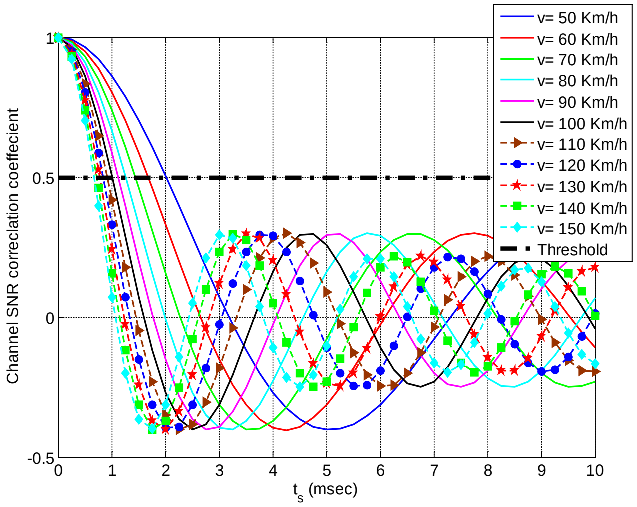


Figure 4.7: SNR correlation coefficient for a Gaussian process at different mobile speeds and carrier frequency of 2.6GHz

4.6.1.1 Performance and complexity comparison of the proposed scheme with the GRA scheme

In this part we compare the EE performance and complexity of the optimal and heuristic versions for both of the proposed and GRA schemes. It is worth noting that the GRA scheme does not utilize the RT engine knowledge about the users' CSI as is the case with the proposed scheme. As discussed in the previous sections, the EE is compared in conjunction with satisfying a quasi-instantaneous target rate for each UE. Due to the inherent complexity of the optimal solution, for both schemes, we run the optimal schedulers for only 200 frames to find the global optimal allocations and compare them with those in the case of the heuristic. We also assume a small size system which allows only 3 UEs, each with single downlink connection with a unique required rate and competing over 3 available RBs. The selected rates are 13.3kb/sec, 64kb/sec and 128kb/sec that typically support VoIP, audio streaming and FTP connections, respectively.

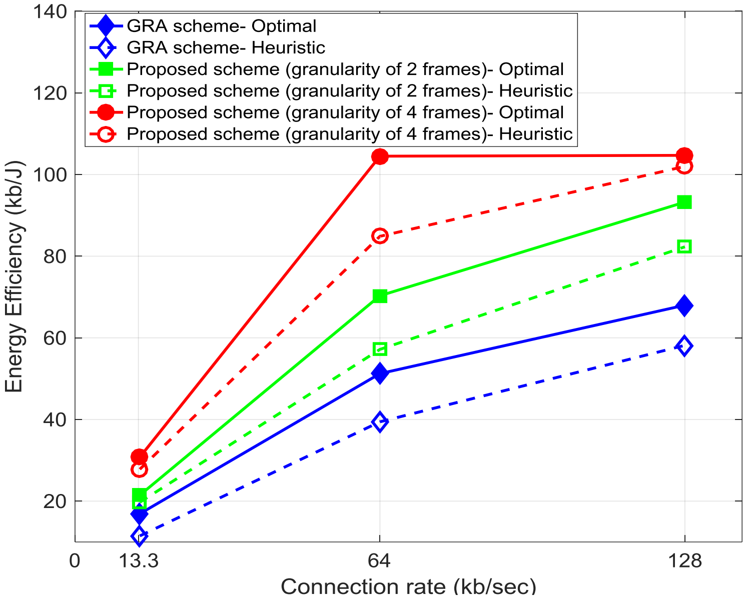


Figure 4.8: Energy efficiency comparison for the proposed vs. the GRA scheduler

Fig. 4.8 shows how the proposed scheduling scheme performs and compares with the GRA scheme in terms of the achieved EE especially when increasing the scheduling time granularity (*i.e.*, measured in frames) for our proposed scheme. It can be seen that the

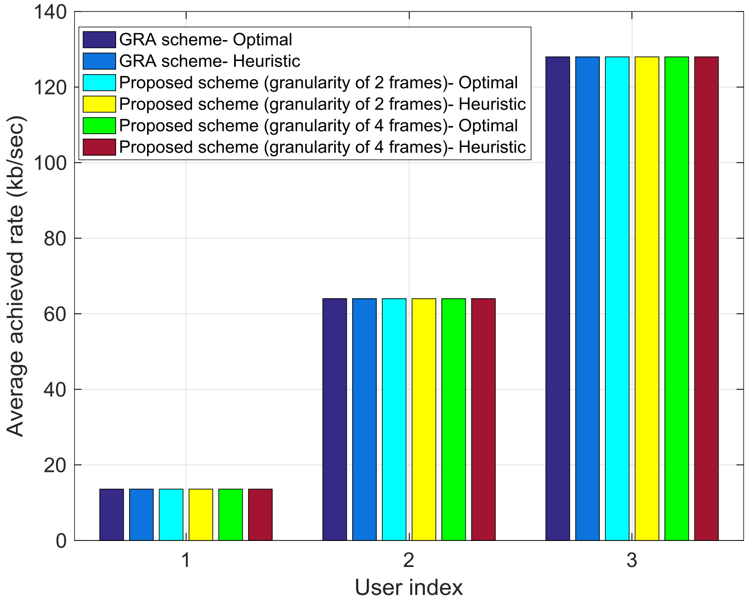


Figure 4.9: Quasi-instantaneous rate satisfaction for the proposed vs. the GRA scheduler

proposed scheduler shows a significant EE improvement when acquiring greater knowledge about the UE's CSI which leads to increasing the optimization problem's solution space. It is also noticed that the EE of the higher rate connections is greater than that of the lower rate. This is due to the fact that for higher rate connections the scheduler relatively allocates more resources within each TTI to support its high volume of data. Thus, the number of bits received for every joule consumed by the UE per TTI becomes larger on average. However, it should be noticed that the UE with the 128kb/sec connection have similar EE to that of the 64kb/sec UE in case of the proposed scheme at 4 frames of scheduling granularity (*i.e.*, last two points on the red solid line). This can be attributed to the scheduler's increased ability to fully satisfy the connection rate requirement for the 64kb/sec UE at 4 frames of granularity, while spending half the energy consumed by 128kb/sec UE.

The results shown in Fig. 4.9 support the significance of that shown in Fig. 4.8 because they show that the rate requirements of all UEs for both of the proposed and GRA schedulers are equally met. From another perspective, Fig. 4.10 provides more insight on the increased capability, at larger time granularity, of the proposed scheduler for satisfying the quasi-instantaneous rate (*i.e.* measured every 80msec) of the 13.6kb/sec connection.

Fig. 4.11 shows the increase in batteries lifetime in case of the proposed scheme (as

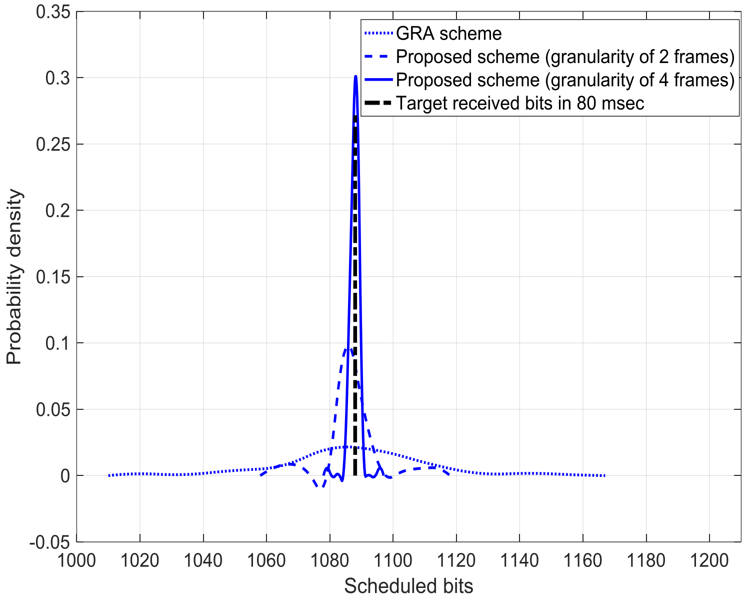


Figure 4.10: The proposed vs. the GRA scheduler capability for satisfying the 13.6kb/sec connection

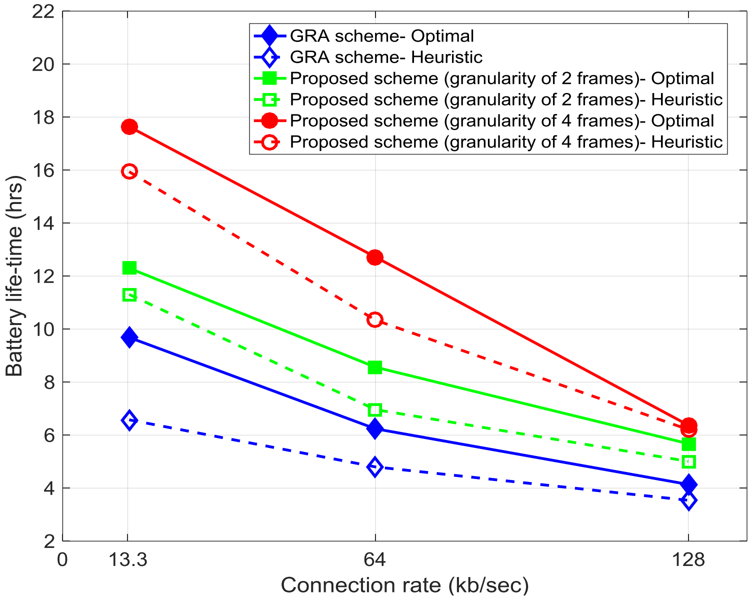


Figure 4.11: UE battery lifetime for the proposed vs. the GRA scheduler

function of the scheduling granularity) compared to the GRA scheme for different UEs. It is worth noting that the lifetime values obtained are strictly dependent on the energy consumption by the UE while receiving data in the downlink and an assumed battery capacity of 2915mAh. Thus, the lifetime results shown in Fig. 4.11 do not account for any other factors that are known to deplete the cell phone battery (*e.g.*, running any kind of applications, uplink transmission, synchronization with the eNB, etc).

On the other hand, the complexity of the proposed scheme is compared with the GRA scheme in terms of the computation time spent by the CPU to solve the allocation problem. The computation times, as recorded by the MATLAB Profiler, are shown in Table 4.2.

Table 4.2: Complexity comparison

Computation Time **Scheme**	**Optimal**	**Heuristic**
GRA scheme	56.2hrs	184ms
Proposed scheme	92.9hrs @ granularity of 2 96.7hrs @ granularity of 4	213.3ms @ granularity of 2 267ms @ granularity of 4

4.6.1.2 Buffer queue stability with system overloading in case of heuristic schedulers

After benchmarking the heuristic schedulers for both of the proposed and the GRA schemes in part 1 of the results, in this part we only consider the heuristic schedulers (for both schemes) for investigating the system stability when stressing the system by increasing the number of UEs. This is due to the substantial speed-up for the heuristic algorithm compared to solving the problem's optimal formulation. We also increase the value of M to 5000 frames (with the same 10msec frame duration). For evaluating high data rate applications, the number of available resources per TTI is increased to 25 (*i.e.*, equivalent to the 5MHz LTE channel) instead of 3 as used in the previous part. The number of UEs is allowed to increase from 3 to 15 in a step of 1. In addition, a granularity of 8 frames for the scheduler is added to the test to uncover a bigger picture for the system's behavior. All UEs are assumed to have a single downlink connection with a rate of 400kb/sec (*i.e.*, recommended bit rate for a 240p YouTube live stream). It is worth noting that the delay requirement for each connection is considered in this analysis as highlighted in the

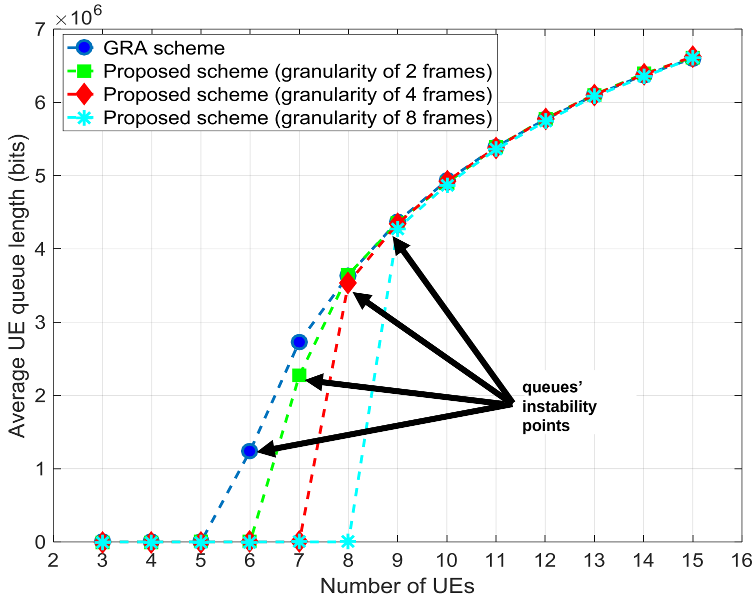


Figure 4.12: UE's buffer queue stability

last paragraph of Section 4.2.1. In particular, the delay is bounded by satisfying the effective quasi-instantaneous rate for all the requested connections within a time horizon that does not exceed 100msec (*i.e.*, the typical packet delay budget for various services such as conversational voice, real-time video and games [11]). In addition, the packet delay jitter which is a crucial QoS parameter for certain traffic types (*e.g.*, VoIP and online gaming) is not considered in this work. However, more generalized system model and optimization framework which accounts for modelling and controlling the packet delay jitter (for jitter sensitive applications) is rigorously studied in the following chapters.

The results depicted in Fig. 4.12 show how the UE buffer queue length increases with the number of UEs for both of the proposed scheme and the GRA scheme [78]. This increase is a direct result of overloading the system available resources with a potentially increasing number of connections. The results confirm that the proposed scheme outperforms the GRA scheme especially when increasing the scheduling time granularity. This increase leads to maintaining the average queue length per UE at an acceptable level for larger number of admitted UEs. More specifically, for the GRA scheme it is clear that the instability point (*i.e.*, point at which the UE buffer length grows without bound) appears at a smaller number of UEs compared to the proposed scheme. As a result, the proposed

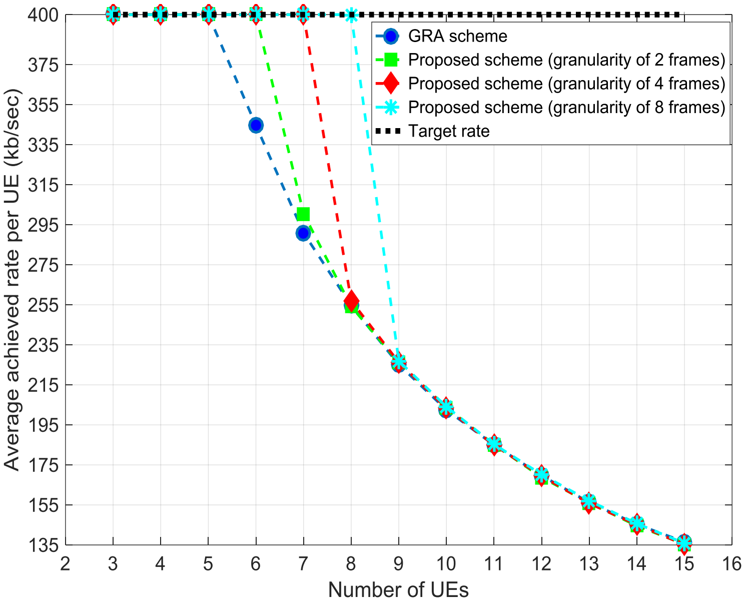


Figure 4.13: UE's average achieved rate

scheme is capable of increasing the system's resistance to instability, and hence, potentially increases the system's capacity to admit more users. This is obvious for the proposed scheme with 8 frames of granularity that can support up to 8 UEs compared to the GRA scheme which can only support up to 5 UEs while having an equal number of available resources. In addition, the results shown in Fig. 4.12 confirm the idea explained in Fig. 4.1. Our predictive scheduler is capable of meeting the QoS requirements while allocating less number of resources when operating at higher time granularity, and thus having higher UEs' admittance capacity.

As a consequence of the queue stability regions shown in Fig. 4.12, the average achieved rate per UE depicted in Fig. 4.13 could be directly justified. In other words, the deviation of the average achieved rate per UE from the target rate (*i.e.*, 400kb/sec) as the number of UEs grows reflects the growth in the queue length noticed in Fig. 4.12. However, the system capacity is the same (*i.e.*, the total cell rate), as it should be, although the average rate per UE is changing with the number of UEs. From another perspective, the distribution of the total cell rate among different UEs, as shown in Fig. 4.14 for the case of 15 UEs, evaluates the inherent fairness property of our proposed heuristic algorithm among different UEs with the same traffic connection requirements. This property is implicitly understood

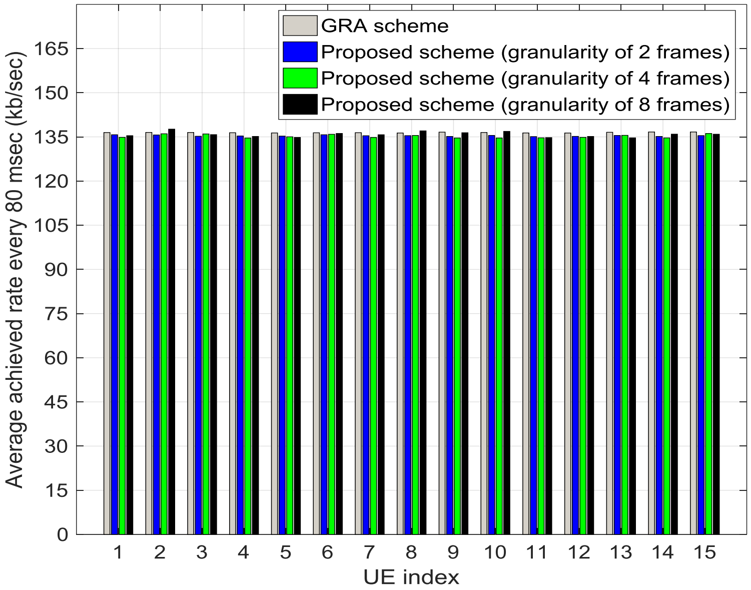


Figure 4.14: Distribution of the total cell rate in the case of 15 UEs

from the explanation of the algorithm provided in Section 4.5. In particular, the uniform distribution of the total cell rate among all UEs (*i.e.*, taken as a fairness indicator) depicted in Fig. 4.14 is the result of continuously changing the scheduling priority among users having the same service requirements based on their varying queue lengths (*i.e.*, equivalent to the throughput history). The same strategy is used in the well known proportional fair (PF) scheduling policy [88]. It is also worth noting that the fairness is considered in the optimal model described in (4.11) by setting equal values to the weighting factor $\alpha_{k,h}$ to connections having the same service requirements across different UEs.

In addition to its ability of increasing the number of serviced UEs, the energy efficiency improvement provided by our proposed scheme is noticed. This could be seen in figures 4.13 and 4.15. We start by focusing on the common stability region for all curves where the number of UEs increases from 3 to 5. A significant increase in the EE is clearly noticed in Fig. 4.15 for the proposed scheme over the GRA scheme. This increase ranges from 38.43 $\%$ at granularity of 2 frames and to 62.47 $\%$ at granularity of 8 frames. This is due to a substantial drop in the energy consumption by the same percentages as noticed in Fig. 4.15 while almost having the same average rate per UE as shown in Fig. 4.13.

Yet another conclusion could be drawn from Fig. 4.15. It could be noticed that, for

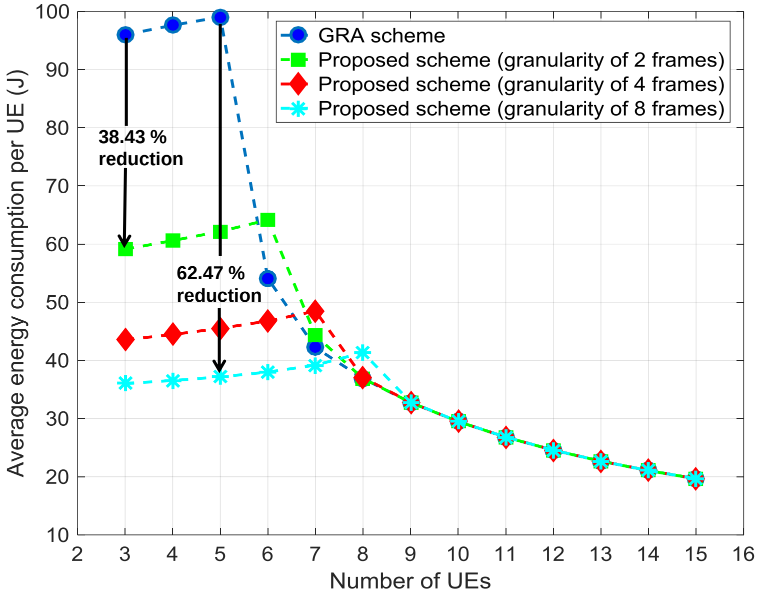


Figure 4.15: Average energy consumption per UE

each curve, the energy consumption slightly increases as the number of UEs grows until it reaches the maximum value before the corresponding instability point (*e.g.*, 6 UEs for the GRA scheme curve). This is due the higher load experienced by the system's frequency resources. In other words, when the system is stable (or relaxed), increasing the number of UEs slightly limit the energy optimization due to the increasing load on the limited available resources. Hence, the average energy consumed per UE shows a slight increase (*i.e.*, lower optimization efficiency). However, that effect becomes less pronounced in the case of the proposed scheme as the scheduling granularity increases compared to the GRA scheme. This can be seen when comparing the rising slopes of the GRA scheme and the proposed scheme with granularity of 8 frames curves in Fig. 4.15.

4.6.2 Scenario 2: RT-based Channel

In this scenario, we examine a practical use case for measuring the performance of our scheduling scheme in comparison with the GRA scheme. We used a commercial radio propagation prediction software named Wireless Insite that is offered by Remcom Inc. [19] to build a realistic 3D urban scenario as shown in Fig. 4.6a. The scenario detailed parameters are listed in Table 4.3. The reason behind conducting this experiment with real ray tracing measurements for the propagation channel is to provide an insight on the performance bounds of our scheduling scheme with two different channel models, one of which is based on a practical scenario. Therefore, the results of scenario 2 should be looked at in comparison with that in scenario 1 which utilizes one of the common channel models used in the literature (*i.e.*, QSBR model).

Table 4.3: Simulation parameters

Parameter	**Setting**
Number of UEs	15
UE speed	50km/h
UE route length	690m
Number of available RBs	3
Operating frequency	2.6GHz
Available MCS	refer to Table II in [1]
Channel model	RT using SBR technique
Layout	urban with 1 microcell
Cell dimensions	1016m (L) x 673m (W)
Building walls relative permittivity (ε_r)	3 **[47]**
Building walls conductivity (σ)	0.005S/m **[47]**
Building walls thickness	20cm
Ground permittivity (ε_r)	15 **[47]**
Ground conductivity (σ)	7S/m **[47]**
eNB antenna type	vertical isotropic
eNB antenna height (above the ground)	57m
eNB transmit antenna input power	48dBm
UE antenna height (above the ground)	2m
Simulation time	50sec (*i.e.*, 5000 frames)

Unlike the results obtained in Fig. 4.12, both of the buffers' stability performance and the system's admittance capacity for the proposed scheme showed a slight improvement over the GRA scheme as demonstrated in Fig. 4.16. We attribute this different behavior due to the slow fading channel of the chosen urban scenario where the UEs are moving with a speed of $V_{UE} = 50$km/h. This slow speed results in much greater channel coherence time (*i.e.*, $\tau_{coh} >$10msec) compared to that used in scenario 1 (*i.e.*, τ_{coh}= 1msec). This is clearly illustrated in Fig. 4.17 which captures 5000 samples of both channels. Consequently, our proposed predictive scheduling scheme is not able to exploit better scheduling chances, as

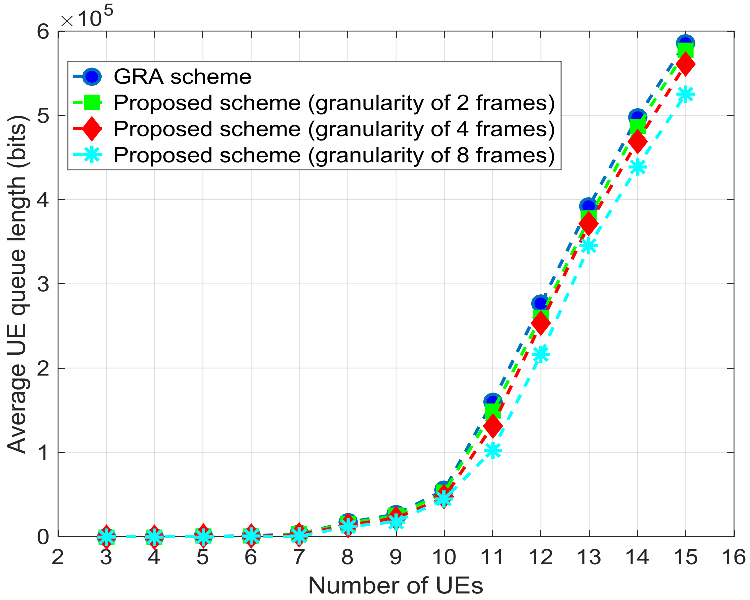


Figure 4.16: UE's buffer queue stability

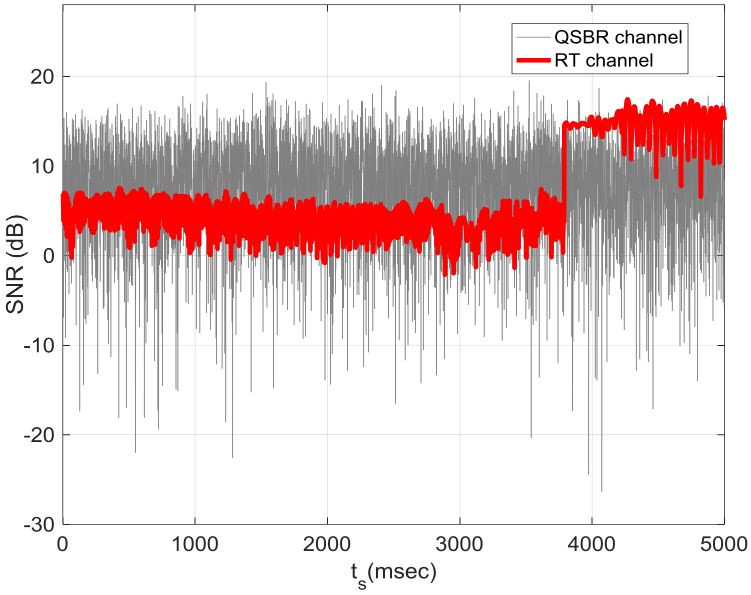


Figure 4.17: Snap shot for 5 sec of the QSBR and RT channels

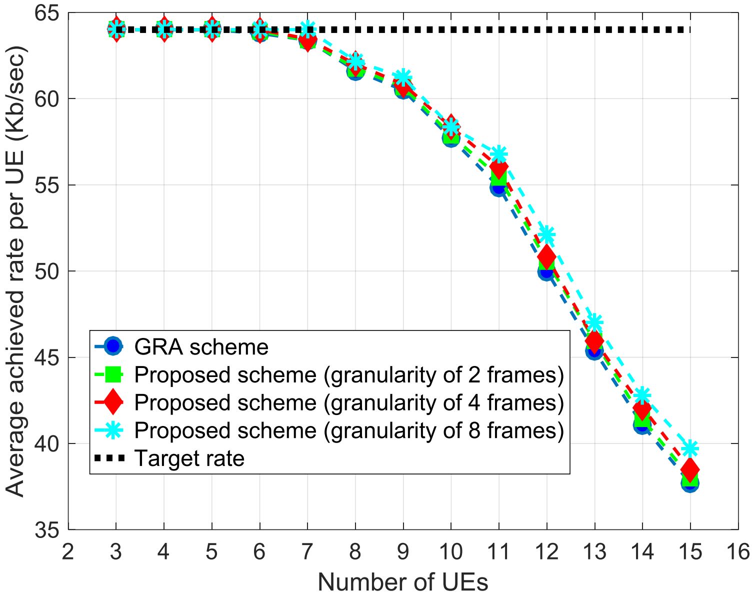


Figure 4.18: UE's average achieved rate

is the case in simulation scenario 1, especially at the selected low granularities. However, it is still believed that arbitrarily increasing the scheduling granularity would be able to show better stability performance and increased system's admittance capacity than that shown in Fig. 4.16. However, this is beyond the scope of this chapter due to the expected delay limitations which might appear in this case. In the same context, the drop in the average rate per UE that appears in Fig. 4.18 is consistent with the results shown in Fig. 4.16.

On the other hand, despite the stability performance observed in Fig. 4.16, a substantial energy reduction percentage per UE for the proposed scheme within the stability region (*i.e.*, number of UEs= 3 to 10) in Fig. 4.19 still exists. That reduction ranges from 25 $\%$ to 56.6 $\%$ compared to the GRA scheme. Thus, although failing to boost the system's capacity, our proposed scheme is still able to improve the UE's EE by increasing the scheduler's time granularity in the presence of slow varying channels.

4.7 Chapter Summary

In this chapter, we developed a framework for implementing a QoS-aware energy efficient predictive scheduling approach for the downlink in OFDMA based cellular systems utiliz-

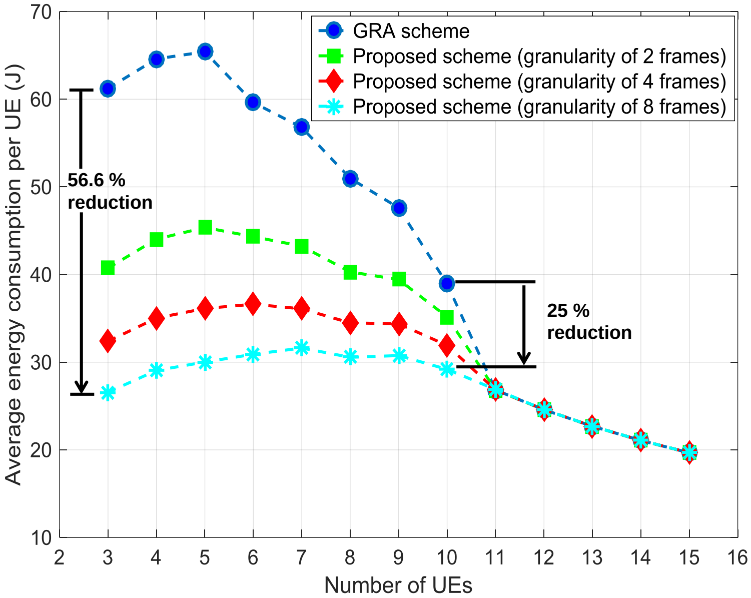


Figure 4.19: Average energy consumption per UE

ing the well known, site-specific, ray tracing approach.

First, we proposed the downlink ray tracing based scheduling system. Second, based on a practical model for the LTE UE power consumption, we formulated a hard constrained quasi-instantaneous rate problem with an objective to minimize the UE's receiving energy consumption in the downlink. Due to the natural channel capacity limitations, and accounting only on the dominant components of the UE's receiver power consumption model, our problem formulation undergoes a series of modifications until we reach a practical formulation. Third, we designed a heuristic algorithm to relax the inherent computational burden in the optimal scheduler. To study the performance bounds, the proposed schedulers were comparatively evaluated with respect to the exisitng GRA scheme [78] twice, once in the presence of fast (*i.e.*, QSBR model) and again in the slow (*i.e.*, practical 3D urban scenario) fading channels. In the presence of the fast fading channel, our proposed scheme was able to improve the EE and the scheduler's capacity to serve more UEs by up to 62.47 $\%$ and 60 $\%$, respectively, compared to the GRA scheduler. On the other hand, in the presence of the slow fading channel, despite showing no effect on the scheduler's admittance capacity, the proposed scheme was still able to improve the UE's EE by up to 56.6 $\%$ compared to the GRA scheme.

In sum, it could be seen that our proposed scheduling scheme can work effectively and cooperatively with the current 3GPP LTE DRX power management scheme to prolong today's smart cell phones battery lifetime per charge.

Chapter 5

Modelling and Optimizing Delay Jitter in Communication Networks

5.1 Introduction

Delay jitter of data packets is known to be a crucial quality of service (QoS) measure especially for real-time applications (*e.g.*, Voice over LTE (VoLTE)). It takes place as a result of the queuing, scheduling and routing latencies within the network. However, control schemes that directly tackle the jitter problem in today's advanced wireless systems are rare. To enable such schemes, proper modelling of the packet delay jitter is an essential preliminary step.

In the first part of this chapter, a comprehensive mathematical modelling for the packet delay jitter in a simple queuing system with one traffic buffer of infinite length, one server and single hop is presented. In contrast to independent and identically distributed (iid) models, the analysis focuses on the correlated nature of service intervals. The presented models study different scenarios and parameters for the queue in terms of the system's utilization and the probability distribution of data packets' service and interarrival times, respectively. Numerical simulations demonstrate the high accuracy achieved by the presented models.

In the second part, we study the EE of the user equipment (UE) in the LTE downlink and the delay jitter as a fundamental QoS metric for most real-time applications. The study

---

The first part of this chapter has been published in [89], and the second part is submitted to [90].

focuses mainly on the VoLTE traffic as being a heavily used service. We provide a multi-objective optimization for both the EE and the delay jitter subject to fixed delay budget. To address the complexity of the optimal scheduler, two different heuristic algorithms for the proposed packet scheduler were developed. Numerical results demonstrate that our proposed schedulers achieve better EE/jitter performance for the UE compared to existing state-of-the-art schedulers.

The rest of this chapter is organized as follows: In Section 5.2, a detailed analytical modelling for the delay jitter in different queuing systems is presented. The optimization for the delay jitter of the VoLTE traffic, and its effect on the UE's EE, in LTE multiuser environment is then presented in Section 5.3. Finally, Section 5.4 concludes the chapter.

5.2 Analytical Approximation of Packet Delay Jitter in Simple Queues

5.2.1 Background

The unabated evolution of today's wireless technologies, which support extremely high data rates, becomes evidently the main driver of the current wide spectrum of multimedia services. This spectrum ranges from services such as VoIP, real-time video streaming, social networking, interactive on-line gaming and ends up by the new evolving concept of internet of things (IoT). The emergence of such services over the currently deployed 4G or even the future 5G networks is associated with stringent QoS requirements. One vital QoS metric that substantially affects the end-to-end experience of real-time services is the packet delay jitter.

In literature, various attempts have been carried out either to control the delay jitter [91, 92] or to provide an analytical approximation to it [93, 94]. In [91], Houeto *et al.* developed a jitter-constrained admission control mechanism to provide eventual guarantees on the delay jitter bounds in multi-service ATM networks. Utilizing a different approach, Oklander *et al.* provided an anlytical model for the well known jitter buffer mechanism in [92]. The jitter buffer mechanism is known as a post-processing scheme implemented at the receiver side to compensate for the delay jitter encountered by the packet throughout its network route. In contrast to Houeto's approach, Matragi *et al.* proposed a probabilistic model for estimating the end-to-end jitter of a periodic traffic (by means of estimating the departure process) traversing through multiple nodes in an ATM network [93]. Similarly, Wen *et al.* provided a theoretical evaluation for the packet delay jitter of a real-time service in double queue single server with limited capacity queuing system [94]. The real-time traffic flow was modeled as a Two-state Markov-modulated Bernoulli process (MMBP-2) while the other non real-time flow was modeled as an interrupted Bernoulli process (IBP).

Considering the simplest queuing model with a single flow per node server and single hop for each packet, the packet delay jitter is due to the stochastic nature of both of the packets evolution and the channel quality serving these packets. In this context and to the best of our knowledge, it has been noted that none of the published research has provided a clear mathematical expressions describing the jitter behavior. As a result, this section is devoted to presenting a solid mathematical characterization for the delay jitter of the simple queuing model in different scenarios. The underlying objective is to help derive heuristic algorithms for a jitter-efficient packet scheduler that could be utilized in today's wireless networks.

The rest of this section is organized as follows: The jitter modelling framework is explained in Subsection 5.2.2. Subsections 5.2.3, 5.2.4 and 5.2.5 present the jitter model for the single-flow single-server queuing system under different traffic loads, and packets' interarrival and service time statistics. The numerical validation for the derived models is then provided in Subsection 5.2.6.

5.2.2 Jitter modelling framework

Following [93], we define jitter $\Delta t(k+1,k) = \tau_{k+1,d} - \tau_{k,d}$ as a time difference between delays experienced by two sequential packets indexed as k and $k+1$, where $\tau_{k,d}$ is the queuing delay for the packet indexed k. At this point the values of $\Delta t(k+1,k)$ could be negative, in spite of the standard practice of considering only the absolute value of the jitter. In this subsection, we will be interested in approximating the distribution of $\Delta t(k+1,k)$ and, above all, the approximation of its mean $m_j = \mathcal{E}\{\Delta t(k+1,k)\}$, absolute value mean $m_{|j|} = \mathcal{E}\{|\Delta t(k+1,k)|\}$ and the variance $\sigma_j^2 = \mathcal{E}\left\{(\Delta t(k+1,k))^2\right\} - m_j^2$. In general, the problem is hard to solve analytically. In order to simplify the analysis we consider three modes of queue operation based on the system's utilization factor ρ:

1. underloaded (underutilized) system $\rho \ll 1$
2. critically loaded system $1 - \rho \ll 1$
3. intermediate case

5.2.3 Underutilized queue

In the underutilized system the queue is very shallow, consisting mainly of a single packet. In other words, as soon as a packet gets into the queue, it starts being served. The difference in the delay between two sequential packets in underutilized queue is, thus, defined only by a difference in the service times (and independent of the arrival process), *i.e.,*

$$\Delta t(k+1, k) = \tau_{k+1,s} - \tau_{k,s} \tag{5.1}$$

where $\tau_{k,s}$ is the time needed to serve the k-th packet. As a result, the statistics of $\Delta t(k+1, k)$ is defined by the joint distribution of two sequential service times $p_{2s}(\tau_1, \tau_2)$. Using well known results from statistics [95], the distribution of difference of two random variables is given as:

$$p_j(z) = \int_{-\infty}^{\infty} p_{2s}(\tau_1, \tau_1 + z) d\tau_1 \tag{5.2}$$

In some cases, such as *G/M/1* queues, the service time for different packets are independent which allows for a simpler expression of the jitter distribution:

$$p_j(z) = \int_{-\infty}^{\infty} p_s(\tau_1) p_s(\tau_1 + z) d\tau_1 \tag{5.3}$$

i.e., it depends only on the marginal PDF $p_s(\tau)$ of the service time. It is worth noting here that for deterministic constant service time $p_s(\tau) = \delta(\tau - T_s)$ one immediately obtains $p_j(z) = \delta(z)$, *i.e.,* in the case of a light load there is no jitter introduced by the queue (*i.e.*, arrival process) and the server combined.

5.2.3.1 G/M/1 queue

In this case, the service time is exponentially distributed with the average rate μ packets per second:

$$p_s(\tau) = \mu \exp(-\mu\tau)u(\tau) \tag{5.4}$$

Here $u(\tau)$ is the Heaviside unit step function [96]. Making use of equation (5.3) one obtains:

$$p_j(z) = \mu^2 \int_{-\infty}^{\infty} \exp(-\mu\tau_1)\exp(-\mu\tau_1 - \mu z)u(\tau_1)u(\tau_1 + z)d\tau_1 = \frac{\mu}{2}\exp\left(-\mu|z|\right) \tag{5.5}$$

Using the PDF (5.5) one can obtain the following expressions for the mean, absolute mean and the variance of jitter:

$$m_j = 0,\ m_{|j|} = \frac{1}{\mu},\ \sigma_J^2 = \frac{2}{\mu^2} \tag{5.6}$$

Thus, in order to reduce jitter, one has to increase the service rate μ.

5.2.3.2 Gilbert-Elliot (GE) channel

Let us assume that packets of a fixed length L bits are being served by a Gilbert-Elliot channel with the service rate R_B in the "BAD" state B, and the rate $R_G > R_B > 0$ in the "GOOD" state G. The transition between the states are described by the following transition matrix [63]:

$$\mathbf{T} = \begin{bmatrix} (1-d)P_G + d & (1-d)(1-P_G) \\ (1-d)P_G & (1-d)(1-P_G) + d \end{bmatrix} \tag{5.7}$$

In this model P_G is the probability of the "GOOD" state, while $0 \leq d \leq 1$ defines the correlation properties of the channel. If $d = 0$ the states are changing in an independent manner, while $d = 1$ corresponds to the situation of no transition to another state (constant channel). The service time of a packet in the state G is $\tau_G = L/R_G$, while in the state B is given by $\tau_B = L/R_B$. Considering two sequential packets, one can observe that no jitter will appear if the channel does not change its state (*i.e.,* GG or BB combination appears), while the difference $\tau_G - \tau_B = -\tau_J$ corresponds to the channel changing from B to G and $\tau_B - \tau_G = \tau_J$ corresponds to GB transition. Here

$$\tau_J = \frac{L}{R_B} - \frac{L}{R_G} = \frac{L(R_G - R_B)}{R_G R_B} \tag{5.8}$$

Therefore, the distribution of jitter is given as:

$$\begin{aligned} p_j(z) &= (1 - P_{GB} - P_{BG})\delta(z) + P_{GB}\delta(z - \tau_J) + P_{BG}\delta(z + \tau_J) \\ &= [1 - 2(1 - d)P_G(1 - P_G)]\,\delta(z) + (1-d)P_G(1-P_G)\delta(z-\tau_J) + (1-d)P_G(1-P_G)\delta(z+\tau_J) \end{aligned} \tag{5.9}$$

Using the PDF (5.9) one can obtain the following expressions for the mean, absolute mean and the variance of jitter:

$$m_j = 0, \; m_{|j|} = 2(1 - d)\tau_J P_G(1 - P_G), \; \sigma_J^2 = 2\tau_J^2(1 - d)P_G(1 - P_G) \tag{5.10}$$

It can be seen from (5.10) that the reduction of jitter could be achieved in a number of ways:

- Equalizing the service time in each state of the channel (reduction of τ_J) by the channel inversion [97].
- Increasing speed of a mobile leads to lowering of d, thus, it produces an increased jitter. Converse will reduce jitter.
- Increasing the probability of one state over the other will lead to reduction of jitter. This could also be achieved by channel inversion.

5.2.3.3 Correlated exponential service time

There is no single model for the bivariate exponential distribution, even in the Markov case [63]. However, to investigate the effect of service time correlation, we choose a simple distribution, suggested in [98] for exponentially correlated Markov processes:

$$p_{2s}(\tau_1, \tau_2) = (1-d)\mu^2 \exp\left[-\mu(\tau_1+\tau_2)\right] u(\tau_1)u(\tau_2) + d\mu \exp(-\mu\tau_1)u(\tau_1)\delta(\tau_1-\tau_2) \tag{5.11}$$

where τ_1 and τ_2 represent two consecutive and correlated service times, and d is the correlation parameter as in (5.7). Simple algebra results into the following expression for the jitter distribution:

$$p_j(z) = (1-d)\frac{\mu}{2}\exp(-\mu|z|) + d\delta(z) \tag{5.12}$$

Therefore, the mean and the variance are given as:

$$m_j = 0,\ m_{|j|} = (1-d)\frac{1}{\mu},\ \sigma_J^2 = (1-d)\frac{2}{\mu^2} \tag{5.13}$$

Not surprisingly, this example confirms the conclusion of subsection 5.2.3.2 that correlation suppresses jitter.

5.2.4 Heavy loaded queue

In the case of a heavy load $1-\rho \ll 1$ the queue is almost never empty. Let us assume that the k-th packet starts being served at time instant $t = 0$. Let $t_0 < 0$ be a time of arrival of the k-th packet to the queue. According our assumption of a long queue, the time $t_0 + \tau_A$ of arrival of the $k+1$-th packet, also precedes $t = 0$, *i.e.,* this packet is in the queue by the beginning of service time of the k-th packet. Here τ_A is the interarrival time. The k-th packet will be served at the time instant $t = \tau_{k,s}$, while the $k+1$-th packet will be served at the time instant $t = \tau_{k,s} + \tau_{k+1,s}$. Therefore, the jitter $\Delta t(k+1,k)$ could be evaluated as:

$$\Delta t(k+1,k) = (\tau_{k+1,s} + \tau_{k,s} - t_0 - \tau_A) - (\tau_{k,s} - t_0) = \tau_{k+1,s} - \tau_A \tag{5.14}$$

Thus, the jitter is now a function of the arrival and service time distributions.

5.2.4.1 M/M/1 queue

Let us assume that packets are arriving at a rate λ and being served at the rate $\mu > \lambda = \mu\rho$. In this case both interarrival time and service time are exponentially distributed, *i.e.,*

$$p_A(\tau) = \lambda \exp(-\lambda\tau)u(\tau),\ p_s(\tau) = \mu \exp(-\mu\tau)u(\tau) \tag{5.15}$$

Following equation (5.2) one can derive the following distribution of the jitter $\Delta t(k+1,k)$:

$$p_j(z) = \frac{\mu\lambda}{\mu+\lambda}\begin{cases} \exp(-\lambda z) & \text{if } z \geq 0 \\ \exp(\mu z) & \text{if } z < 0 \end{cases} \tag{5.16}$$

The corresponding mean, absolute mean and the variance are given by:

$$m_j = \frac{\mu-\lambda}{\lambda\mu} = \frac{1}{\mu}\frac{1-\rho}{\rho} \approx 0,\ m_{|j|} = \frac{1}{\mu}\frac{1+\rho^2}{(1+\rho)\rho} \approx \frac{1}{\mu},\ \sigma_j^2 = \frac{1}{\mu^2}\frac{1+\rho^2}{\rho^2} \approx \frac{2}{\mu^2} \tag{5.17}$$

It can be seen that for the heavy loaded system, the impact on jitter could be similar to that of the light loaded system. It is important to note that this is true only in the case of *M/M/1* queue.

5.2.4.2 D/M/1 queue

In the case of deterministic interarrival intervals $p_A(\tau) = \delta(\tau - T_A)$ where $T_A = 1/\lambda = 1/\rho\mu$ is the interarrival interval. Therefore, the distribution of jitter is just a shifted version of the service time distribution:

$$p_j(z) = \mu \exp\left[-\mu(z+T_A)\right] u(z+T_A) \tag{5.18}$$

The load of the system $\rho = 1/T_A\mu$. The corresponding mean, absolute mean and the variance are given by:

$$m_j = T_A - \frac{1}{\mu} = \frac{1-\rho}{\rho}\frac{1}{\mu} \approx 0,\ m_{|j|} = \frac{2\rho\exp(-1/\rho) + (1-\rho)}{\rho}\frac{1}{\mu},\ \sigma_j^2 = \frac{2}{\mu^2} \tag{5.19}$$

5.2.4.3 GE Channel

Let the server be described by a set of M states with probabilities P_m and service rate $R_m > 0$ in the m-th state. The density of the service time $\tau_{s,m} = L/R_m$ is given by:

$$p_s(\tau) = \sum_{m=1}^{M} P_m \delta(\tau - \tau_{s,m}) \tag{5.20}$$

Furthermore, assuming that $p_A(\tau)u(\tau)$ is the PDF of the interarrival time, the distribution of the jitter could be expressed as:

$$p_j(z) = \sum_{m=1}^{M} P_m p_A(-z + \tau_{s,m}) u(-z + \tau_{s,m}) \tag{5.21}$$

It is worth noting at this stage that the correlation of the underlying Markov channel does not affect the jitter distribution, since a service time for a later packet includes the whole service time of the preceding packet.

5.2.5 Bridging case

In the intermediate case when the incoming traffic does not always keep queue occupied we suggest the following *approximation* to the distribution of jitter. Let P_0 be the probability of the empty queue. In this case

$$p_j(z) = P_0 p_{j,l}(z) + (1 - P_0) p_{j,h}(z) \tag{5.22}$$

Here $p_{j,l}(z)$ is the jitter PDF in the case of low load while $p_{j,h}(z)$ is the PDF of the jitter in the case of high load. The exact value of P_0 is known in some cases of classical queues. For example, for *M/M/1* queue $P_0 = 1 - \rho = 1 - \lambda/\mu$, therefore one obtains

$$p_j(z) = (1 - \rho)\frac{\mu}{2}\exp(-\mu|z|) + \frac{\rho^2}{1+\rho}\begin{cases} \exp(-\lambda z) & \text{if } z > 0 \\ \exp(\mu z) & \text{if } z < 0 \end{cases} \tag{5.23}$$

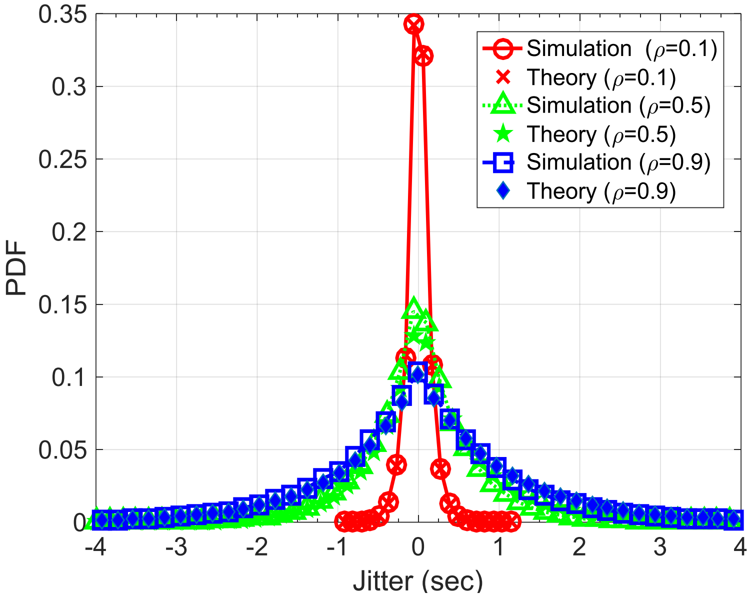


Figure 5.1: Packet delay jitter evaluation for the *M/M/1* queue

Other approximation could be obtained in the very much same manner. In many sources it is suggested that P_0 is substituted by $1 - \rho$ in the case of an arbitrary sources.

5.2.6 Numerical Simulations

This subsection presents MATLAB numerical evaluation for the packet delay jitter models developed in the previous subsections. Two types of traffic are considered. First, the Poisson traffic is considered in the case of *M/M/1*, *M/GE/1* and *M/ExpCorr/1* (*i.e.,* exponentially correlated service time case) queues. The second type is the deterministic traffic, considered in the *D/M/1* queue, where packet arrivals are assumed to be periodic. In both cases the packet arrival rate (*i.e.,* λ) is kept constant at 10^3 packets/sec while the average service time (*i.e.,* $1/\mu$) in msec takes the values $\{0.1,\, 0.5,\, 0.9\}$ to study the system at the light, moderately and heavy loaded queue modes (*i.e.,* $\rho = \{0.1,\, 0.5,\, 0.9\}$ value), respectively.

For the *M/M/1* queue, the results depicted in Fig. 5.1 show a very good agreement for the derived jitter models, described in (5.5), (5.16) and (5.23), with the simulation results at the three queue modes. In addition, it is shown that increasing the queue load ρ (*i.e.,* packets' queuing delay) increases the packet delay jitter variance. In contrast to *M/M/1*, when setting the packets' interarrival time to a constant value (*i.e.,* T_A= 1msec),

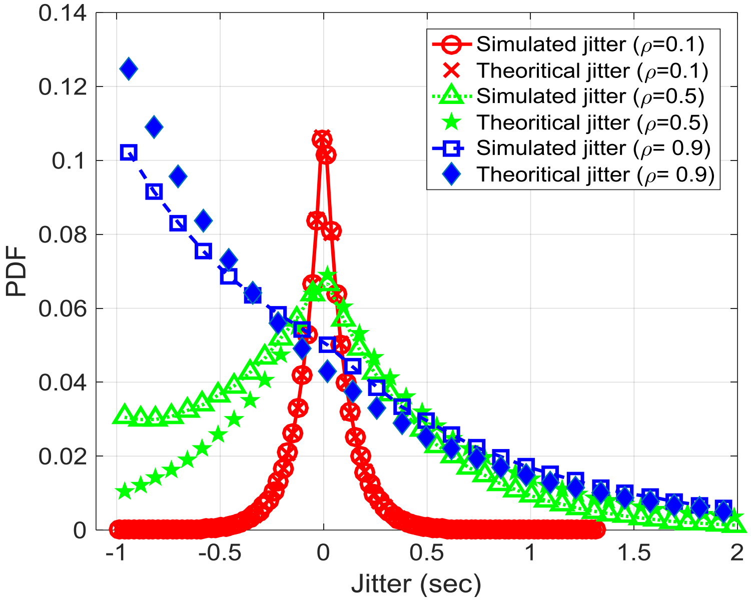


Figure 5.2: Packet delay jitter evaluation for the *D/M/1* queue

the *D/M/1* queue delay jitter was found as shown in Fig. 5.2. The results illustrate that the analytical jitter models developed in (5.5), (5.18) and (5.23) perfectly captures the trends of the simulation results, however, with deviation in some of the points.

To investigate the effect of packets' service time correlation on the jitter performance of the *M/M/1* queue with light load (*i.e.,* ρ= 0.1), the Markov model for exponentially correlated process developed in [98] is utilized to set exponentially correlated service times for the generated packets. The results presented in Fig. 5.3 show the performance of the jitter model developed in (5.12) in comparison with the simulation results, and the effect of changing the correlation parameter d on both. It is noted that the analytical model shows a good representation for the simulation results especially at low correlation. On the other hand, increasing the packets' service time correlation sharpens the delay jitter distribution around zero (*i.e.,* the delta term in equation (5.12)). This behaviour pertains to the fact that high correlation for the service time (*i.e.,* large value for d), while knowing the service time to be the sole factor affecting the jitter in case of low loaded system, implies high correlation for the packet delay, and hence, low packet delay jitter.

For the case of GE server, and *M/GE/1* queue, the analytical jitter models derived in (5.9) and (5.21) are numerically evaluated in Fig. 5.4 and Fig. 5.5 for low and high load

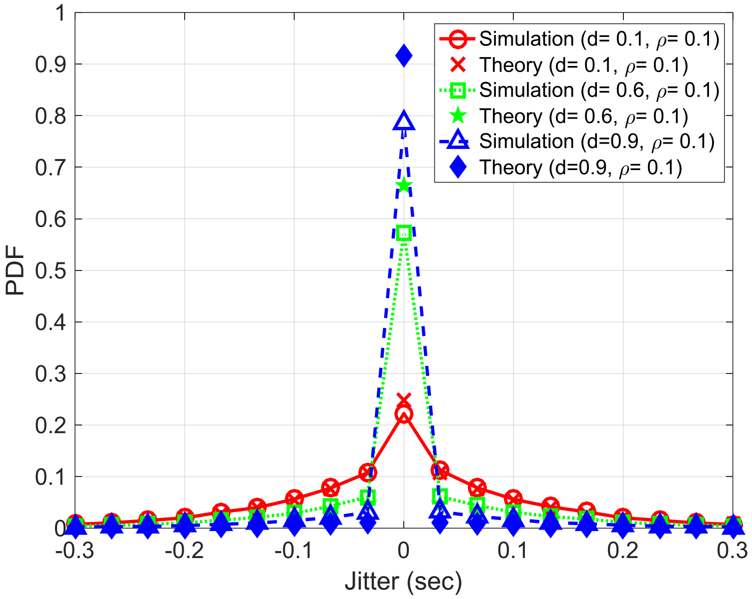


Figure 5.3: Packet delay jitter evaluation for the *M/ExpCorr/1* queue

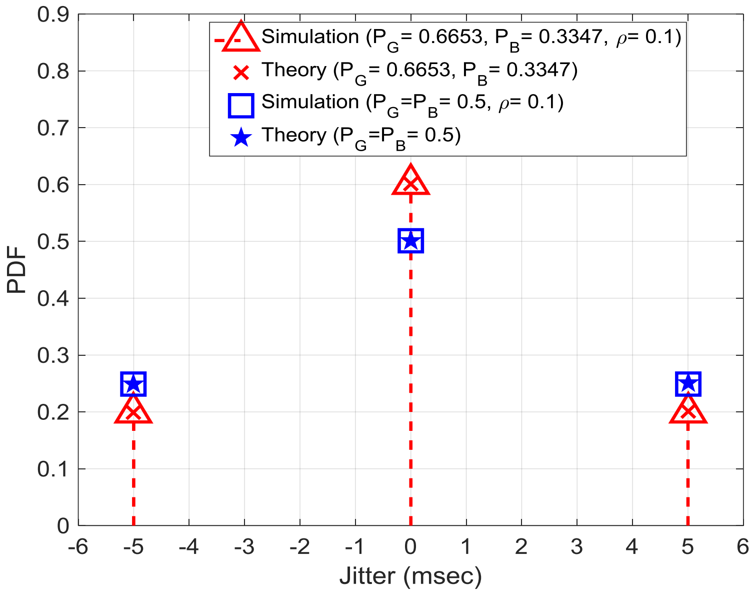


Figure 5.4: Packet delay jitter evaluation for underutilized *M/GE/1* queue

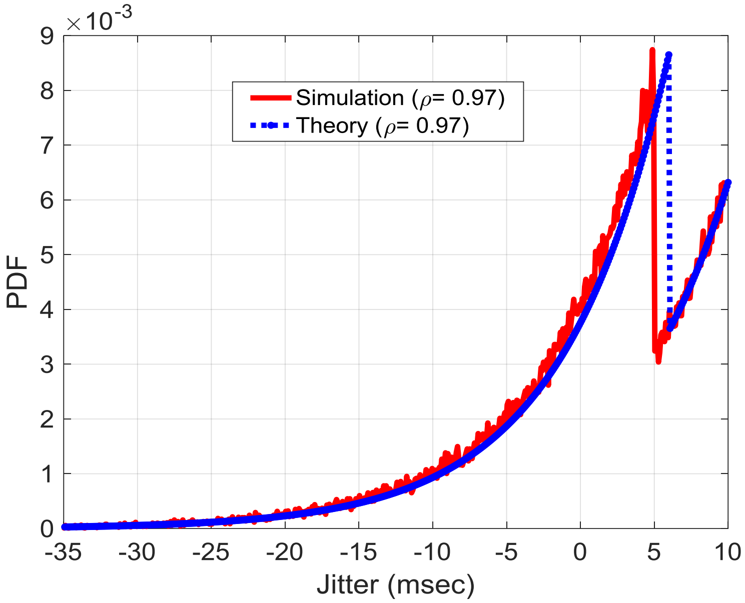


Figure 5.5: Packet delay jitter evaluation for heavy loaded *M/GE/1* queue

cases, respectively. In the case of underutilized queue, Fig. 5.4 results show the discrete distribution for the jitter, as in (5.9), at the values 0 and $\pm\tau_j$. It should be noted that the GE model was set to have τ_G= 5msec and τ_B= 10msec for both cases. Moreover, the results shown in Fig. 5.4 demonstrate a subtle match between the theoretical model and simulation at different probabilities for the good and bad states. Finally, at high queue load, the jitter model suggested in (5.21) for the *M/GE/1* queue highly represents a real simulated queue as illustrated in Fig. 5.5.

5.3 Investigating the Energy-Efficiency/Delay Jitter Trade-off for VoLTE Traffic in LTE Downlink

5.3.1 Background

The energy efficient wireless communications, sometimes called green wireless communications, is currently receiving remarkable attention in both of the research and the industrial domains [72]. This is due to its huge economic and environmental benefits. However, from the user equipment (UE) side, smart cell phones with prolonged battery lifetime per charge represent the direct result of designing an energy efficient wireless system. This is a research problem towards which extensive efforts have been spent due to the current consumer's urgent need for smart cell phones with extended battery lifetime.

In addition to the battery lifetime, the user strictly judges the quality of service (QoS) provided by the network operator especially for those services which are time sensitive. As a result, today's LTE networks supporting data demanding real-time services set high standards for the QoS levels. Delay jitter of data packets is known to be an essential QoS metric for real-time traffic which affects the users' quality of experience. One important example of a real-time service, which we exclusively consider in this paper, is the emerging Voice over LTE (VoLTE) technology [99]. The VoLTE system allows carrying voice calls over the LTE network's data bearers, just as data packets, instead of relaying over the traditional voice network (*i.e.*, 3G circuit-switched voice call, and its extension over the High Speed Packet Access (HSPA) network [100]). As a result, users will take advantage of the fast LTE speeds to establish high quality (*i.e.*, low delay and jitter) voice, and video, calls with faster set-up times compared to regular calls.

In the literature, few works [69, 74, 78, 79] have addressed the UE's energy efficiency (EE) in the LTE downlink which utilizes the OFDMA scheme. EE is known to be affected by the UE's radio frequency (RF) and baseband circuits' energy consumption for receiving and decoding the data packets carried by the LTE's *physical downlink shared channel* (PDSCH). The idea of improving the UE's EE was first introduced by the 3GPP's LTE discontinuous reception mechanism (DRX) [79]. The mechanism was designed to conserve the LTE's UE battery energy by controlling the circuit power operation in an ON-OFF manner while maintaining certain bounds for the packet delays. The framework provided the optimum criteria for selecting the DRX mode and parameters, in both of the LTE network's connected and idle states, for different types of applications. In [74], Gupta *et al.* proposed a framework for jointly optimizing the base station and the UE's EE, by optimizing the number of downlink transmission time slots, in the OFDMA system subject to non guaranteed bit rate (non GBR) and delay constraints. Based on the LTE's DRX mechanism developed in [79], Chu *et al.* proposed a green resource allocation scheme in [78] that directly optimizes the UE's EE in OFDMA systems subject to fairness among users. The problem was formulated as a nonlinear integer programming problem for minimizing the number of wake-up transmission time intervals (TTIs) during which the UE turns ON its receiver circuit. Unlike the framework presented in [78], Hammad *et al.* proposed in [69] a cloud-based predictive scheduling model to optimize the receiver's circuit operation in longer time horizons. The scheduling framework was based on a practical power consumption model for the LTE's UE, proposed in [82], and the effective bandwidth theory [101] for ultimately and fairly satisfying the UE's buffers QoS requirements.

Despite their importance, none of the reported works highlighted above considered the packet delay jitter as a crucial QoS metric for real-time traffic flows when studying the UE's receiver EE. Instead, the packet delivery delay and data rate were the only measures commonly taken for meeting the real-time performance. Hence, we put more emphasis on the delay jitter in our framework and present it as an objective besides the EE while also maintaining the packet delivery delay time threshold for VoLTE traffic. The main contributions of this section are summarized as follows. First, a multi-objective optimization problem for the UE's EE and the delay jitter subject to delay constraints for VoLTE traffic in LTE downlink is presented. Second, two low complexity heuristic algorithms are proposed for solving the optimization problem due to its inherent intractability. Finally, the obtained results reveal an essential trade-off between the UE's EE and the packet delay jitter.

The rest of the section is organized as follows. Subsection 5.3.2 introduces the system model and the resource allocation problem formulation. The heuristic algorithms proposed to solve the optimal formulation are described in Subsection 5.3.3. Simulation results are provided in Subsection 5.3.4.

5.3.2 System Model and Problem Formulation

We consider a single cell of LTE downlink multiuser system as shown in Fig 5.6. The evolved Node B (eNB) transmits single VoLTE connection to each of the K UEs located within the cell coverage. As defined by the LTE standard, the overall cell bandwidth is divided equally, during each transmission time interval (TTI), into N resource blocks (RBs). Each RB consists of 12 adjacent subcarriers with a total bandwidth of 180kHz. In light of the FDD LTE frame type 1, the radio frame encompasses 10 TTIs (*i.e.*, subframes) each of which has a duration of T_s= 1msec.

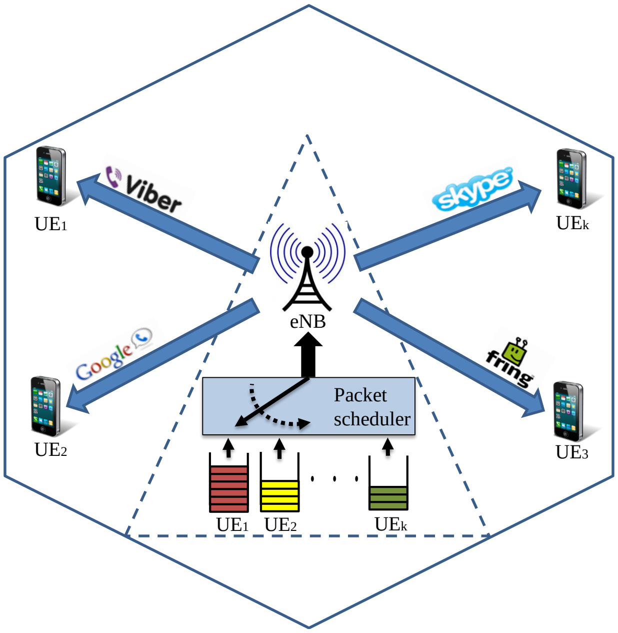


Figure 5.6: System model

The eNB's central packet scheduler targets to fairly allocate the required number of RBs to each of the K UEs within a time horizon of M TTIs to optimize both the UE's EE and packet delay jitter performance subject to the VoLTE packet delay budget (*i.e.*, 100msec). We assume a scheduling time granularity of 10msec (*i.e.*, one LTE frame), and hence, an accurate channel state information (CSI) for each UE is available at the eNB for the whole frame (*i.e.*, M= 10TTIs). To address the gauranteed delay service for the VoLTE UEs, one popular policy known as the largest weighted delay first (LWDF) [102] is commonly employed. The LWDF utilizes the head-of-line (HOL) packet delay for each UE buffer ($D_{HOL,k}$) by defining its metric function over single RB as follows:

$$X_{k,n}^{LWDF}(m) = \alpha_k \, D_{HOL,k} \tag{5.24}$$

where $X_{k,n}^{LWDF}(m)$ is the LWDF metric function of UE k over the RB n within TTI m, and α_k is a UE distinguishing parameter which determines the weight of its metric function as:

$$\alpha_k = -\frac{\log \delta_k}{D_k^{\max}} \tag{5.25}$$

where δ_k is packet loss rate threshold of UE k, and $D_k^{\max}$ is the packet delay budget of UE

k.

In our framework, the optimal scheduler ideally targets to schedule all the queued packets for all UEs (*i.e.*, not just the head packets) on the M TTIs horizon. In this context, we noted a potential shortcoming of the LWDF policy which hinders its utilization in our optimal framework. That is, looking only at the $D_{HOL,k}$ results in scheduling all packets belonging to the queue with the largest $D_{HOL,k}$ before other imminently expiring packets belonging to other queues with smaller $D_{HOL,k}$ values. As a result, some users will perceive good delay performance while others will not. To mitigate this issue and achieve fair scheduling on the packet level in terms of the attained average packet delay per UE, our scheduler proposes another policy which fits our optimal framework namely fair LWDF (F-LWDF). The F-LWDF policy metric function is expressed as follows:

$$X_{k,a}^{F-LWDF}(m) = W_k^a(m) \tag{5.26}$$

where $W_k^a(m)$ is the waiting time of the packet with index a in the queue of UE k up to TTI m.

In contrast to the LWDF, our F-LWDF policy provisions the waiting time in the queue for each packet in each UE buffer. Consequently, packets with the closest deadline expiration are given high priority scheduling regardless of which UE buffer they belong to. This way, our optimal scheme avoids the fairness issue of the LWDF policy.

On the other hand, optimizing the UE's EE is achieved by increasing the *bits-per-joule* metric. In other words, reducing the energy consumed by the UE's receiver circuit to receive and decode the same number of VoLTE packets. That reduction is attained by minimizing the number of TTIs during which the UE's receiver circuit is in the wake-up state [78, 79] for receiving the VoLTE packets. Based on the model developed in [82] and our pervious analysis in [69], the energy consumed by the UE's receiver circuit within a single TTI is given by:

$$E_k = T_s \left(m_{idle} \, P_{idle} + \overline{m_{idle}} \, (P_{on} + P_{rx}) \right) \text{ Joules} \tag{5.27}$$

where m_{idle} is a binary logic control variable to identify the UE's operation state (*i.e.*, idle or active), P_{idle} is the idle state power consumption (equal to 0.5W [82]), P_{on} is the active state power consumption (equal to 1.53W [82]), and P_{rx} is the base power consumed by the receiver chain during the active state (equal to 0.42W [82]).

The packet delay jitter is defined as the time difference between two successive packet delays for each UE buffer [93]. This can be expressed for two successive packets indexed as $a-1$ and a as follows:

$$\Delta t_k(a-1,a) = D_{a,k} - D_{a-1,k} \tag{5.28}$$

where $D_{a,k}$ is the queuing delay for packet a of UE k.

Substituting for $D_{a,k}$ and $D_{a-1,k}$ with their corresponding arrival and departure times, the jitter could be further expressed in terms of the packets' interarrival and interdeparture times as follows:

$$\begin{aligned} \Delta t_k(a-1,a) &= \left(t^D_{a,k} - t^A_{a,k}\right) - \left(t^D_{a-1,k} - t^A_{a-1,k}\right) \\ &= \left(t^D_{a,k} - t^D_{a-1,k}\right) - \left(t^A_{a,k} - t^A_{a-1,k}\right) \\ &= \tau^D_k(a-1,a) - \tau^A_k(a-1,a) \end{aligned} \tag{5.29}$$

where $t^A_{a,k}$ is the arrival time of the packet with index a to the buffer of UE k, $t^D_{a,k}$ is the departure time of the packet with index a from the buffer of UE k, $\tau^A_k(a-1,a)$ is the interarrival time between packets a and $a-1$ for UE k, and similarly $\tau^D_k(a-1,a)$ is the interdeparture time.

In sum, for each UE, our proposed scheduler strives to minimize the circuit energy consumption and packet delay jitter, expressed in (5.27) and (5.28) (or 5.29), respectively, subject to the delay constraint provisioned by the fair policy expressed in (5.26). Thus, the resource allocation optimization problem which addresses the previous objectives and constraints can be formulated as follows:

Min

$$\sum_{k=1}^{K} \sum_{a=1}^{A_k(m_o)} \frac{X^{F-LWDF}_{k,a}(m_o)}{D^{\max}_k} \sum_{m=m_o}^{m_o+M-1} \sum_{n=1}^{N} \left(\begin{array}{l} T_s P_c \, \Phi_k(m) + \\ \Delta t_k(a-1, a\,|m) \Psi^a_k(m,n) \end{array} \right) \tag{5.30a}$$

Subject to

$$\sum_{a=1}^{A_k(m_o)} \sum_{m=m_o}^{m_o+M-1} \sum_{n=1}^{N} B_k(m,n)\Psi_k^a(m,n) \geq \omega_k(m_o), \ \forall k \tag{5.30b}$$

$$D_k^a(m)\Psi_k^a(m,n) \leq D_k^{\max}, \ \forall k,m,n,a \tag{5.30c}$$

$$\sum_{k=1}^{K} \Psi_k^a(m,n) \leq 1, \quad \forall m,n,a \tag{5.30d}$$

$$\Psi_k^a(m,n) - \Phi_k(m) \leq 0, \ \forall k,m,n,a \tag{5.30e}$$

$$\begin{aligned} &\left(D_k^a(m) + t_{a,k}^A\right) \Psi_k^a(m,n) \leq \\ &\left(D_k^{a+1}(m) + t_{a+1,k}^A\right) \Psi_k^{a+1}(m,n), \ \forall k,a,m \end{aligned} \tag{5.30f}$$

where $A_k(m_o)$ is the total number of queued packets in the buffer of UE k at TTI m_o, P_c is the UE's constant power consumption within each TTI (*i.e.*, P_{idle} in the idle state or P_{on}+P_{rx} in the active state), $\Phi_k(m)$ is a binary decision variable which indicates whether the receiver circuit for UE k is turned on during TTI m or not, $\Delta t_k(a-1, a\,|m)$ is the packet delay jitter for packet a of UE k if scheduled at TTI m, $\Psi_k^a(m,n)$ is a binary decision variable which indicates whether packet a of UE k has been assigned to RB n during TTI m or not, $B_k(m,n)$ is the number of allocated bits for user k over RB n during TTI m, $\omega_k(m_o)$ is the total buffer size (in bits) for UE k at the first TTI (*i.e.*, m_o) of the current scheduling horizon, and $D_k^a(m)$ is the attained delay of packet a for UE k if scheduled at TTI m.

It can be seen that the cost function defined in (5.30a) is a multi-objective function which simultaneously minimizes the total energy consumed by each UE receiver circuit and the packet delay jitter within a time horizon of single LTE frame (*i.e.*, M= 10). As previously explained, the optimization for each UE is weighted by the F-LWDF metric, however, normalized by the packet delay budget. The first constraint in (5.30b) ensures that all packets queued in each UE buffer until the beginning of the current scheduling horizon, at TTI m_o, is completely scheduled. The second constraint in (5.30c) sets a bound to the scheduling of each packet in time with respect to the VoLTE packet delay budget. The following constraint in (5.30d) restricts the allocation of each RB, within single TTI, to only single UE to avoid intra-cell interference. The constraint in (5.30e) penalizes the cost function in each TTI with P_c if any UE is configured to be in the active state even if receiving over single RB. The final constraint in (5.30f) ensures causal departure times (*i.e.*, *first-in-first-out*) for the packets with respect to their arrival times.

The formulation in (5.30) might be unfeasible in situations where the UE's channel capacities are not sufficient to accommodate all packets waiting in the buffers for all UEs. Hence, an alternative practical formulation should consider two important factors. First, the channel capacity limitation for some UEs. Second, the priority of each UE packet in case the total number of packets for all UEs exceeds the number of available RBs. To meet these conditions, we adopt the penalty method [87] to relax the constraint in (5.30b). The new unconstrained formulation can be written as follows:

Min

$$\sum_{k=1}^{K}\sum_{a=1}^{A_k(m_o)}\frac{X_{k,a}^{F-LWDF}(m_o)}{D_k^{\max}}\left\{\sum_{m=m_o}^{m_o+M-1}\sum_{n=1}^{N}\left(\begin{array}{l}T_sP_c\,\Phi_k(m)+\\ \Delta t_k(a-1,a\,|m)\Psi_k^a(m,n)\end{array}\right)\right.$$
$$\left.+\left(\omega_k(m_o)-\sum_{m=m_o}^{m_o+M-1}\sum_{n=1}^{N}B_k(m,n)\Psi_k^a(m,n)\right)\right\} \tag{5.31a}$$

Subject to

$$\sum_{a=1}^{A_k(m_o)}\sum_{m=m_o}^{m_o+M-1}\sum_{n=1}^{N}B_k(m,n)\Psi_k^a(m,n)\le\omega_k(m_o),\ \forall k \tag{5.31b}$$

$$(5.30c),(5.30d),(5.30e),\text{ and}(5.30f) \tag{5.31c}$$

The new term, commonly known as the penalty function, added to the cost function in (5.31a) allows the scheduler taking decisions to schedule as much packets as possible for all UEs based on the normalized F-LWDF weight for each UE packet.

Despite the fact that the formulation in (5.31) addresses the feasibility issue inherent in that of (5.30), it retains a discrete time stochastic nature which complicates its solution approach. In particular, the optimization of the jitter term $\Delta t_k(a-1,a\,|m)\Psi_k^a(m,n)$ has a huge solution space due to the recursive dependency between the jitter of the A_k packets for each UE. Although constrained Markov decision process (MDP) formulation is one way to solve such type of problems, the solution of MDPs is known to suffer from the dimensionality problem [103]. In (5.31), that dimensionality is function of K, $A_k(m_o)$, and M. As a result, we propose two different low complexity heuristic algorithms in the following subsection to solve the problem in (5.31).

5.3.3 Heuristic Algorithms

In this subsection, two computationally efficient heuristic schemes are proposed for solving the posited optimal energy and jitter efficient VoLTE scheduling problem formulated in (5.31). It is imperative to point out that in the heuristic domain, for both schemes, the fair delay-aware scheduling for the VoLTE users can be easily achieved using the LWDF policy. This is in contrast to the F-LWDF policy that was designed for the optimal model in (5.31). The reason is that the heuristic algorithm, as will be explained in the next paragraphs, schedules a single packet per each UE at a time. Thus, both of the proposed schemes use the same LWDF policy for prioritizing UEs. In particular, in each iteration, the proposed algorithms schedule only the head packet of each UE (*i.e.*, one at a time). The head packet with the largest LWDF metric value (*i.e.*, closest expiration deadline) goes first until the last head packet having the least metric value. The algorithms continues, in iterations, in the fashion until all UEs buffers are empty or all RBs become allocated.

The first heuristic algorithm is explained in Table 5.1. The main strategy adopted to optimize the packet delay jitter is allocating the best possible RB(s) which minimize the difference between the packets interarrival and interdeparture times as highlighted in (5.29). So for each UE packet, the RBs selected are those which minimize the absolute difference between the average interarrival and the average interdeparture times of the past scheduled packets for the same UE. As explained in the above paragraph, the main WHILE loop in line 3 keeps iterating the algorithm until one of the stopping conditions is valid. It shoud be noted that the function $isempty(\mathcal{R})$ gives logical output 1 if the matrix $\mathcal{R}$ contains at least one empty RB allocation. Prioritizing the UEs head packets based on the LWDF policy is then done in lines 4 and 5. Based on the priority set, the algorithm allocates suitable resources for each UE head packet, one at a time, using the FOR loop structure in line 6. For each UE head packet, the algorithm calculates the absolute interarrival and interdeparture times for the past scheduled packets (*i.e.*, stored in the set A_k^*) in line 8. In line 9, the specific RBs (*i.e.*, N^*) and their corresponding TTIs (*i.e.*, M^*) which satisfy

Table 5.1: Heuristic Algorithm 1

```
Require: $K$, $M$, $N$
Initialize emtpy RB allocations matrix $\mathcal{R}$
while $(isempty(\mathcal{R})$ AND $A_k \neq 0,\ \forall\, k)$ do
    Calculate $D_{HOL,k}\ \forall\, k$
    Idx = Sort($D_{HOL,k}$,'descend')
    for $i = 1$ to $K$ do
        Set $k$ = Idx($i$)
        Calculate $\left|\tau_k^A(a-1,a)\right|$, $\left|\tau_k^D(a-1,a))\right|$, $\forall\, a \in A_k^*$
        Find $N^*$, $M^*$to satisfy FIFO
        Update $N^*$, $M^*$based on $D_k^{\max}$
        Sort_EE($N^*$)
        if length($A_k^*$) == 0 then
            Find $\Psi_k^a(m,n)$, $a = HOL_k$, $n \in N^*$, $m \in M^*$
            Update $\mathcal{R}$, $A_k^*$, $A_k$
        else if length($A_k^*$) == 1 then
            Calculate $\left|\Delta t_k(A_k^*, a\,|m)\right|$, $\forall n \in N^*$, $m \in M^*$, $a = HOL_k$
            Sort ($N^*$, $\left|\Delta t_k(A_k^*, a\,|m)\right|$,'ascend')
            Update $M^*$
            Find $\Psi_k^a(m,n)$, $a = HOL_k$, $n \in N^*$, $m \in M^*$
            Update $\mathcal{R}$, $A_k^*$, $A_k$
        else
            Calculate $\left|\tau_k^A(A_{k,last}^*, a)\right|$, $a = HOL_k$
            Calculate $\left|\tau_k^D(A_{k,last}^*, a\,|m)\right|$, $a = HOL_k$, $m \in M^*$
            Sort_JITTER1($N^*$)
            Update $M^*$
            Find $\Psi_k^a(m,n)$, $a = HOL_k$, $n \in N^*$, $m \in M^*$
            Update $\mathcal{R}$, $A_k^*$, $A_k$
        end if
    end for
end while
```

the FIFO constraint in (5.30f) are determined. The RBs and TTIs sets are then updated in line 10 after dropping those which violate the delay constraint in (5.30c). The updated RBs in the set N^* are then sorted in descending order of their capacity in line 11, the arrangement that leads to EE scheduling as proposed in [69]. Optimizing the packet delay jitter is then achieved by re-sorting the RBs of line 11 based on the history of the scheduled packets. This essentially goes through three possible cases. The first case, that is addressed in lines 12-14, occurs only once at the beginning of the UE connection where no history for scheduled packets exits. In this case no jitter optimization is needed, and the algorithm finds the RBs based on their order in line 11. The algorithm then updates the RB allocation matrix $\mathcal{R}$, the set of scheduled packets indexes A_k^* and the buffer size A_k. The second case, in lines 15-20, occurs when only one packet exist in the A_k^*. In this case, the algorithm calculates the attained jitter (line 16) for the current UE head packet at each TTI for the RBs sorted in line 11 with respect only to the delay of the scheduled packet (*i.e.*, stored in A_k^*). Based on the jitter calculations, the RBs in N^* are re-sorted (line 17) in an ascending order. The algorithm then updates $\mathcal{R}$, A_k^* and A_k. Finally, the last case (lines 21-27) resembles the steady state case where the UE has a history of scheduled packets (*i.e.*, 2 packets or more). In this case, and according to the setting in equation (5.29), the algorithm allocates the best RB(s) for the current UE head packet (*i.e.*, the packet awaiting scheduling) such that the absolute difference between the means of interarrival and interdeparture times is minimum. This is done by first calculating the absolute interarrival time (line 22) between the current UE head packet and the last packet stored in A_k^* (*i.e.*, $A_{k,last}^*$). Then, the possible interdeparture times between the current packet and $A_{k,last}^*$ at each of the available RBs in N^* across the TTIs stored in M^* are calculated in line 23. At each calculated interdeparture time, the function **Sort_{JITTER1}** concatenates that value and the interarrival time to those of the history packets, calculated in line 8. The function, then, calculates the absolute difference between the means of interarrival and interdeparture times. The same calculation is repeated at all RBs in N^* that belong to TTIs M^*. Based on these calculations, the function **Sort_{JITTER1}** re-sorts N^* (*i.e.*, initially sorted in line 11) in an ascending order. The algorithm, then, finds the best RBs from the sorted list and updates $\mathcal{R}$, A_k^* and A_k. The algorithm keeps iterating in the same fashion for the head packets (one at a time) of the sorted UEs in line 5 until one of the WHILE stopping conditions becomes

Table 5.2: Heuristic Algorithm 2

```
1:  Require: $K$, $M$, $N$
2:  Initialize emtpy RB allocations matrix $\mathcal{R}$
3:  while $(isempty(\mathcal{R})$ AND $A_k \neq 0,\ \forall\, k)$ do
4:      Calculate $D_{HOL,k}\ \forall\, k$
5:      Idx $=$ Sort$(D_{HOL,k},' descend')$
6:      for $i = 1$ to $K$ do
7:          Set $k =$ Idx$(i)$
8:          Find $N^*$, $M^*$to satisfy FIFO
9:          Update $N^*$, $M^*$based on $D_k^{\max}$
10:         Sort_EE$(N^*)$
11:         if length$(A_k^*) == 0$ then
12:             Find $\Psi_k^a(m, n)$, $a = HOL_k$, $n \in N^*$, $m \in M^*$
13:             Update $\mathcal{R}$, $A_k^*$, $A_k$
14:         else
15:             Calculate $\left|\Delta t_k(A^*_{k,last}, a\,|m)\right|$, $\forall n \in N^*$, $m \in M^*$, $a = HOL_k$
16:             Sort_JITTER2$(N^*)$
17:             Update $M^*$
18:             Find $\Psi_k^a(m, n)$, $a = HOL_k$, $n \in N^*$, $m \in M^*$
19:             Update $\mathcal{R}$, $A_k^*$, $A_k$
20:         end if
21:     end for
22: end while
```

valid.

In contrast to the first algorithm which considers the whole history of the scheduled packets in optimizing the jitter of the future packets, the second proposed algorithm in Table 5.2 only provisions the delay of the last scheduled packet to optimize the jitter of the following packet in an instantaneous manner. Thus, only two cases exist when optimizing the packet delay jitter. The first case, expressed in lines 11-13, is similar to that of algorithm 1 (*i.e.*, lines 12-14) where no jitter is existing and only EE scheduling takes place. The second case, addressed in lines 14-19, utilizes the function **Sort$_{\textbf{JITTER2}}$** which re-sorts RBs in an ascending order in reference to the attained jitter between the delays of the current head packet and the last recorded packet in A_k^* (*i.e.*, $A^*_{k,last}$) at each TTI in M^*.

According to the previous discussion of the two algorithms, it could be seen that algorithm 1 has higher complexity than algorithm 2 in terms of the memory requirement and jitter optimization which both depend on the whole history of scheduled packets. However, the Big O notation is known to be the formal representation for the computational complexity. For algorithm 1, the complexity for the sorting in line 5 is O(Klog$K$). Having a single VoLTE packet arrival every 20msec during the ON period with an average duration of 3sec, each of the interarrival and interdeparture times calculation in line 8 has the order of O($25T_{tot,k}$), where $T_{tot,k}$ is the total duration for UE-k VoLTE connection and $25T_{tot,k}$ is the approximate average number of VoLTE packets generated for UE-k throughout its connection. The searching operation in line 9, as well as in line 10, has the complexity of O(MN). Similar to line 5, the complexity of the sorting function in line 11 is O(MNlog(MN)). Lastly, the worst complexity for the if-statement in line 12 corresponds to the third case (*i.e.*, lines 22-27) and is equal to O(MN), MN O($25T_{tot,k}$)+O(MNlog(MN)) and O(MN) for lines 23, 24 and 26, respectively. Since $T_{tot,k} >> NM$log(NM). Therefore, the asymptotic upper limit for the complexity of algorithm 1 is approximately KMN O($25T_{tot,k}$), where the factor K corresponds to the FOR loop in line 6. Similarly, inspecting the second algorithm in Table 5.2, the complexity order for algorithm 2 is approximately $4K$ O(MN)+ $2K$ O(MNlog(MN)). To summarize, algorithm 2 is obviously computationally more efficient by a factor of $25T_{tot,k}$ (*i.e.*, equivalent to the UE-k buffer length history).

5.3.4 Numerical Simulations

In this subsection, the performance of the proposed schemes is investigated in comparison with the energy efficient scheduling scheme proposed in [78], named as the green resource allocation (GRA) scheme. It should be noted that the GRA scheme is primarily designed to optimize only the UE's EE subject to fairness among users. Thus, and to ensure fair comparison, we implemented different delay-aware versions of it supported by well known real-time traffic scheduling policies such as LWDF, M-LWDF [104] and EXP [105]. The propagation channel for each UE is independently modeled as a quasi-static block Rayleigh (QSBR) fading channel [14] with an average SNR of 10dB. More specifically, the channel gain and phase are characterized by time selectivity from one TTI to another and frequency selectivity from one sub-band to another. The 3MHz LTE channel is utilized, and hence,

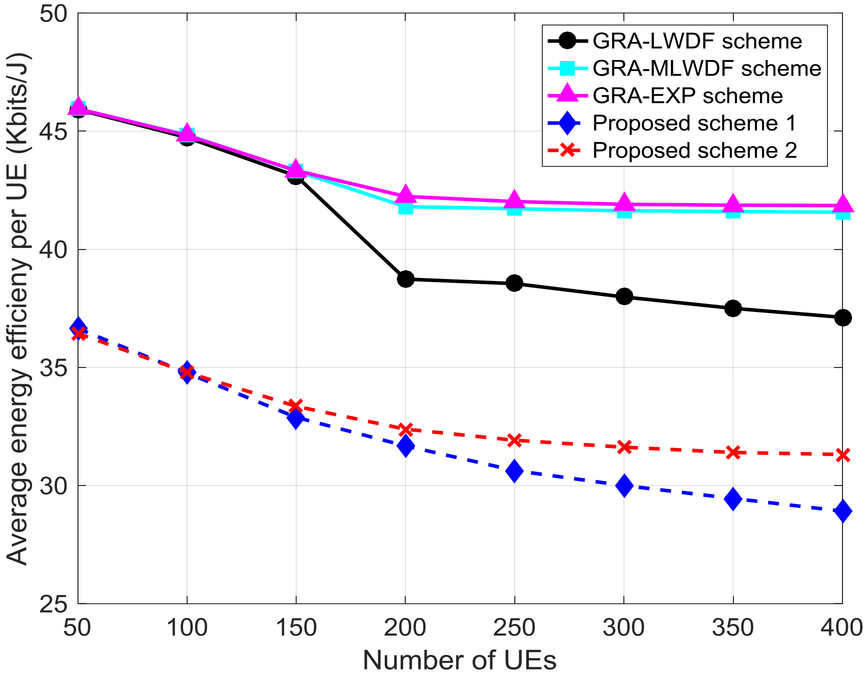


Figure 5.7: Energy efficiency

15 RBs are available in each TTI. For each UE buffer, VoLTE packets are generated using the adaptive multirate (AMR) codec based model used in [106]. The model employs the ON-OFF process which simulates the talk (ON) and silence (OFF) periods of a voice conversation. The duration of the ON and OFF periods are negative exponentially distributed with an average of 3sec. Setting the AMR codec to a rate of 12.2Kbps, VoLTE packets are generated during the the ON period with a size of 244 bits at a 20msec of interarrival time. We investigate the system's performance under increasing traffic loads by increasing the number of VoLTE UEs from 50 to 400 in a steps of 50 for a total simulation time of 500sec (*i.e.*, 50k LTE frames). It is worth mentioning that decreasing the step size for the number of UEs within the same selected range did not change the general trends of the obtained results.

The results in Fig. 5.7 show the EE performance for our proposed schedulers compared to the aforementioned existing schemes. For all schemes, the EE generally drops as the number of UEs increases due to the network's high traffic load which limits the EE optimization ability of the scheduler. The EE performance of our proposed schemes is found to be lower than all other traditional schemes. However, this was found to be the cost of

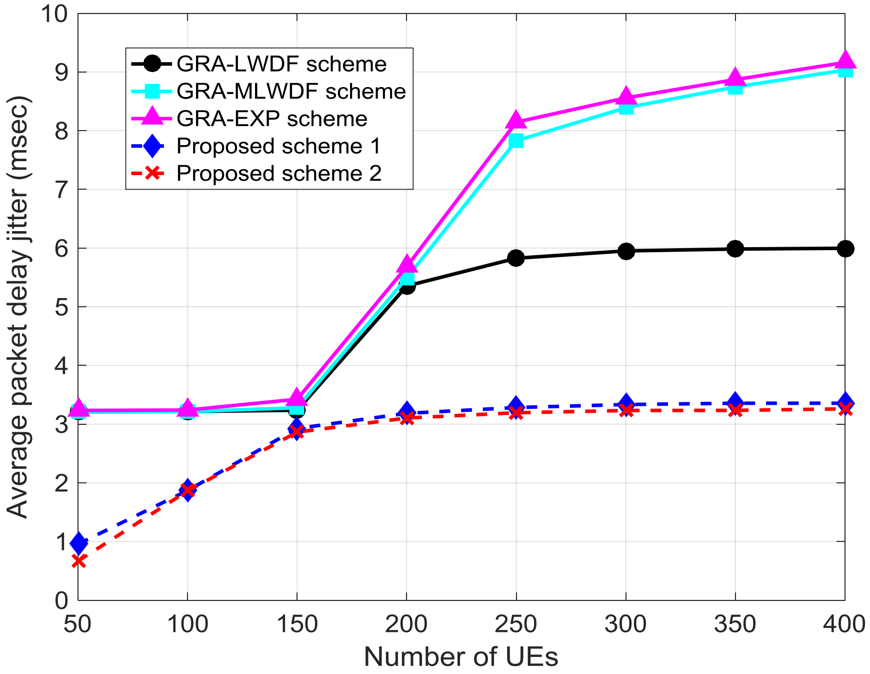


Figure 5.8: Average packet delay jitter

improving the delay jitter performance by a substantial high percentage (compared to that of the EE loss) as shown in Fig. 5.8. This EE/jitter trade-off is attributed to the fact that our proposed jitter-efficient resource allocation algorithms tend to statistically spread out the scheduling of resources in time to compensate for the jitter. That behavior is totally opposite to the EE scheduling [79] which targets to minimize the UE's active periods of reception to conserve the battery energy. The trade-off is quantitatively illustrated in Fig. 5.9 and confirms the importance of our proposed schedulers. For instance, scheme 2 was found to achieve at least 65% improvement in the jitter performance at the expense of 25% loss in the EE (compared to the GRA-EXP scheme) when the system is highly congested.

Fig. 5.9 also shows that the proposed scheme 2 is attaining a better trade-off compared to scheme 1. This pertains to the jitter optimization mechanism of scheme 1 that is dependent on the jitter history of past scheduled packets. This dependence essentially creates a jitter adjustment error, for each packet, that accumulates over time and directly affects the allocation time of resources (*i.e.*, EE). The effect of that error becomes even worse at highly loaded system. On the other side, scheme 2 adjusts the delay jitter for each packet only based on the departure time of the last scheduled packet in the queue. Thus,

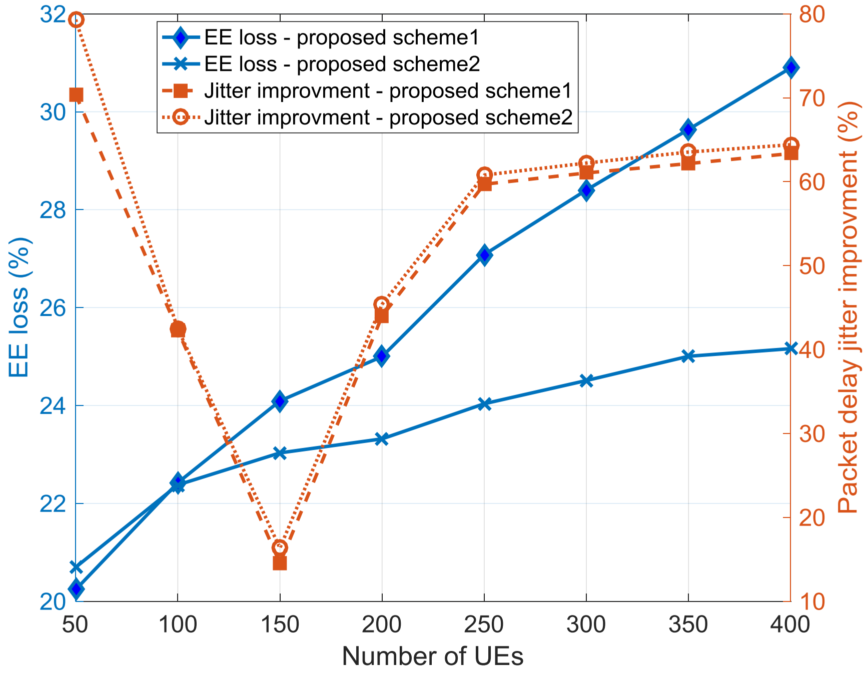


Figure 5.9: EE vs. packet delay jitter trade-off

the jitter adjustment error imposed by scheme 1 does not exist in scheme 2.

Not only the EE loss is found to be the cost of improving the delay jitter performance for our proposed schemes, but also the average packet delay as shown in Fig. 5.10. As expected, the EXP rule-based scheduler shows the best delay performance compared to the M-LWDF and LWDF policies, however, the worst jitter performance. The increased average packet delay for our jitter efficient schedulers is understood in light of their design nature. That is, improving the jitter dictates that the scheduler keeps the delay variation between successive packets as low as possible. With this in mind, any packet which happens to experience a large delay value, the scheduler strives to keep this large delay unchanged for all subsequent packets to achieve low delay jitter.

Finally, in terms of fairness, Fig. 5.11 depicts the Jain Fairness Index (JFI) [107] in terms of the average packet delay for all schemes. The results confirm that our proposed schedulers attain comparable fairness (even better at high network load) compared to others.

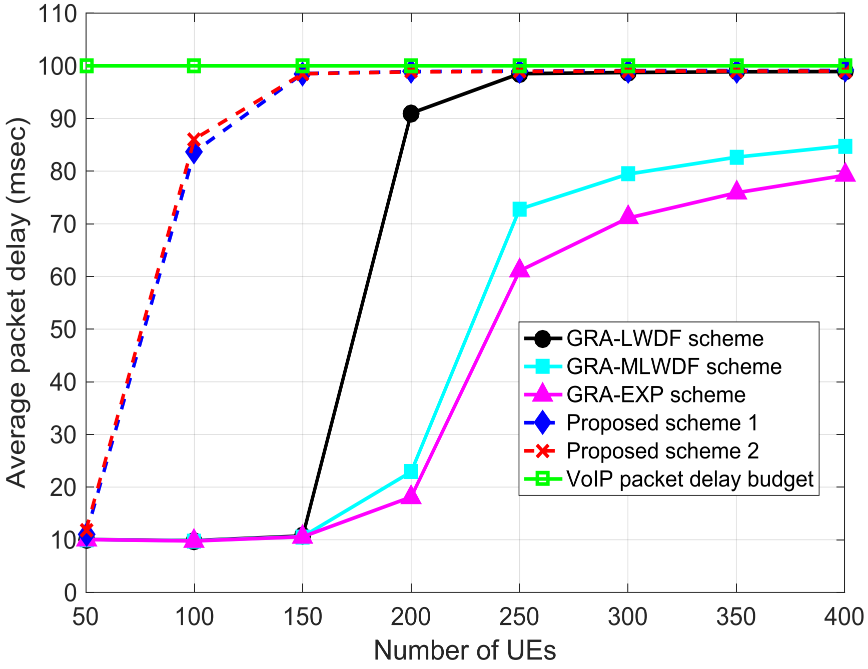


Figure 5.10: Average packet delay

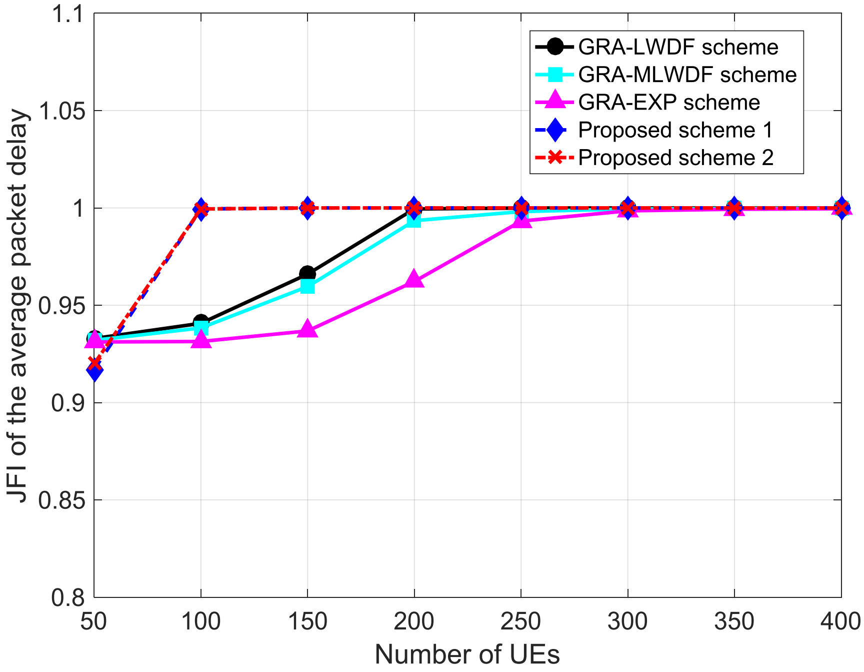


Figure 5.11: JFI

5.4 Chapter Summary

In the first part of this chapter an extensive analytical model for the delay jitter in the single-queue-single-server queuing system has been presented. The parameters affecting the delay jitter were found to change according to the system's utilization level and the service statistics of the queue. The presented expressions were tested via numerical simulations and were found to be accurate. The presented formulas could be augmented for deriving jitter formulas of large scale scenarios needed for designing jitter-aware packet schedulers in the current 4G networks. The insights gained in this part were then utilized, in the second part of the chapter, to design jitter-efficient heuristic packet scheduling algorithms for LTE networks.

In the second part, a novel perspective for the UE's EE in the LTE downlink was presented. Considering a real-time VoLTE traffic, the presented framework revealed an inherent trade-off between the EE and the delay jitter performance. The resource allocation problem was formulated as a binary integer programming (BIP). Due to the jitter optimization problem intractability, two computationally efficient heuristic algorithms were designed for the proposed scheduler. The proposed low complexity heuristic schedulers were able to strike a proper balance between the EE and the attained QoS, in terms of delay jitter, constrained by a fixed packet delay budget. Numerical results confirm that a substantial improvement in terms of the EE versus the delay jitter was fulfilled by our proposed schedulers compared to existing state-of-the-art schedulers.

Chapter 6

Studying the Energy/Delay Jitter Efficiencies Trade-off in Heterogeneous Traffic LTE Networks with QoS Awareness

6.1 Introduction

The staggering leap that occurred in the wireless technologies field has resulted in a tremendous expansion for the wireless market. Two big industries have remarkably and simultaneously emerged as a result of that expansion, that are, multimedia services and smart cell phones.

As one can remember, the multimedia services started with the first deployment of the 2G system namely the global system for mobile communications (GSM) system in the beginning of the 90's of the last century by just sending text, small images and short voice messages. Getting down the wireless communications standards road, today's LTE (*i.e.*, 4G) smart cell phone is simply capable of replacing any other electronic device such as laptop computers, GPS devices, and cameras. More specifically, with the aid of the super fast data transfer rates (*i.e.*, ultimately 300Mbit/s in downlink, and 75Mbit/s in uplink for the 20MHz channel [10]), the LTE smart cell phone has enabled a wide spectrum of multimedia services which range from VoIP, web-browsing, e-mail and social networking to data demanding services such as: Netflix and YouTube streaming, and interactive online gaming. Furthermore, it is envisioned that in the fast approaching new wireless technology,

---

A version of this chapter is submitted to [108].

known as the 5G [109, 110, 111], the wireless network will become a massive integrated environment of diversified devices and data services. That ambitious concept is known as the internet of things (IoT) [41, 112]. The IoT network will be seamlessly connecting the smart cell phone with the physical world (*e.g.*, intelligent traffic and transportation systems, smart grids, smart homes, health monitoring systems, media, environmental monitoring systems and emergency services) for creating better opportunities based on the user's location and mindset in terms of economic benefits, security, medical care and green environment. As a result of all of these advancements that occurred and those which are expected to come in the near future, the new wireless era (*i.e.*, delivered by the 5G and IoT technologies) is evidently going to have an even bigger effect on people's daily life.

6.1.1 Research Problem

Despite all of the promising capabilities that the current 4G system is delivering or even those coming in the future 5G system, one major problem which occupies the attention of both the mobile users and operators, and hence, the research and idustrial communities is the *Energy Efficiency* (EE). The problem is getting even more exigent with the rocketing progression of the wireless standards and data hungry multimedia services. From the user's viewpoint, the EE problem is perceived as the insistent need of having today's smart cell phones with longer battery lifetime than what the current battery technology is delivering while maintaining the increasing processing power needs [113]. Nevertheless, from the operator's perspective, all carriers seek to reduce the high operational expenditure (OPEX) coming from the electricity consumption bills and to meet the new governments' regulations: for conserving the environment's natural resources, and decreasing the carbon footprint.

In this work, we tackle the EE problem from the UE's side. Our previous work in [69] has invesitgated the EE improvement while considering the effective bandwidth requirements [101]. However, in this work, we provide a dual insight on both of the system's EE and delay jitter performance (for real-time services) while concurrently considering real behavior of different traffic types with different QoS requirements. In particular, for real-time and jitter-sensitive traffic, our framework targets to simultaneously improve the

UE's EE and the packet delay jitter performance while meeting strict packet delay or rate constraints. This is in contrast to the majority of the published works which target improving the EE while only meeting certain delay or rate constraints and disregarding the delay jitter (*i.e.*, fundamental QoS metric for most real-time applications). On the other hand, for non real-time traffics, the objective is to improve the UE's EE subject to throughput requirements and the fair distribution of the cell throughput among multiple users, with less strict packet delay constraints.

6.1.2 Related Work

The problem of improving the EE has become a fundametal aspect of modern wireless access networks, and hence, has been heavily studied in the literature [72]. One of the most popular metrics used to define the EE is the *bits-per-joule* metric. *Bits-per-joule* is defined as the number of information bits transmitted (or received) per each unit joule consumed. Thus, based on that definition, improving the EE is achieved by two regimes. The first is equivelant to increasing the number of transmitted (or received) bits (*i.e.*, throughput) per each joule consumed by the transceiver circuit, while the second is conversely equivelant to decreasing the amount of energy consumed for the same amount of information transmitted (or received) by the transceiver circuit. Despite the fact that the two regimes seem to be bilateral, they are found to be used distinctly under two major classifications of the resource allocations problems, namely rate and margin adaption (RA and MA) [114]. The RA problem corresponds to the first regime, whereas the MA problem corresponds to the second. Although both of the RA and MA regimes appear to increase the *bits-per-joule* metric, the MA is generally considered to be crucial for energy-efficient scheduling especially for battery-limited hand-held devices [115]. Based on our research problem highlighted in the previous section, we strictly consider throughout the following survey the EE from the UE side either in the uplink or downlink. Hence, we focus in the following discussions on the MA regime only from the UE side.

Despite having similar definition, the application of the MA regime for improving the UE's EE in the uplink differs from that in the downlink. In the uplink, it either adapts the UE's transmission rate and power based on both of the channel and buffer states [116], or uses a stricter mechanism to schedule the UE to transmit at time instances during which the channel's signal-to-noise ratio (SNR) is high subject to delay constaints. The strict scheduling mechanism is denoted as stochastic scheduling [117]. The optimal EE performance for the MA-based scheduler in the uplink usually comes at the expense of the average packet delay, and hence, the buffer overflow and the packet loss rate (PLR). In the downlink, the MA regime is mainly time-based and is commonly known as *power-management* approach [78]. More specifically, the UE's EE in decoding the received data packets dictates scheduling the UE to turn on its receiver circuits in the minimum possible transmission time intervals (TTIs) (*i.e.*, equivalent to the minimum possible energy consumption) to receive the same amount of data bits. The most popular *power-management* approach is the one currently implemented in LTE networks and is known as the discontinuous reception (DRX) mechanism [79]. It should be noted that the application of the MA regime for optimizing the UE's EE in the downlink is found to be less studied in literature compared to that in the uplink. As a result, we directed our efforts in our previous work [69] as well as in this work for studying the EE problem in the downlink from the UE side by designing suitable MA-based scheduling schemes.

Regardless of whether studying the problem in the uplink or the downlink, the major complexity of addressing the EE problem is generally due to other associated conflicting constraints from one side and the limited availability of spectral resources from the other side. That intricate and manifold picture could be explained as seeking an energy-efficient operation for the UE while fulfilling some stringent QoS levels (*i.e.*, rate or delay) and some degrees of fairness among different users (or different traffic classes) with a scarce spectral resources over time and frequency varying channels. In the following paragraphs, various examples of that picture (in both of the uplink and downlink) are provided to shed light on some of the recent works published in the literature in the context of energy-efficient wireless communications for battery-limited devices.

In the uplink, extensive studies have been conducted. Some of these works were reported in [1, 59, 60, 65, 118, 119] and summarized in Table 6.1.

In the downlink, although numerous works (*e.g.*, [73, 75, 77]) studied the EE problem yet few of them [69, 74, 78, 79, 120] have considered it from the UE side. This can be attributed to the common belief that the uplink power consumption dominates the UE's battery power consumption budget [65] because of the RF power requirements, especially at poor channel conditions and long distance transmission. However, according to the experimental results reported in [82], the UE's power consumed by the receiver's baseband and RF circuits for decoding the received data in the downlink occupies at least 40% of the total power consumption budget. As a result, the UE's EE in the downlink turned out to be of significant importance and worth studying unlike what was commonly believed. A summary of the works which have considered the UE's EE in the downlink is provided in Table 6.1.

Table 6.1: Some of the recent EE schedulers reported in literature

Ref.	**Link direction**	**Problem description**	**Solution methodology**
[1]	Uplink	Minimizing the total transmit power for all UEs in an LTE cell subject to rate, delay and maximum transmit power constraints within single TTI.	• Two low complexity heuristic schedulers based on the greedy algorithm were proposed to solve the resource allocation problem. • The optimal scheduler is formulated as a binary integer programming (BIP) problem to benchmark the proposed heuristic schemes.
[59]	Uplink	Developing an energy-efficient SC-FDMA packet scheduler which reduces the transmit power allocation per unit time constrained by a fixed transmission rate and the on-going retransmissions of the synchronous hybrid automatic repeat request (HARQ).	• To eliminate the ARQ blocking, the author proposed a method for limiting the number of new transmissions within the ARQ window by appropriately selecting the duration of the scheduling epoch. • The author proposed two power-efficient heuristic schedulers which showed comparable performance to the optimal method with reduced complexity.

[60]	Uplink	In contrast to [1], a global optimal scheduler for minimizing the power sum of all UE's subject to delay constraints was proposed.	• The scheduling process is formulated as a dynamic programming (DP) problem. • The proposed framework considers the dynamic nature of the traffic load and the maximum transmit power threshold. • Two low complexity heuristic schedulers were proposed to solve the DP problem.
[65]	Uplink	Developing a low complexity energy-efficient scheme compared to a previously proposed (by the same authors) iterative search approach while maintaining proportional fairness among users for an OFDMA system.	• Utilizing the time-average bits-per-joule metric, the author derived a closed form expression for the user's energy-efficient link adaptation followed by an expression for the energy-efficient resource allocation. • The author compared the proposed schemes with the global optimal solutions that were found by an exhaustive searches.
[118]	Uplink	Minimizing the average transmission energy subject to packet delay constraints for a single user time-slotted system.	• Packet scheduling policy that utilizes the time correlation properties in Poisson traffic. • The QoS is provisioned by studying the (maximum transmission rate/buffer overload probability) and (queue size/PLR) relationships.

[119]	Uplink	Minimizing the average transmission energy, for an individual point-to-point wireless link, constrained by an upper bound time deadline for transmitting certain amount of data and a specified signal's power level at the receiver.	• The proposed scheduling policy exploits the time-varying channel characteristics to find the energy-optimal time instant to communicate based on the optimal stopping theory. • The proposed scheduler uses a finite decision horizon by considering QoS constraints. • The proposed scheduler provisions the trade-off between the EE (*i.e.*, holding the transmission for better channel conditions) and the energy consumption cost of further channel observation constrained by the time deadline.
[69]	Downlink	Increasing the UE's EE and the system's admittance capacity subject to various effective bandwidth constraints.	• An optimal framework which minimizes the energy consumption of the UE receiver circuit while satisfying a constant rate (i.e., effective bandwidth) constraint was proposed. The framework has utilized a novel predictive scheduling scheme employing a cloud-based ray tracing channel prediction. • A low complexity heuristic algorithm to solve the optimal formulation was proposed.
[74]	Downlink	Jointly optimizing the base station and the UE EE in the OFDMA system subject to non guaranteed bit rate (NGBR) and delay constrains.	• A general optimization formulation to jointly minimize both of the UE circuit power consumption and the base station transmit power consumption was provided. • Due to the non-convexity and intractability of the joint formulation, a sub-optimal method was proposed to approximate the solution of the problem on the expense of the achieved rate.

[78]	Downlink	Optimizing the UE's EE in OFDMA systems subject to fairness.	• A nonlinear integer formulation for minimizing the number of wake-up TTIs during which the UE turns its receiver circuit ON to decode the downlink traffic was provided. • To solve the optimization problem, an iterative search algorithm was proposed to efficiently reach the optimal solution. • The proposed algorithm utilizes the fact that the global optimal solution, for the nonlinear problem, is located at one of the vertexes of a polytope which encloses all the feasible solutions.
[79]	Downlink	Conserving the LTE mobile terminal's battery power using the enhanced DRX mechanism while maintaining certain bounds for the packet delays.	• The power saving methods in both of the LTE network connected and idle states are discussed. • The optimum criteria for selecting the DRX mode and the DRX parameters, to maximize the power efficiency and reduce the UE wake-up time (*i.e.*, packet delays), was presented.
[120]	Downlink	Same as [74]. However, unlike [78], a frequency selective fading is assumed. Moreover, the problem was formulated as a BIP problem.	• A BIP formulation for the joint optimization of the receiving and transmitting energies of the UE and the base station, respectively, was provided. • To solve the BIP problem, a time-slot oriented column generation based algorithm was proposed. The algorithm is based on splitting the original problem into two sub-problems and solving each separately.

6.1.3 Scope and Contribution

In this work, we consider the LTE downlink with the frame structure type 1 (*i.e.*, frequency devision duplexing (FDD) oriented) [11]. After thoroughly searching the literature, none of the published works in the context of energy-efficient radio resource allocation for battery-limited devices in the OFDMA system has considered the packet delay jitter as a target QoS metric for real-time traffics. Even with those few works found to be dealing with the EE problem from the UE side in the downlink [69, 74, 78, 79, 120], none of them considered the delay jitter performance of real-time (and jitter-sensitive) traffic flows. Therefore, in this chapter we propose an energy and jitter efficient predictive scheduler for battery-limited devices in downlink LTE systems. The proposed scheduler strives to achieve dual optimal energy and jitter performances while satisfying other heterogeneous QoS requirements.

In our framework, the proposed scheduler deals with three different metric functions each of which characterizes a distinct QoS class. The metric functions used are for best-effort, rate-constrained and delay-constrained traffic types. That is the most common classification used for provisioning the QoS of wireless networks [13, 81, 121]. Each QoS class is represented by a single unique traffic type. The scheduler deals fairly with the heterogeneous QoS classes using the utility-based scheduling methodology [81]. In addition to the traditional metric functions of each of the three aforementioned QoS classes, a new metric function (for the delay-constrained class) is proposed to address the delay jitter QoS metric simulatenously with the delay requirment. The proposed optimization problem has a dynamic objective function which changes based on the target traffic type (*i.e.*, QoS class). In particular, when allocating resources to jitter-sensitive traffic, the objective function incorporates two objectives which are the EE and the delay jitter performance. For all other jitter-insensitive traffics, the EE is the only objective targeted by the scheduler. The constraints (*i.e.*, delay and rate) are also dependent on the traffic type. The resource allocation problem is solved twice with respect to the scheduling time granularity. The scheduler first solves the optimization problem by utlizing the channel and buffer states' information (CSI and BSI, respectively) within a single LTE frame horizon (*i.e.*, 10msec of duration) similar to most traditional schedulers, and shows the EE versus delay jitter trade-off. Then in another higher predictive level, the scheduler utilizes our previously proposed C-RAN based ray tracing predictive scheduling model [69] to solve the optimization problem in a longer time horizon (*i.e.*, multiple LTE frames). Since we do not consider any traffic prediction method, the short-term knowledge of the BSI essentially limits the performance of our proposed predictive scheduler despite the available future CSI (*i.e.*, provided by our previous predictive model [69]). To overcome that problem, we propose a sliding window mechanism to ensure efficient operation for our predictive scheduler targeting better EE and delay jitter performance compared to traditional short range schedulers.

The contributions of this work are summarized as follows:

- We propose an optimal packet scheduling framework for improving the UE's EE and the delay jitter performance for real-time traffic flows in presence of heterogeneous traffic requirements for the downlink of LTE networks. The resource allocation problem is formulated as a mutli-objective integer linear programming (*i.e.*, binary integer programming (BIP)) problem.
- To ensure the ability of our proposed scheduler in supporting different QoS requirements, the proposed framework utilizes the popular utility-based scheduling approach to simultaneously deal with different traffic types which belong to best-effort, rate-constrained and delay-constrained QoS classes.
- We propose a two stage paradigm for improving the packet delay jitter. In the first stage, a newly proposed metric function that keeps monitoring the jitter performance (besides the delay) is used for jitter-sensitive connections. In particular, the conventional delay metric function is altered by the delay jitter resulting in a composite delay/jitter prioritization scheme. In the second stage, two jitter-efficient resource allocation mechanisms are proposed for minimizing the packet delay jitter.
- We propose two different heuristic versions of our packet scheduler based on the scheduling granularity. The first version tackles the EE/delay jitter problem within the commonly employed time horizon of single LTE frame. The second version provisions further potential in improving the EE and delay jitter performances by utilizing our previously proposed cloud-based predictive scheduling model. For better referencing throughout the text, we denote the first and second versions by short range version (SRV) and predictive version (PRV), respectively.
- To enable the PRV version of our proposed scheduler, a window-based mechanism

is proposed to alleviate the short-term BSI/long-term CSI imbalance problem.

- To address the complexity of the optimal scheduler, a total of four heuristic algorithms are proposed based on the designed jitter control mechanisms for each version of our proposed scheduler.

The rest of this chapter is organized as follows: Section 6.2 presents the system model, design objectives, and the proposed utility-based energy and jitter efficient packet scheduler. A literature survey focusing on modelling and controlling delay jitter for real-time traffic is provided in Section 6.3. The formulation of the resource allocation problem describing our proposed scheduler is thoroughly studied in Section 6.4. Section 6.5 introduces various low complexity heuristic solutions for solving the formulation of Section 6.4. Numerical validation of our proposed schemes compared to other existing schemes is provided in Section 6.6. Finally, Section 6.7 concludes the chapter.

6.2 System Model

We focus on a single cell of downlink LTE system that employs the OFDMA access technique. As shown in Fig. 6.1, the radio access part of the mobile network centralizes a pool of traditional base stations (*i.e.*, base band units (BBUs)) in a virtual cloud of shared wireless resources. The evolving architecture is known as centralized radio access network (C-RAN) [80]. At a glance, the C-RAN architecture offers greener and more cooperative cloud computing solution compared to the traditional RAN architecture. It allows dynamic sharing of baseband processing resources, reduction of the cost of building and maintenance for fixed base station sites, increasing the system resource utilization, and energy savings.

In addition to its C-RAN based architecture, the studied model of Fig. 6.1 illustrates the structure of the proposed downlink scheduling system. It is close to our earlier model proposed in [69]. However, it is more precise in dealing with heterogeneous traffic QoS requirements. In [69], the model provisioned the QoS requirements of each UE connection by ultimately meeting its effective bandwidth [101]. In this work, the employed model (*i.e.*, depicted in Fig. 6.1) distinctly provisions the QoS requirements of each UE connection (*i.e.*, bearer) type based on its target metrics (*e.g.*, rate, delay, jitter and average throughput). That

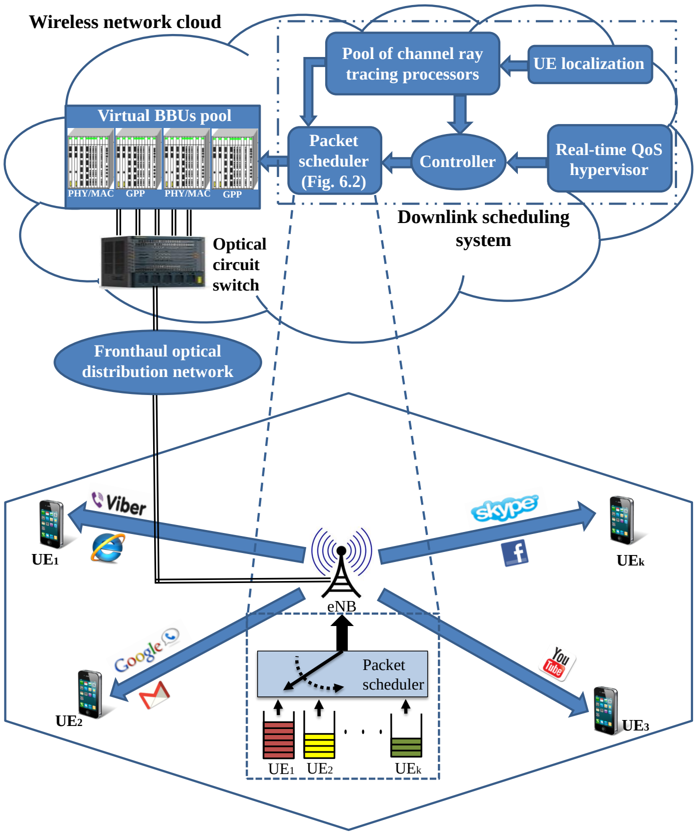


Figure 6.1: System model

is done using the real-time QoS hypervisor which is responsible for monitoring the QoS level of each UE connection in a strict and timely manner. Based on the attained QoS levels and the QoS class metrics, the packet scheduler is configured using a scheduler controller. The controller is used to set the appropriate resource allocation algorithm, metric function parameters and the solution space (*i.e.*, scheduling time granularity) of the packet scheduler. The detailed structure of the packet scheduler is provided and explained later in Fig. 6.2.

As will be shown later in Section 6.5, despite having the utility-based approach as the main prioritization scheme between different QoS connections, the proposed packet scheduler implements different resource allocation algorithms for real-time connections (*e.g.*, VoIP) than that used for non real-time connections (*e.g.*, video streaming and FTP). The difference is due to designing jitter-efficient resource allocation algorithms for real-time (*i.e.*, jitter-sensitive) applications which directly tackle the packet delay jitter requirement. This is unlike most of the published works which loosely assume acceptable jitter performance by just meeting certain level for the average packet delay. As mentioned before, the selection of the appropriate resource allocation algorithm is the controller's first function. The second function for the controller is to configure the priority weighting factors or even the metric function itself (as will be shown later in the case of VoIP). This is done for different QoS classes either in a dynamic manner or based on the operator's revenue policy. Last but not least, the scheduler controller can potentially enhance the system's performance, in terms of EE and jitter, by dynamically expanding the scheduler's time granularity. The expansion is based on the UE's channel future statistics and how far the UE's connection is from meeting the target QoS levels. The channel future statistics for each UE is predicted using a single processing thread from the pool of shared RT engines available in the cloud. As thoroughly explained in our earlier work in [69], the RT engine is capable of accurately pre-estimating the UE's propagation characteristics by just knowing the geometrical and morphological description of the propagation environment [18, 33] and the UE's geographical location. As proven in [69], increasing the scheduling time granularity supported by an accurate RT knowledge about the UE's future channel conditions has a significant impact on the system's EE performance. However, on the expense of the computational complexity. In this chapter, we proposes different framework that rigorously considers a heterogeneous traffic environment. The framework utilizes a predictive approach, as [69], in the second part of our analysis to further investigate its impact on the overall system's performance and trade-offs. Those results will be evaluated in comparison with those for the initially proposed short range schedulers that work within the traditional scheduling time horizon (*i.e.*, typically single frame) without the need for future RT channel predictions. More specifically, in the first part of our study we propose two different heuristic algorithms for the QoS-aware energy and jitter efficient scheduler designed primarily to work within a single LTE frame duration (*i.e.*, 10msec). For simple identification purposes, the proposed schedulers take the label SRV. In the second part, we redesign two more predictive versions of the SRV schedulers, labeled as PRV. The predictive versions schedule the users' traffic reception for multiple future frames by taking advantage of future RT-based CSI predictions in conjunction with another proposed window mechanism which addresses the UEs' short-term BSI. The reason behind designing the PRV schedulers is to investigate the upper bound for the overall system performance, and ultimately improve it, over our initially proposed SRV schedulers and other schedulers existing in the literature.

As illustrated in Fig. 6.1, we assume that the evolved Node B (eNB) (remote radio unit (RRU) as it is sometimes labeled in the C-RAN context) is located at the center of the cell and communicates with K UEs. As conventionally known in LTE, the total cell bandwidth is equally divided into N OFDMA resource blocks (RBs) each containing 12 sub-carriers with an overall bandwidth of 180kHz (*i.e.*, 15kHz bandwidth for each sub-carrier). Considering the LTE's frequency division duplexing (FDD) frame structure, each single frame consists of 10 subframes (TTIs) with a 1msec duration for each. According to the LTE standard [11], the eNB generally schedules the downlink transmissions for each UE over the physical downlink shared channel (PDSCH) during one TTI at a time. However, in our framework we assume that the normal setting of the eNB's packet scheduler is capable of scheduling downlink transmissions for the whole frame at a time. This is supported by the fact that the standard periodic channel state reporting mechanism [11] for the UE over the physical uplink control channel (PUCCH) - or the physical uplink shared channel (PUSCH) - can span up to 160msec intervals which is much greater than the frame duration. Moreover, utilizing our previously proposed RT-based model [69] for the eNB's scheduler in its simplest setting (*i.e.*, CSI prediction for single frame at a time), the eNB becomes aware of the UE's CSI for the whole frame, and thus, capable of scheduling the whole frame transmissions at once. The CSI knowledge is an essential information for the eNB to adapt the modulation and coding scheme (MCS), and hence, the transport block size per TTI, based on the UE's channel condition (*i.e.*, effective SNR γ_k) to maintain target block error rate (BLER) level. In a higher level, as explained in the previous paragraph, our proposed PRV schedulers empower the eNB to efficiently schedule the UEs' downlink data across multiple future frames as will be discussed later in Section 6.5.

6.2.1 Utility-Based Scheduling

The eNB is assumed to send H_k connections for each of the K UEs located within the cell coverage area. For the sake of studying the practical scheduling problem in a heterogeneous traffic network, each of the UE's connections is assumed to belong to one of three QoS classes. The classes which are widely adopted are: average throughput, rate-constrained and delay-constrained. Those classes are used to characterize the QoS requirements for best-effort (*i.e.*, NGBR), rate and delay sensitive (*i.e.*, GBR) traffic types, respectively. Unlike the previous works, in our framework we further add the delay jitter for the delay-sensitive QoS class as a QoS metric which is optimized simultaneously with the EE while provisioning the delay metric. As will be elaborated in Sections 6.4 and 6.5, the delay jitter optimization takes place in two stages. In the first stage, the conventional delay metric function [81] is adjusted by the jitter metric as follows:

$$X_k^h(m) = \alpha_k(h) - \frac{\overline{D_k^h(m)}}{D^{\max}} - \frac{\overline{\Delta t_k^h(m)}}{\Delta t^{\max}}, \quad \forall k \in K,\ h \in D^* \tag{6.1}$$

where $X_k^h(m)$ is the metric function for connection index h of UE k at TTI m, $\alpha_k(h)$ is a weighting factor designated for the QoS class characterizing the connection h of UE k such that $0 \leq \alpha_k(h) \leq 1$, $\overline{D_k^h(m)}$ is the average packet delay for connection h of UE k experienced up to TTI m, $D^{\max}$ is the packet delay budget for delay-constrained connections, $\overline{\Delta t_k^h(m)}$ is the attained average packet delay jitter for connection h of UE k until TTI m, $\Delta t^{\max}$ is a predefined threshold for the packet delay jitter, and D^* is the set of indexes for delay/jitter sensitive connections among the K UEs. The second stage of optimizing the delay jitter resides in the jitter-efficient resource allocation algorithms that will be discussed in Section 6.5.

For the other two classes of traffic (*i.e.*, best-effort and rate-constrained), we utilize the metric functions defined in [81] as follows:

$$X_k^h(m) = \frac{\overline{S_k^h(m)}}{\max_h \left\{\overline{S_k^h(m)}\right\}} - \alpha_k(h), \quad \forall k \in K,\ h \in E^* \tag{6.2}$$

$$X_k^h(m) = \frac{\overline{S_k^h(m)}}{S^{\max}} - \alpha_k(h), \quad \forall k \in K,\ h \in R^* \tag{6.3}$$

where $\overline{S_k^h(m)}$ is the average achieved throughput for connection h of UE k until TTI m, $\max_h \left\{\overline{S_k^h(m)}\right\}$ is the maximum achieved average throughput among the best-effort connections up to TTI m, E^* is the set of indexes for the best-effort connections among the K UEs, $S^{\max}$ is the maximum required data rate for the rate-constrained connections, and R^*

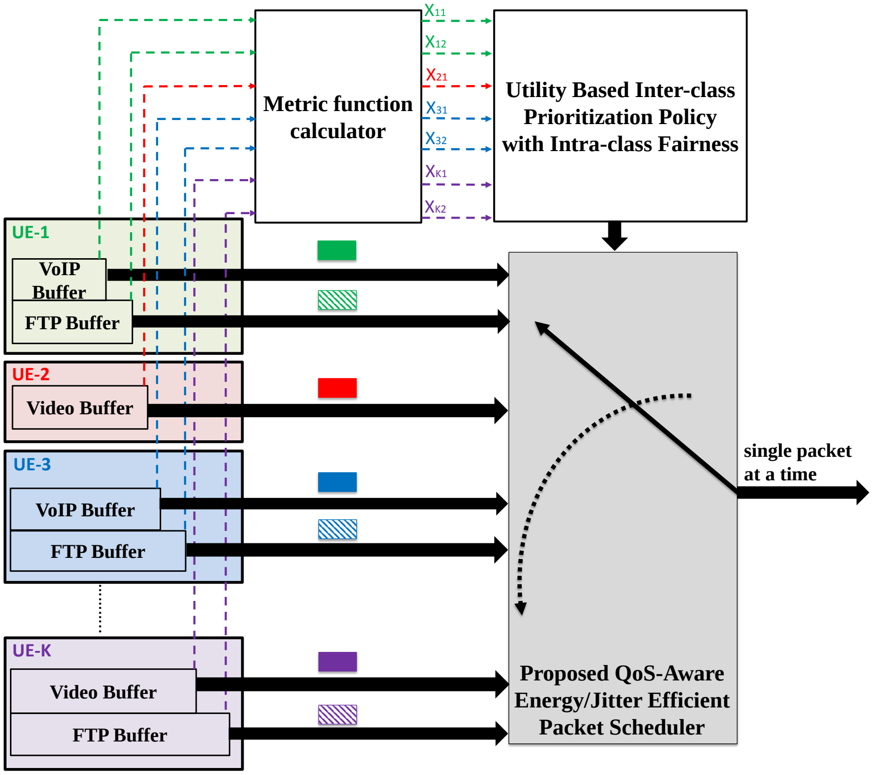


Figure 6.2: Proposed utility-based energy/jitter efficient packet scheduler

is the set of indexes for the rate-constrained connections among the K UEs.

The above metric functions provide a quantitative measure for the QoS perceived by the corresponding UE's traffic connection. In other words, the greater the metric function value, the higher the QoS level attained by the UE's connection. The metric functions are then used to build a utility-based inter-class prioritization platform with intra-class fairness for our proposed energy and jitter efficient packet scheduler. This could be illustrated with the aid of Fig. 6.2. It shows the detailed structure of our proposed packet scheduler (*i.e.*, initially presented in Fig. 6.1). The metric function calculator determines the metric function value for each UE connection as of equations (6.1), (6.2) and (6.3). All UEs' connections (*i.e.*, corresponding to all traffic classes) associated with their calculated metric functions are then placed in the same priority pool. Traffic connections are then prioritized out of the pool based on their calculated utility functions irrespective of which UEs they belong to. Thus, the utility-based prioritization policy is capable of dealing simultaneously with the heterogeneous QoS requirements, highlighted above, for all UEs based on a unified scale for all traffic types. In particular, the utility approach provides our proposed packet scheduler with a composite inter-user and intra-class prioritization policy. The utility function value for each UE connection reflects the degree of satisfaction for its correponding QoS metric. Hence, the higher the utility function value for a certain UE connection the lower the priority given (momenterly) to scheduling its packets. In addition, the intra-class fairness is coming from the fact that all connections which belong to the same QoS class have the same threshold(s) for the target QoS metric(s). In other words, all connections belonging to the same QoS class are weighted equally. However, the three defined QoS classes are weighted differently in the utility function, as shown below, based on the standard QoS class identifier (QCI) prioritization [11] produced by the European Telecommunications Standards Institute (ETSI). More specifically, by selecting the VoIP, video streaming and FTP traffics to represent the delay-constrained, rate-constrained and best-effort QoS classes, respectively, VoIP traffic takes the highest priority followed by the video streaming and the FTP traffic comes at the end.

The reason behind selecting the widely deployed utility-based prioritization policy in our framework pertains to its dynamic ability of continuously changing the priority between different traffic classes based on the network state (*i.e.*, UEs' CSI, BSI and fairness) to maximize the social welfare of the system (*i.e.*, summation of utility functions for all UEs' connections) [122]. This is in contrast to other strict priority techniques reported in literature (*e.g.*, differentiated service-based scheduling [123], and QCI-based scheduling [124]). The utility function used to prioritize UEs' connections is expressed as follows [81]:

$$U_k^h\left(X_k^h(m)\right) = 1 - e^{-\beta_h X_k^h(m)}, \;\; \forall k \in K, \, h \in H_k \tag{6.4}$$

where $U_k^h\left(X_k^h(m)\right)$ is the utility function for connection h of UE k, and β_h is the inter-class prioritization parameter. Generally speaking, β_h is left to be designed by the network operator based on either revenue or standard perspectives. However, for comparison purposes later in Section 6.6, in our work we stick to the β_h values assumed in [81] (which follow the standard ETSI QCI prioritization) for the VoIP, video streaming and FTP traffic types as follows:

$$\beta_h = \begin{cases} 4 & , h \in D^* \\ 3 & , h \in R^* \\ 2.5 & , h \in E^* \end{cases} \tag{6.5}$$

It can be seen in (6.4) that larger values for β_h imply higher sensitivity of the utility function (*i.e.*, steeper slope) to the variation of $X_k^h(m)$ across different UEs connections, and hence, higher priority.

6.2.2 Energy-Efficient Scheduling

The energy-efficient operation for the UE in the downlink, as originally defined in [79], dictates optimizing the operation time for the UE's receiver circuits subject to the packet delay constraint. In essence, the packet scheduler must find the minimum possible number of TTIs, within a certain interval of time, for the UE to receive the required amount of data that maintains stable buffer queue length and the average packet delay within a predefined threshold. During the rest of the scheduling interval's TTIs, the UE is switched to the sleep (or idle) mode during which the UE only listens to the dowlink channel (*i.e.*, primary sychronization signal (PSS) or secondary sychronization signal (SSS) [10]) ocassionally for synchronization purposes. According to the LTE UE's power consumption model developed by Jensen *et al.* in [82], the UE receiver circuits (*i.e.*, RF and baseband) consume constant power during the wake-up mode that is approximately four times the amount in the idle mode. Based on Jensen's model and our insights in [69], the critical energy (*i.e.*, only the constant component dominating the total budget) consumed by the LTE UE's receiver circuit in the downlink every single TTI can be expressed as follows:

$$E_k = T_s \left(\underbrace{m_{idle} \, P_{idle}}_{\text{idle state}} + \underbrace{\overline{m_{idle}} \, (P_{on} + P_{rx})}_{\text{wake}-\text{up state}} \right) \ \text{J} \tag{6.6}$$

where T_s is the TTI duration in seconds, m_{idle} is a logic variable that determins the UE's operation state, P_{idle} is the idle state power consumption (*i.e.*, equal to 0.5w [82]), P_{on} is the active state power consumption (*i.e.*, equal to 1.53w [82]), and P_{rx} is the base power consumed by the receiver chain during the active state (*i.e.*, equal to 0.42w [82]).

According to the previous discussion, a simplified formulation for the energy-efficient scheduling of K UEs (each having single connection) during M TTIs interval is expressed as follows:

$$\text{Objective :} \quad \underset{n \in N, H_{k \in K}=1}{\text{Min}} \sum_{k=1}^{K} \sum_{m=m_o}^{m_o+M} W_k \, E_{k,wake-up} \Gamma_k(m,n) \tag{6.7a}$$

Subject to:

$$D_k(m)\Gamma_k(m,n) \le D_k^{\max}, \ \forall k, \, m, \, n \tag{6.7b}$$

$$\sum_{k=1}^{K} \Gamma_k(m,n) \le 1, \ \forall m, \, n \tag{6.7c}$$

where W_k is an energy optimization weighting factor for UE k, $E_{k,wake-up}$ is the UE's receiver circuit wake-up energy consumption (*i.e.*, $T_s(P_{on} + P_{rx})$), $\Gamma_k(m,n)$ is a binary decision variable which determines the allocation decision of the RB with index n at TTI m to UE k, $D_k(m)$ is the packet delay attained by UE k if scheduled for transmission at TTI m (over any of the available N RBs), and $D_k^{\max}$ is the packet delay threshold for the traffic class of UE k connection.

The QoS constraint in (6.7b) ensures that the packet delay for each UE does not violate the predetermined threshold. On the other hand, the constraint in (6.7c) avoids allocating single RB to more than one user during a specific TTI. The weighting factor W_k in the objective function of (6.7a) could be dynamically adjusted. For instance, this can be based on either the UE's remaining battery capacity or a specific EE fairness criteria, especially in situations when the network is highly congested.

6.3 Delay Jitter Background

Packet scheduling schemes which support real-time applications over data communication networks have been a well studied subject by researchers since the 90s of the previous century. Some of these early attempts [125, 126] have put extensive efforts to control the end-to-end packet delay jitter levels in the network. In [125], the authors introduced a method of gauranteeing bounded delay jitter in a connection-oriented packet-switching store-and-forward wide area network (WAN). The method envolved keeping the delay experienced by any packet between a pre-determined minimum and maximum thresholds. In other words, the method assumed that gauranteeing certain bounds on the packet delay at each of the network's intermediate nodes will consequently assure acceptable end-to-end delay jitter performance. In [126], the authors proposed two jitter compensating scheduling schemes to control the end-to-end delay jitter levels of real-time flows in asynchronous transfer mode (ATM) networks. The proposed schemes utilized two methodologies for controlling the delay jitter. One methodology was defined by setting a serving priority, at each intermediate node along the packet's path through the ATM network, to different flows according to a target packet departure time. The other methodology partitioned the node's server capacity among various flows, with backlogged packets in their buffers, based on the head-of-line (HOL) packet delay. Another unique approach was noted to deal with the delay jitter problem, after bypassing the traffic source and the network routing charactersitics, by only compensating it at the receiver side. The approach is known as jitter buffer mechanism [127]. The mechanism implied buffering each of the transmitted packets at the receiver side and delaying it by a certain amount of time (*i.e.*, less than the packet expiration time), before playing it again out of the buffer to the decoding circuit, such that the jitter imposed by the network is minimized. Both the buffer size and its play-out delay time were dynamically adapted based on the traffic statistics to maintain acceptable trade-off between the packet's delay jitter and loss rate performances. Moreover, relatively recent works [94, 128] also addressed the delay jitter problem within heterogenous traffic environments. The authors in [128] proposed an alternative scheduling algorithm to the well known guaranteed-rate (GR) scheme [129] targeting the IPTV traffic over IEEE 802.11 based wireless mesh networks (WMNs). The objective of the proposed scheme, namely virtual reserved rate GR, was to improve the delay and jitter performances of the IPTV service while serving other lower priority traffic classes. In contrast to the GR scheme which allocates fixed reserved rate, the proposed scheduler implemented a prioritization scheme in the medium access control (MAC) layer which dynamically increases the reserved rate, at each network node (*i.e.*, router), for the IPTV traffic (*i.e.*, the target traffic) while sacrificing lower priority traffic types. The rate adaptation scheme has showed lower delay and jitter bounds compared to the GR-scheme. In [94], the authors rather modeled the packet delay jitter of real-time services considering the effect of having coexisting non real-time ones in packet-switched networks. The model utilized a double-queue single-server and limited-cache vacation queuing set-up to model the real-time and non real-time buffering at each network node. On the other hand, the interaction between both queues was modeled using the Markov theory.

It could be noted that the above highlighted researches have tackled the delay jitter problem mainly on the network layer. That is, the delay jitter originating from the routing latencies within multi-hop packet-switched data networks. However, without loss of generality, considering the case of direct downlink transmission within a single cell of today's advanced LTE cellular networks, the delay jitter problem is potentially emanating from the scheduling and queuing latencies in the MAC layer. Those latencies are the result of the network congestion level and the wireless channel capacity limitation in time and frequency. In this work, we focus on that case in dealing with the delay jitter problem as it was noted to be a less studied subject in the literature. Before formulating the complete scheduling problem, which integrates the utility-based and energy-efficient scheduling mechanisms (explained in Sections 6.2.1 and 6.2.2, respectively) and the delay jitter optimization, we first present in the following paragraph the definition and notation for the packet delay jitter.

Considering the downlink connection h (essentially VoIP traffic) of UE k, the packet delay jitter of a specific packet indexed a is the difference between its experienced queuing delay and the queuing delay of the preceding packet, in the queue, indexed $a-1$, *i.e.*,

$$\Delta t_k^h(a-1,a) = D_{k,h}^{a} - D_{k,h}^{a-1} \tag{6.8}$$

Assuming that the arrival times of packets a and $a-1$ are equal to $t_{k,h}^A(a)$ and $t_{k,h}^A(a-1)$, respectively, and similarly $t_{k,h}^D(a)$ and $t_{k,h}^D(a-1)$ for the departure times, (6.8) could be re-written as follows:

$$\Delta t_k(a-1,a) = \tau_k^D(a-1,a) \quad - \quad \tau_k^A(a-1,a) \tag{6.9}$$

where $\tau_k^D(a-1,a)$ and $\tau_k^A(a-1,a)$ are the interdeparture and interarrival times between two consecutive packets indexed $a-1$ and a, respectively. Thus, according to (6.9), minimizing the delay jitter for packet a is equivalent to keeping its interdeparture time as close as possible to its interarrival time with respect to the preceding packet $a-1$. However, utilizing (6.9) for a stream of packets leads to minimizing the average delay jitter across a whole stream of packets. That idea is later employed in one of the proposed heuristic scheduling algorithms presented in Section 6.5 for optimizing the VoIP jitter performance.

6.4 Problem Formulation

In this section, the optimal resource allocation problem for the heterogeneous traffic environment is formulated. The problem formulation provisions the QoS requirements for the three traffic classes highlighted in Section 6.2.1. The problem's global objective for all traffic types is to optimize the UE's EE, according to the methodology explained in Section 6.2.2, subject to their corresponding constraints. However, and unlike previous works, the delay jitter is defined as an additional objective for the VoIP traffic (*i.e.*, delay and jitter sensitive). In other words, the scheduling problem is a multi-objective optimization problem only for the VoIP traffic connections in terms of the EE and the delay jitter subject to the packet delay budget. On the other hand, single objective optimization in terms of the EE is the case for the video and FTP connections. The resource allocation for video connections is constrained by the minimum acceptable rate, whereas, best-effort allocations are considered for the FTP. Furthermore, the inter-user and intra-class fairness are attained using the utility-based approach as discussed in Section 6.2.1. All of the these diversified requirements are combined together in a single optimal formulation for the proposed packet scheduler as follows:

Min

$$\begin{aligned} Z_1 = &\sum_{k=1}^{K}\sum_{h=1}^{H_k}\sum_{m=m_o}^{m_o+M-1} w_k\, E_{k,wake-up}\, \Phi_k(m) \; + \\ &\left(\sum_{k=1}^{K}\sum_{h\in D^*} \underbrace{\frac{1}{1-e^{-\beta_h\left(\alpha_k(h)-\frac{\overline{D_k^h(m_o)}}{D^{\max}}-\frac{\overline{\Delta t_k^h(m_o)}}{\Delta t^{\max}}\right)}}}_{U_k^h\left(X_k^h(m_o)\right)\big|_{h\in D^*}} \times \right. \\ &\left. \sum_{a=1}^{A_k^h(m_o)}\sum_{m=m_o}^{m_o+M-1}\sum_{n=1}^{N} \Delta t_k^h(a-1,a|m)\Psi_{k,h}^a(m,n) \right) \end{aligned} \tag{6.10a}$$

Subject to

$$\sum_{a=1}^{A_k^h(m_o)}\sum_{m=m_o}^{m_o+M-1}\sum_{n=1}^{N} B_k^h(m,n)\Psi_{k,h}^a(m,n) \;\geq\; \omega_k^h(m_o)\,,\;\; \forall k,h \tag{6.10b}$$

$$D_{k,h}^a(m)\Psi_{k,h}^a(m,n) \leq D^{\max}\,,\;\; \forall k, h\in D^*,\; a,m,n \tag{6.10c}$$

$$\sum_{m=m_o}^{m_o+M-1}\sum_{n=1}^{N} B_k^h(m,n)\Psi_{k,h}^a(m,n) \;\geq\; S^{\min}T_sM,\;\;\; \forall k, h\in R^* \tag{6.10d}$$

$$\begin{aligned} &\left(D_{k,h}^a(m)+t_{k,h}^A(a)\right)\Psi_{k,h}^a(m,n) \leq \\ &\left(D_{k,h}^{a+1}(m)+t_{k,h}^A(a+1)\right)\Psi_{k,h}^{a+1}(m,n),\;\; \forall k,h,\; a,m \end{aligned} \tag{6.10e}$$

$$\sum_{k=1}^{K}\Psi_{k,h}^a(m,n) \;\leq\; 1\,,\;\;\; \forall h,a,m,n \tag{6.10f}$$

$$\Psi_{k,h}^a(m,n) - \Phi_k(m) \leq 0\,,\;\;\; \forall k,h,a,m,n \tag{6.10g}$$

where H_k is the number of traffic connections for UE k, m_o is the first TTI of the currently observed scheduling time interval, M is the scheduler's time granularity (conventionally single frame) measured in TTIs, $\Delta t_k^h(a-1,a|m)$ is the delay jitter for the packet indexed a with its predecessor indexed $a-1$ for connection h of UE k if scheduled at TTI m, $A_k^h(m_o)$ is the total number of packets waiting in the queue for buffer h of UE k up to TTI

m_o, $\Psi_{k,h}^{a}(m,n)$ is a binary decision variable which indicates the scheduler's allocation decision for RB n during TTI m for packet indexed a of connection h of UE k, $\Phi_k(m)$ is another binary decision variable which adds $E_{k,wake-up}$ to the energy consumption budget of UE k if any of the N available RBs during TTI m is allocated to any of the packets belonging to its connections, $B_k^h(m,n)$ is the number of physical data bits that could be delivered by the RB n during TTI m for connection h of UE k based on its effective SNR and the corresponding configured MCS, $\omega_k^h(m_o)$ is the total length (in bits) for connection h traffic buffer of UE k at the beginning of the current scheduling interval (*i.e.*, m_o), $D_{k,h}^{a}(m)$ is the UE k experienced delay for packet a of connection h if scheduled over any of the RBs available during TTI m, $S^{\min}$ is the minimum required bit rate for rate-sensitive connections (*i.e.*, video traffic connections), and $t_{k,h}^{A}(a)$ is the arrival time for the packet indexed a to the buffer of connection h belonging to UE k.

The above formulation described in (6.10) is an NP-hard multi-objective BIP optimization problem. In particular, for the objective function of (6.10a), the term before the summation operator is the EE objective, whereas, the term after the summation constitutes the delay jitter objective for the delay (and jitter) sensitive VoIP connections. The delay jitter optimization is weighted by the inverse of the proposed delay and jitter sensitive utility function as described in (6.1) and (6.4). Therefore, the utility function is calculated for all VoIP connections (across all UEs) up to the beginning of the scheduling observation time interval (*i.e.*, $m = \{m_o, m_o + M - 1\}$) to set the jitter optimization priority for each UE connection. The constraint in (6.10b) sets a strict requirement for scheduling all packets waiting in the various queues of all UEs. The second constraint defined in (6.10c) assures meeting the packet delay deadline for each packet of the VoIP connections. Similarly for the rate-constrained connections, constraint (6.10d) guarantees meeting the minimum required data rate to support video connections. To maintain first-come-first-serve (FCFS) discipline of the proposed packet scheduler for each single UE queue, the constraint presented in (6.10e) ensures earlier departure times for earlier packet arrivals compared to those of the later arrived packets. The fifth constraint in (6.10f) is to avoid the intra-cell interference which occurs if single RB got allocated to more than one UE within the same TTI. Finally, the two binary decision variables, $\Psi_{k,h}^{a}(m,n)$ and $\Phi_k(m)$, are used together as shown in (6.10g) (*i.e.*, if-then constraint) to ensure that the objective function (6.10a) is penalized each TTI by $E_{k,wake-up}$ for each UE which has been assigned at least one RB for any of its connections.

It is obvious that the formulation described in (6.10) has a sort of ideality which questions its feasibility. This is due to the strict requirement addressed by the constraint (6.10b) in scheduling all packets waiting in all queues for all UEs. In practice, the system's limited spectral resources as well as the time and frequency varying CSI (which limits the system's capacity) restrain the scheduler from ideally guaranteeing the allocation of resources enough to serve 100% of the total traffic load for all users. Even though network operators implement call admission control policies to maintain stable operation for the network and secure certain levels for the QoS, the total traffic load is still being served in a queue according to the adopted scheduling policies due to the aforementioned practical limitations. Therefore, the constraint (6.10b) makes the formulation presented in (6.10) practically unfeasible.

To fix the practicality issue associated with the formulation in (6.10), we adopt the well known penalty method [87] to relax the constraint in (6.10b). The relaxation implies partial satisfaction of the constraint in (6.10b) for each UE connection based on a certain weighting mechanism. More precisely, the larger the weight designated to a connection the higher the partial satisfaction is achieved relative to other smaller weighted connections. The weighting mechanism utilized follows the utility approach discussed in Section 6.2.1. As a result, the formulation presented in (6.10) can be rewritten as follows:

Min

$$Z_2 = Z_1 + \left(\sum_{k=1}^{K} \sum_{h=1}^{H_k} \frac{1}{U_k^h\left(X_k^h(m_o)\right)} \times \sum_{a=1}^{A_k^h(m_o)} \sum_{m=m_o}^{m_o+M-1} \sum_{n=1}^{N} \omega_k^h(m_o) - B_k^h(m,n)\Psi_{k,h}^a(m,n) \right) \tag{6.11a}$$

Subject to

$$\sum_{a=1}^{A_k^h(m_o)} \sum_{m=m_o}^{m_o+M-1} \sum_{n=1}^{N} B_k^h(m,n)\Psi_{k,h}^a(m,n) \leq \omega_k^h(m_o), \ \forall k,h \tag{6.11b}$$

$$(6.10c), (6.10d), (6.10e), (6.10f) \text{ and } (6.10g) \tag{6.11c}$$

It could be seen that the new objective function in (6.11a) is equal to the former objective function in (6.10a) added to the penalty term. As explained in the previous paragraph, the added penalty term to the objective function provides a weighted partial satisfaction for the constraint in (6.10b) after relaxing it as shown in (6.11b). The rest of the constraints in formulation (6.10) remain unchanged.

Despite the fact that the formulation in (6.11) solves the feasibility limitation of the formulation in (6.10), formulation (6.11) is still intractable to find its solution using direct techniques. In particular, optimizing the second objective (*i.e.*, delay jitter) in Z_1 (*i.e.*, depicted in (6.10a)) incurs dramatically large degrees of freedom and recursive dependency on the scale of single VoIP queue size as well as the total number of active VoIP connections (*i.e.*, size of the indexes set D^*) across the K UEs. This problem is commonly referred to as a discrete time stochastic control process [1]. Some techniques (*e.g.*, Markov decision process [103]) were known to solve such kind of problems. However, the solution was found to be heavily dependent on the problem dimensionality. In our case, the problem dimensionality is a function of K, H_k, M, N and $A_k^h(m_o)$. As a result, the solution of the optimization problem in (6.11) is practically unreachable. Therefore, we propose different heuristic algorithms in the next section to efficiently find a sub-optimal solution for the problem in (6.11).

6.5 Heuristic Solutions

In this section, we propose four computationally-efficient heuristic schemes to facilitate finding a sub-optimal solution for the intricate optimization problem proposed in (6.11). The first two algorithms which correspond to the first version, labeled as SRV, of our proposed scheduler are designed to solve (6.11) for short range scheduling within a single frame time horizon (*i.e.*, $M = 10$ TTIs). The other two algorithms belong to the predictive version of our proposed scheduler, labeled as PRV, and solve (6.11) within longer time horizon that is multiples of single frame (*i.e.*, $M = 10y$ TTIs, where $y = \{2, 3, 4, ...\}$).

As will be elaborated in the following discussions, all of the proposed algorithms are based on the well known recursive greedy strategy. More specifically, the basic idea for all of the designed algorithms is that only single packet for single UE connection is scheduled at a time in an iterative fashion. A UE packet is selected among various packets waiting in other UEs' queues based on the utility-based prioritization policy explained in Section 6.2.1. The algorithms keep iterating over the head packets of all UEs' queues until at least one of two stopping conditions becomes valid. That is, either all of the NM available RBs within the M scheduling epoch become allocated or all UEs buffers are empty. Just as all heuristics, that iterative mechanism provides an acceptable alternative to the optimal solution of (6.11) which, otherwise, uniquely and concurrently finds the optimal allocations for all packets that could possibly be scheduled.

6.5.1 SRV-Based Schedulers

In this part, we introduce two heuristic algorithms representing the short range version (SRV) of our proposed scheduler to solve the optimization problem in (6.11) within a single frame time horizon (*i.e.*, $M = 10$). This version is intended to compare the performance of our proposed scheduler with other existing schedulers which typically work within the same horizon. The scheduler allocates the available resources of the whole LTE frame at once at the beginning of the frame. For every scheduled frame, it considers packets that only arrived to the UE buffer in preceding frames. In other words, each packet arrives to a UE connection buffer within a specific frame gets stored in the scheduler's list of buffered packets till the end of the same arriving frame. At the end of its arriving frame, each packet becomes relocated from the buffered list to the scheduling list to depart its buffer queue in any of the following frames. The details of the proposed allocation algorithms are provided in Tables 6.2 and 6.3.

Considering the first algorithm illustrated in Table 6.2, it works as follows. The algorithm starts, in line 2, by initializing an empty RB allocation matrix $\mathcal{R}$ of size $N \times M$. As explained above, the algorithm goes through iterations. In each iteration (lines 3-36) the algorithm sequentially allocates the required resources for scheduling the head-of-line (HOL) packet for each UE connection, one at a time. The prioritization for HOL packets is done in lines 4 and 5 by calculating the utility function for each UE buffer (*i.e.*, updated each iteration) and sorting them in ascending order, respectively. The FOR loop in line 6 resembles the HOL packet scheduling loop for all UEs connection. In line 8, the algorithm

Table 6.2: SRV-CO algorithm

```
Require: $K$, $H_k$, $M$, $N$
Initialize emtpy RB allocations matrix $\mathcal{R}, iter = 1$
while $(isempty(\mathcal{R})$ AND $A_k^h \neq 0,\ \forall\, k, h)$ do
  Calculate $U_k^h\left(X_k^h(iter)\right) \forall\, k\,,\, h$
  $[\text{Id}_k, \text{Id}_h] =$ Sort$(U_k^h\left(X_k^h(iter)\right), 'ascend')$
  for $i = 1$ to $\sum_{k=1}^{K} H_k$ do
    Set $k = \text{Id}_k(i)$, $h = \text{Id}_h(i)$
    Find $N^*$, $M^*$ to satisfy FCFS
    Update $N^*$, $M^*$ based on $D_{k,h}^{\max}$
    Sort$_{\textbf{EE}}(N^*)$
    if $h_k \in D^*$ then
      Calculate $\left|\tau_{k,h}^{A}(a-1,a)\right|,\ \left|\tau_{k,h}^{D}(a-1,a))\right|,\quad \forall\, a \in A_{k,h}^*$
      if length$(A_{k,h}^*) == 0$ then
        Find $\Psi_{k,h}^{a}(m,n)$, $a = HOL_{k,h}$, $n \in N^*$, $m \in M^*$
        Update $\mathcal{R}$, $A_{k,h}^*$
      else if length$(A_{k,h}^*) == 1$ then
        Calculate $\left|\Delta t_k^h(A_{k,h}^*, a\,|m)\right|$, $\forall n \in N^*$, $m \in M^*$, $a = HOL_{k,h}$
        Sort $(N^*, \left|\Delta t_k^h(A_{k,h}^*, a\,|m)\right|, 'ascend')$
        Update $M^*$
        Find $\Psi_{k,h}^{a}(m,n)$, $a = HOL_{k,h}$, $n \in N^*$, $m \in M^*$
        Update $\mathcal{R}$, $A_{k,h}^*$
      else
        Calculate $\left|\tau_{k,h}^{A}(\left.A_{k,h}^*\right|_{last}, a)\right|$, $a = HOL_{k,h}$
        Calculate $\left|\tau_{k,h}^{D}(\left.A_{k,h}^*\right|_{last}, a\,|m)\right|$, $a = HOL_{k,h}$, $m \in M^*$
        Sort$_{\textbf{JITTER1}}(N^*)$
        Update $M^*$
        Find $\Psi_{k,h}^{a}(m,n)$, $a = HOL_{k,h}$, $n \in N^*$, $m \in M^*$
        Update $\mathcal{R}$, $A_{k,h}^*$
      end if
    else
      Find $\Psi_{k,h}^{a}(m,n)$, $a = HOL_{k,h}$, $n \in N^*$, $m \in M^*$
      Update $\mathcal{R}$, $A_{k,h}^*$
    end if
  end for
  Set $iter = iter + 1$
end while
```

then finds the feasible RB allocations indexes in frequency and time (*i.e.*, N^* and M^*) from the matrix $\mathcal{R}$ which meet the FCFS constraint in (6.10e). The RB indexes are then updated, in line 9, after dropping those which violate the current packet delay threshold (*i.e.*, $D_{k,h}^{max}$). The RB indexes of line 9 are then sorted in line 10 according to the energy-efficient scheduling strategy proposed in [69] (*i.e.*, RBs with higher capacity have higher allocation priority than lower capacity ones). In line 11, the algorithm identifies the packet type to set the corresponding resource allocation approach. If the currently observed packet belongs to VoIP connection, then the scheduler will re-arrange the RBs of line 10 to optimize the packet delay jitter in addition to the EE, otherwise (*i.e.*, video or FTP packets) the scheduler finds the RB allocations (in lines 31-32) based on their arrangement in line 10. In case of VoIP type packet, the algorithm's key strategy is optimizing the average packet delay jitter for the whole VoIP connection (in addition to the EE), which makes the algorithm perceived as connection oriented (CO) in terms of jitter optimization, hence, named SRV-CO. The optimization of the connection's average jitter is achieved by selecting the RB allocations which results in a minimum absolute difference between the means of the previously scheduled packets' interarrival and interdeparture times, including the current packet, as highlighted in (6.9). To reach this objective, the interarrival and interdeparture times for the past scheduled packets of the current observed UE connection (*i.e.*, $A_{k,h}^*$) are first calculated (line 12). Three cases exist for the past scheduled packet interarrival and interdeparture times. The first case (lines 13-15) occurs at the beginning of the connection where no history of scheduled packets is available. Consequently, the algorithm only considers the EE when allocating resources using the RBs arrangement of line 10. After allocating the resources that fit the current packet size, the allocation matrix $\mathcal{R}$ and the scheduled packets history list $A_{k,h}^*$ are updated accordingly. The second case (lines 16-21) is when only one past packet exists in $A_{k,h}^*$. The jitter of the current packet, in this case, is optimized only with respect to the delay of the single packet stored in $A_{k,h}^*$ by calculating the jitter (as in line 17) that would be experienced at each RB allocation addressed by the sets N^* and M^*. The RBs in N^* are then re-sorted (in line 18) in ascending order based on the calculated jitter values. The algorithm then searches for the required resources in the sorted RBs list, as in lines 14-15, and updates $\mathcal{R}$ and $A_{k,h}^*$. The third and final case for scheduling a VoIP packet is addressed in lines 22-28. This case is clearly the dominating case in which the UE connection has more than one packet in the scheduling history list $A^*_{k,h}$. The jitter optimization in this case is uniquely attained by selecting the RBs which result in the least absolute difference between the average interarrival and interdeparture times of the packets in $A^*_{k,h}$ after adding the new corresponding packet departure time. For this, in line 23, the algorithm updates the interarrival calculations of line 12 by the current packet interarrival time. Following in line 24, the possible interdeparture times for the current packet over all of the available RB allocations addressed by N^* and M^* are calculated. For each value of the calculated interdeparture times, the absolute difference between the means of the interdeparture and interarrival times (after accounting for the current packet) is evaluated. Accordingly, the RBs in N^* are re-sorted in ascending order of the interarrival/interdeparture deviation. Both of the sorting function and inter interarrival/interdeparture deviation calculations are conducted by the function **Sort$_{\textbf{JITTER1}}$** in line 25. After updating the TTI indexes order in M^*, the algorithm finds the sufficient allocations to schedule the VoIP packet and updates $\mathcal{R}$ and the corresponding UE connection scheduled packets history list $A^*_{k,h}$. The algorithm continues in the same fashion, for the head packets of all UEs connections, inside the WHILE loop of line 3 until either no empty allocations are existing in the matrix $\mathcal{R}$ or no buffered packets are spotted in any of the UEs' queues. It should be noted that the logical function $isempty$ in line 3 searches for any empty RB allocations inside the matrix $\mathcal{R}$. Thus, it gives logic "1" if at least one empty allocation is found available in matrix $\mathcal{R}$.

As explained in the previous paragraph, the SRV-CO algorithm provisions the VoIP connection's whole history of scheduled packets in minimizing its overall average delay jitter. In contrast, the second algorithm, described in Table 6.3, optimizes the delay jitter of each single packet (*i.e.*, packet oriented (PO)) only with respect to the delay of its preceding packet, hence, named as SRV-PO. As a consequence, the SRV-PO algorithm in Table 6.3 follows the general approach of the SRV-CO algorithm except for the jitter optimization part. More specifically, for the VoIP packet case (lines 12-21), only two cases exist for allocating resources to the observed packet. The first case (lines 12-14) is similar to that of the SRV-CO algorithm (lines 13-15 in Table 6.2) which corresponds to the beginning of the connection where no history for scheduled packets is recorded. The second case (lines 15-20) encompasses the system's dominating state where at least one packet has

Table 6.3: SRV-PO algorithm

```
Require: $K$, $H_k$, $M$, $N$
Initialize emtpy RB allocations matrix $\mathcal{R}, iter = 1$
while $(isempty(\mathcal{R})$ AND $A_k^h \neq 0, \forall k, h)$ do
    Calculate $U_k^h \left( X_k^h(iter) \right) \forall k , h$
    $[\mathrm{Id}_k, \mathrm{Id}_h] =$ Sort$(U_k^h \left( X_k^h(iter) \right),' ascend')$
    for $i = 1$ to $\sum_{k=1}^{K} H_k$ do
        Set $k = \mathrm{Id}_k(i)$, $h = \mathrm{Id}_h(i)$
        Find $N^*$, $M^*$to satisfy FCFS
        Update $N^*$, $M^*$based on $D_{k,h}^{\max}$
        Sort$_{\mathbf{EE}}(N^*)$
        if $h_k \in D^*$ then
            if length$(A_{k,h}^*) == 0$ then
                Find $\Psi_{k,h}^a(m,n)$, $a = HOL_{k,h}$, $n \in N^*$, $m \in M^*$
                Update $\mathcal{R}$, $A_{k,h}^*$
            else
                Calculate $\left| \Delta t_k^h( A_{k,h}^* \Big|_{last}, a \left| m \right) \right|$, $\forall n \in N^*$, $m \in M^*$, $a = HOL_{k,h}$
                Sort$_{\mathbf{JITTER2}}(N^*)$
                Update $M^*$
                Find $\Psi_{k,h}^a(m,n)$, $a = HOL_{k,h}$, $n \in N^*$, $m \in M^*$
                Update $\mathcal{R}$, $A_{k,h}^*$
            end if
        else
            Find $\Psi_{k,h}^a(m,n)$, $a = HOL_{k,h}$, $n \in N^*$, $m \in M^*$
            Update $\mathcal{R}$, $A_{k,h}^*$
        end if
    end for
    Set $iter = iter + 1$
end while
```

been scheduled. In this case, as in lines 17-21 in Table 6.2, the available RBs addressed by the sets N^* and M^* are re-sorted by the function **Sort$_{\mathbf{JITTER2}}$** which only takes the delay of the last scheduled packet (*i.e.*, $A^*_{k,h}\Big|_{last}$) into account.

Based on the noted difference between the computational requirements of the proposed SRV-based schedulers explained above, it could be deduced that the SRV-PO algorithm is less complex than that of the SRV-CO. This pertains to the packets' history-based heavy computational requirement for the SRV-CO. This could be further confirmed by evaluating the complexity for both algorithms using the standard O notation. For the SRV-CO algorithm in Table 6.2, the complexity of the utility calculations and sorting in lines 4 and 5 is equal to O$\left(\sum\limits_{k=1}^{K} H_k\right)$ and O$\left(\sum\limits_{k=1}^{K} H_k.log\left(\sum\limits_{k=1}^{K} H_k\right)\right)$, respectively. The searching operation in lines 8 and 9 costs O(NM). As of line 5, the complexity for the sorting operation in line 10 is O($NMlog(NM)$) in the worst case scenario. For the IF-statement spanning lines 11-33, the worst case for the complexity corresponds to the VoIP type packet when $A^*_{k,h}$ has a record of scheduled packets (line 12 and lines 23-28). The worst complexity for the interarrival and interdeparture time calculation in line 12 is at the end of the VoIP connection time interval and is equal to O($25T^{tot}_{k,h}$), where $T^{tot}_{k,h}$ is the total duration for the VoIP connection h of UE k. The value $25T^{tot}_{k,h}$ is equivalent to the average number of VoIP packets generated during the connection's total time interval. The calculation is based on the ON-OFF traffic model [106] with a single packet generated every 20msec within the ON spurts each with a mean duration of 3sec. Finally, complexities of O(NM), NMO($25T^{tot}_{k,h}$)+O($NMlog(NM)$) and O(NM) are spent in lines 24, 25 and 27, respectively. Hence, the asymptotic upper limit for the complexity of a single iteration of the SRV-CO algorithm is approximately in the order of O($25T^{tot}_{k,h}$). Similar inspection for the complexity of the SRV-PO algorithm, in Table 6.3, leads to O(NM)+O($NMlog(NM)$). The previous complexity results strictly confirm a substantial complexity reduction for the SRV-PO algorithm compared to the SRV-CO algorithm as $T^{tot}_{k,h} >> NM\text{log}(NM)$.

It is compelling to highlight at this point that both the SRV-CO and SRV-PO algorithms achieve the delay jitter optimization in two stages. The first stage encompasses the delay/jitter prioritization for VoIP connections (in line 4 of Tables 6.2 and 6.3), with respect to video and FTP connections, with the aid of the proposed metric function in (6.1). The second stage is the jitter-efficient resource allocation approaches explained previously for both algorithms.

6.5.2 PRV-Based Schedulers

In contrast to the SRV-based algorithms presented above, the proposed PRV-based algorithms in this section investigate a different approach to reconcile the EE/delay jitter trade-off (as will be quantitatively illustrated in Section 6.6) resulting from the short-term scheduling and knowledge of the channel. Conversely, the PRV-based algorithms utilize the long-term knowledge about the UEs' CSI provided by the cloud-based ray tracing channel prediction system shown in Fig. 6.1, and as initially proposed in our recent study in [69]. This draws a new picture for the system of having a short-term demand (*i.e.*, packets arriving to UEs' buffers every frame), and long-term information about the system's frequency resources (*i.e.*, provided by the pool of ray tracing processors available in the cloud). In other words, the CSI is known for multiple future frames, whereas the BSI is only available every single frame (*i.e.*, no traffic prediction is employed) upon the generation of new packet(s). Generally speaking, the PRV-based algorithms have similar structure to their SRV counterparts in terms of optimizing the EE and the delay jitter. This is, however, tailored to the new aforementioned picture. In particular, after having all the new arriving packets of each UE connection settled inside their corresponding buffer until the end of the arriving frame (*i.e.*, frame during which the packet is generated), the PRV-based scheduler utilizes the RT future CSI knowledge in scheduling the buffered packets across multiple future frames using a sliding window mechanism as depicted in Fig. 6.3. The picture of Fig. 6.3 shows an example of the proposed scheduling mechanism for the PRV-based scheduler with five frames of scheduling granularity. The scheduler's granularity, or the total scheduling window size, is practically determined by the RT engine computing power (*i.e.*, beyond the scope of this paper). As will be seen in the following discussion, the PRV scheduler with its new extended scheduling time granularity might appear as a generalization for the single frame granularity SRV scheduler in optimizing the EE and delay jitter, however, with different execution procedure.

Considering a total of five frames (*i.e.*, M= 50 TTIs) for the scheduling time horizon

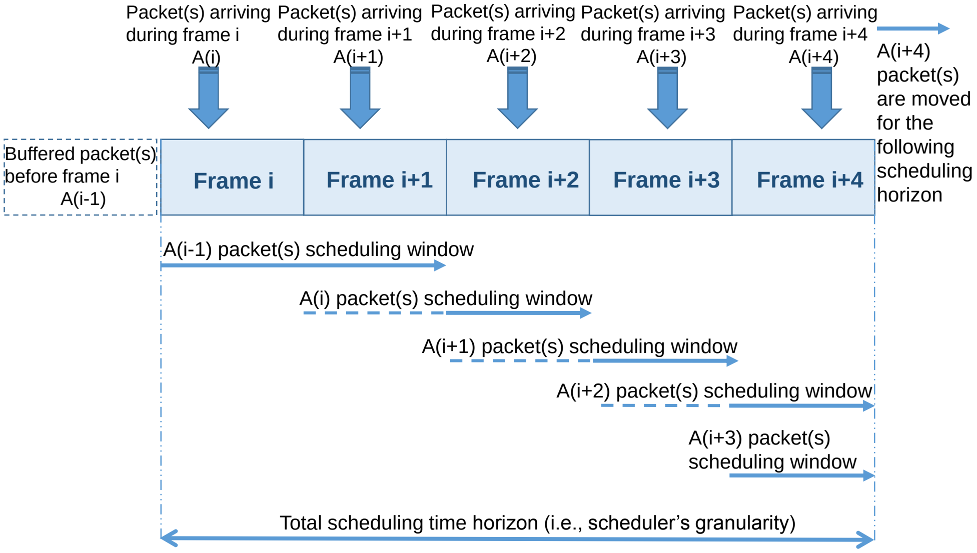

Packet(s) arriving during frame i
A(i)
Packet(s) arriving during frame i+1
A(i+1)
Packet(s) arriving during frame i+2
A(i+2)
Packet(s) arriving during frame i+3
A(i+3)
Packet(s) arriving during frame i+4
A(i+4)
A(i+4) packet(s) are moved for the following scheduling horizon
Buffered packet(s) before frame i
A(i-1)
Frame i
Frame i+1
Frame i+2
Frame i+3
Frame i+4
A(i-1) packet(s) scheduling window
A(i) packet(s) scheduling window
A(i+1) packet(s) scheduling window
A(i+2) packet(s) scheduling window
A(i+3) packet(s) scheduling window
Total scheduling time horizon (i.e., scheduler's granularity)

Figure 6.3: Sliding window predictive scheduling mechanism

as shown in Fig 6.3, each of the buffered packets inside a single UE buffer can be theoretically scheduled in any frame following its arriving frame (*i.e.*, causal time scheduling) within the scheduling horizon. However, to avoid having the scheduler bottle-necked, we limit the scheduling time horizon for each of the buffered packets to a smaller window of only two frames, for instance, following its arriving frame. The two frames window, then, slides in time with an overlap of one frame for subsequent frames arriving packets. The motivation behind employing the sliding window mechanism in scheduling packets could be explained as follows. For instance, having one of the buffered packets (*i.e.*, for single UE buffer) scheduled in the last frame of the scheduling time horizon will lead to having most (if not all) of the subsequent buffered packets potentially staying in the buffer for the following horizon, resulting in a buffer-overflow. The overflow problem is even more severe in case of demanding traffic types (*e.g.*, video and FTP). In addition to the buffer-overflow problem, which causes packet losses, most of the frequency resources in the beginning of the scheduling time horizon are potentially wasted. Therefore, the trade-off between better exploiting more information about the CSI in future frames - for reaching objectives (*i.e.*, optimizing the EE and delay jitter for VoIP, and EE for video and FTP) and satisfying constraints (*i.e.*, delay for VoIP and throughput for video and FTP) - and the spectral inefficiency and high queuing delays has to be carefully provisioned by setting proper sizes for the scheduling time horizon and the smaller sliding window.

Similar to the SRV-based schedulers, and based on the previous discussion, the heuristic algorithms for both of the connection and packet oriented versions of the proposed PRV scheduler are illustrated in Tables 6.4 and 6.5, respectively. The main structure is similar to that of the SRV-based algorithms except for the sliding window loop in line 3. In each iteration of the loop, for both of the PRV-CO and PRV-PO algorithms, the algorithm slides the scheduling window as explained in the previous paragraph. In particular, the algorithm creates the empty allocation matrix $\mathcal{R}$ (in line 4) for the frames of the observed window while excluding those allocated within the overlapped frame with the preceding iteration window within the same scheduling time horizon. The sliding window size is assumed to be two frames for both algorithms. The potential packets for each UE buffer that could be scheduled within the currently observed window are then determined in line 5 based on their arrival times. The WHILE loop then keeps iterating to schedule the packets found in line 5, for the current window, the same way as the SRV-based algorithms. The algorithm continues in the same fashion for the subsequent windows within the current scheduling time horizon until the loop of line 3 terminates. Finally, due to the fact the PRV algorithms have similar structures to those of the SRV, the computational complexities calculated in the previous subsection for the SRV apply to the PRV, however, with larger contribution of M in case of the PRV compared to the SRV (*i.e.*, SRV-PO is simpler than PRV-PO by a factor of M).

Table 6.4: PRV-CO algorithm

```
Require: $K$, $H_k$, $M$, $N$
Initialize $iter = 1$
for $j = j_o$ to $j_o + \frac{M}{10} - 1$ do
  Initialize allocations matrix$\mathcal{R}$ excluding previous iteration window allocations
  Find $A_k^h\ \forall k, h$ for the window of frames$j$ and $j+1$
  while $(isempty(\mathcal{R})$ AND $A_k^h \neq 0,\ \forall k, h)$ do
    Calculate $U_k^h\left(X_k^h(iter)\right)\ \forall k\ ,\ h$
    $[\mathrm{Id}_k, \mathrm{Id}_h] =$ Sort$(U_k^h\left(X_k^h(iter)\right),'ascend')$
    for $i = 1$ to $\sum_{k=1}^{K} H_k$ do
      Set $k = \mathrm{Id}_k(i),\ h = \mathrm{Id}_h(i)$
      Find $N^*,\ M^*$to satisfy FCFS
      Update $N^*,\ M^*$based on $D_{k,h}^{\max}$
      Sort$_{\mathbf{EE}}(N^*)$
      if $h_k \in D^*$ then
        Calculate $\left|\tau_{k,h}^{A}(a-1,a)\right|,\ \left|\tau_{k,h}^{D}(a-1,a))\right|,\ \ \forall a \in A_{k,h}^*$
        if length$(A_{k,h}^*) == 0$ then
          Find $\Psi_{k,h}^{a}(m,n),\ a = HOL_{k,h},\ n \in N^*,\ m \in M^*$
          Update $\mathcal{R}, A_{k,h}^*$
        else if length$(A_{k,h}^*) == 1$ then
          Calculate $\left|\Delta t_k^h(A_{k,h}^*, a\,|m)\right|, \forall n \in N^*,\ m \in M^*,\ a = HOL_{k,h}$
          Sort $(N^*, \left|\Delta t_k^h(A_{k,h}^*, a\,|m)\right|,'ascend')$
          Update $M^*$
          Find $\Psi_{k,h}^{a}(m,n),\ a = HOL_{k,h},\ n \in N^*,\ m \in M^*$
          Update $\mathcal{R}, A_{k,h}^*$
        else
          Calculate $\left|\tau_{k,h}^{A}(\left.A_{k,h}^*\right|_{last}, a)\right|,\ a = HOL_{k,h}$
          Calculate $\left|\tau_{k,h}^{D}(\left.A_{k,h}^*\right|_{last}, a\,|m)\right|,\ a = HOL_{k,h},\ m \in M^*$
          Sort$_{\mathbf{JITTER1}}(N^*)$
          Update $M^*$
          Find $\Psi_{k,h}^{a}(m,n),\ a = HOL_{k,h},\ n \in N^*,\ m \in M^*$
          Update $\mathcal{R}, A_{k,h}^*$
        end if
      else
        Find $\Psi_{k,h}^{a}(m,n),\ a = HOL_{k,h},\ n \in N^*,\ m \in M^*$
        Update $\mathcal{R}, A_{k,h}^*$
      end if
    end for
    Set $iter = iter + 1$
  end while
end for
```

Table 6.5: PRV-PO algorithm

```
Require: $K$, $H_k$, $M$, $N$
Initialize $iter = 1$
for $j = j_o$ to $j_o + \frac{M}{10} - 1$ do
    Initialize allocations matrix$\mathcal{R}$ excluding previous iteration window allocations
    Find $A_k^h$ $\forall k, h$ for the window of frames$j$ and $j + 1$
    while $(isempty(\mathcal{R})$ AND $A_k^h \neq 0,\ \forall k, h)$ do
        Calculate $U_k^h\left(X_k^h(iter)\right) \forall k\,,\ h$
        $[\mathrm{Id}_k, \mathrm{Id}_h] =$ Sort$(U_k^h\left(X_k^h(iter)\right),' ascend')$
        for $i = 1$ to $\sum_{k=1}^{K} H_k$ do
            Set $k = \mathrm{Id}_k(i)$, $h = \mathrm{Id}_h(i)$
            Find $N^*$, $M^*$to satisfy FCFS
            Update $N^*$, $M^*$based on $D_{k,h}^{\max}$
            Sort$_{\mathbf{EE}}(N^*)$
            if $h_k \in D^*$ then
                if length$(A_{k,h}^*) == 0$ then
                    Find $\Psi_{k,h}^a(m, n)$, $a = HOL_{k,h}$, $n \in N^*$, $m \in M^*$
                    Update $\mathcal{R}$, $A_{k,h}^*$
                else
                    Calculate $\left|\Delta t_k^h( A_{k,h}^*\Big|_{last}, a\,|m)\right|$, $\forall n \in N^*$, $m \in M^*$, $a = HOL_{k,h}$
                    Sort$_{\mathbf{JITTER2}}(N^*)$
                    Update $M^*$
                    Find $\Psi_{k,h}^a(m, n)$, $a = HOL_{k,h}$, $n \in N^*$, $m \in M^*$
                    Update $\mathcal{R}$, $A_{k,h}^*$
                end if
            else
                Find $\Psi_{k,h}^a(m, n)$, $a = HOL_{k,h}$, $n \in N^*$, $m \in M^*$
                Update $\mathcal{R}$, $A_{k,h}^*$
            end if
        end for
        Set $iter = iter + 1$
    end while
end for
```

6.6 Numerical Results

In this section, the performance of the two versions (*i.e.*, SRV and PRV) for our proposed scheduler with their connection (CO) and packet (PO) oriented set-ups is evaluated. Since the EE is one major objective for our proposed scheduler, the evaluation is conducted in comparison to the energy-efficient resource allocation scheme, namley green resource allocation (GRA), proposed in [78]. However, the GRA scheme was not designed to simultaneously deal with different QoS requirements. Therefore, and to ensure fair comparison, the GRA scheme is integrated with the utility-based scheduling scheme proposed in [81], namely fair class-based packet scheduling (FCBPS), to be able to deal with the heterogeneous traffic environment as our scheduler. For simplicity, the composite scheme is, thus, denoted as GRA-FCBPS. The investigations are carried out using a discrete event simulator built in MATLAB to simulate the real-time behavior of the considered traffic types. As explained in Section 6.2.1, VoIP, video streaming and FTP traffic types are considered in our simulations to address the delay, rate and best-effort QoS classes, respectively. The generation methods for the three traffic types are similar to those adopted in [81]. For video streaming, the minimum and maximum data rates are 64Kbps and 384Kbps, respectively. On the other hand, for FTP traffic, the maximum data rate is 128Kbps. Also, the packet delay budget for the VoIP traffic is 100msec. Without loss of generality, we simulate the system behavior due to an equal increase in the traffic connections requested for each traffic type. In particular, each UE is assumed to handle single traffic type. However, that assumption does not violate the scheduler model depicted in Fig. 6.2, since the scheduler deals with each traffic connection independently based on its utility as explained in Section 6.2.1.

The wireless channel for each UE is assumed to be frequency and time selective within each LTE frame (*i.e.*, fast and frequency selective fading channel). Thus, the channel is modeled as a quasi-static block Rayleigh fading channel [14]. In each TTI, 50 RBs are available for scheduling the UEs' traffic which corresponds to the 10MHz LTE channel.

Looking at the results depicted in figures 6.4, 6.5 and 6.6 which show the EE performance for VoIP, video and FTP users, respectively, three conclusions are deduced. First, the EE has a general decaying trend, for all types of users, with increasing system load. This is understood due to the scheduler's limited optimization ability in highly congested

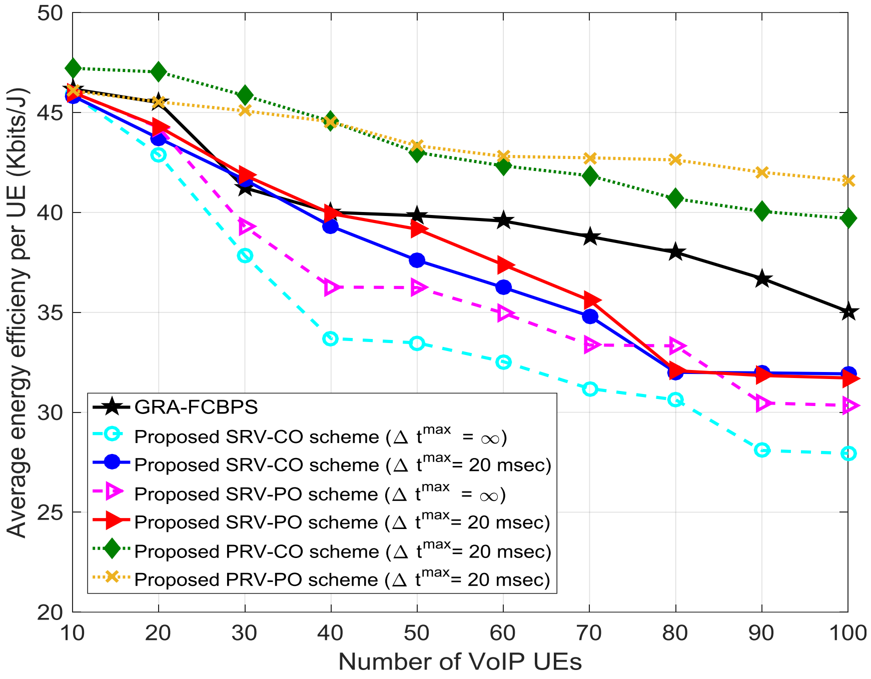


Figure 6.4: EE performance for VoIP UEs

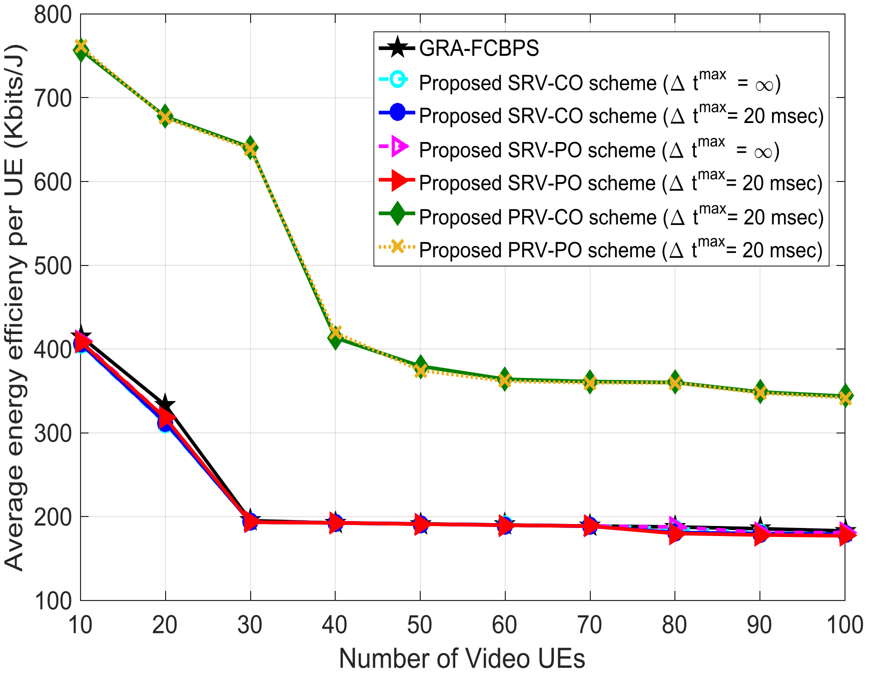


Figure 6.5: EE performance for Video UEs

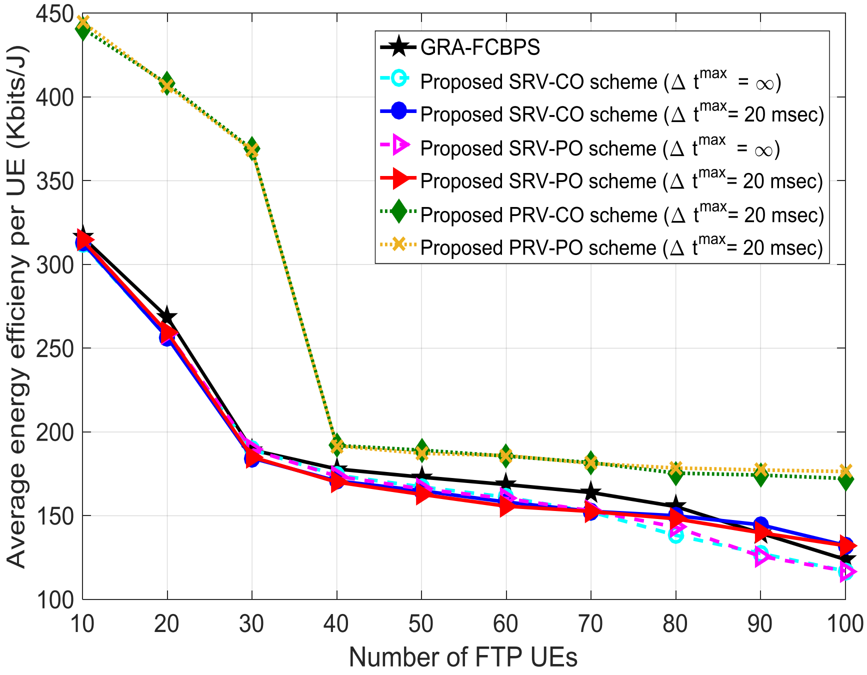


Figure 6.6: EE performance for FTP UEs

network situations during which the limited available resources become incapable of satisfying the increasing traffic demand. Second, the video users have the highest achieved average EE followed by FTP then VoIP. We attribute this to the fact that the UE receiver's circuit consumes the same base power in the wake-up state (*i.e.*, $P_{on} + P_{rx}$ as explained in Section 6.2.2) every TTI regardless of the type of traffic. Consequently, the traffic with the highest average rate certainly achieves the highest average EE. In our case, the video has the highest average rate (*i.e.*, 224Kbps), followed by FTP and VoIP. The third conclusion is regarding the relative EE performance of the proposed SRV-CO and SRV-PO schedulers compared to the GRA-FCBPS. It could be noticed that the difference is highly pronounced for VoIP users than that for video and FTP. This is because the proposed schedulers only optimize the EE in case of video and FTP traffics just as the GRA-FCBPS scheduler, while in case of VoIP the jitter is an additional objective to improve the quality of experience (QoE) perceived by VoIP UEs. Hence, we note a trade-off between the EE and delay jitter performances for the VoIP traffic.

In the following discussions, we start shedding a light on the EE/delay jitter trade-off for the VoIP traffic in the considered heterogeneous traffic environment as originally

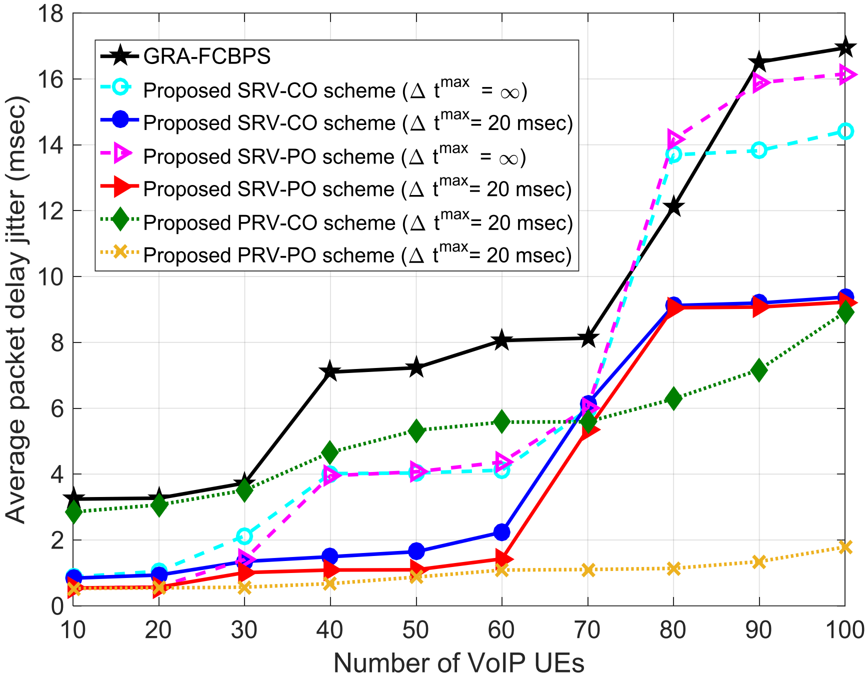


Figure 6.7: Average packet delay jitter for VoIP UEs

envisioned by our proposed framework. To avoid the discontinuity in discussions and confusing the reader, we first focus on the performance of our proposed SRV schedulers, in reference to the GRA-FCBPS, and then conclude by the RT-based PRV schedulers. Considering the VoIP traffic users, the EE results of Fig. 6.4 can be justified in conjunction with those obtained in Fig. 6.7. As shown in both figures, each of the proposed SRV-CO and SRV-PO scheduler is considered twice with different values for the jitter threshold (*i.e.*, $\Delta t^{\max}$) in (6.1). Two extreme cases were assumed. In the first case, the threshold is set to the highest possible value that is equal to ∞ which corresponds to the conventional delay metric function originally used by the FCBPS scheduler in [81]. The second case corresponds to our proposed VoIP utility in (6.1) having $\Delta t^{\max}$ set to an arbitrarily smaller value of 20msec. It should be noted that for both cases, as explained in the previous section, the proposed energy and jitter efficient allocation schemes are implemented. Thus, the first case (*i.e.*, $\Delta t^{\max} = \infty$) implies single stage of jitter optimization, while the second case (*i.e.*, $\Delta t^{\max}$ =20msec) corresponds to two stage jitter optimization. For both cases, the EE performance for both of the SRV-CO and SRV-PO is lower than that attained by the GRA-FCBPS scheduler. This can be directly justified by the remarkable jitter

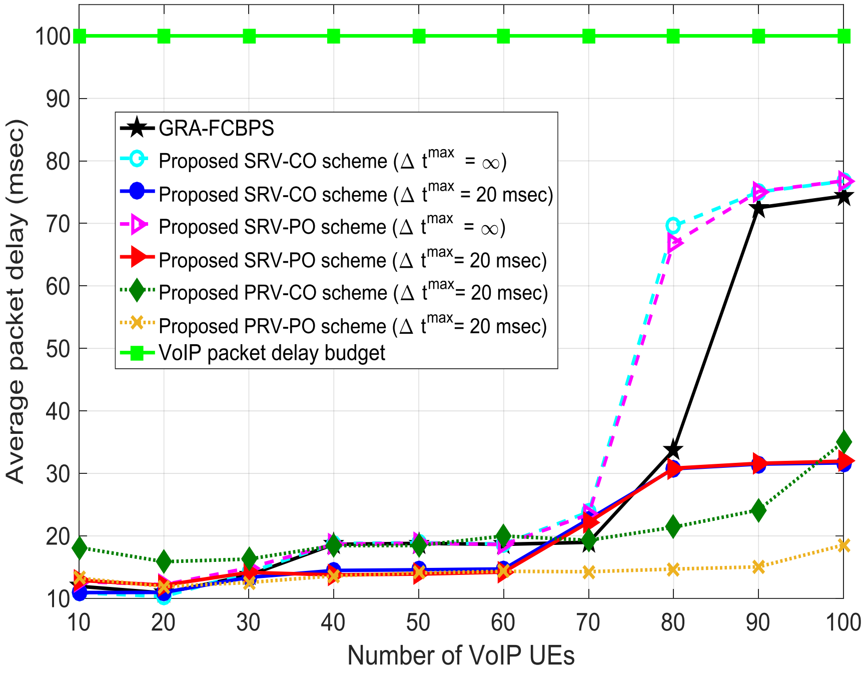


Figure 6.8: Average packet delay for VoIP UEs

improvement attained by our schedulers as illustrated in Fig. 6.7. At this point, it is imperative to indicate that the EE/jitter trade-off for VoIP allocations is attributed to the fact that our proposed jitter-efficient resource allocation algorithms (both the connection and packet oriented) tend to statistically expand the scheduling of resources in time to compensate for the delay jitter induced on the go. That behavior is obviously adverse to the EE scheduling strategy, explained in Section 6.2.2, which targets to minimize the number of allocated TTIs (*i.e.*, circuit wake-up periods of reception) to each UE while receiving the same amount of data. A minor impact for the jitter-efficient allocations of VoIP users is observed on the EE of video and FTP users as depicted in figures 6.5 and 6.6, respectively. That is, only a slight drop in the attained EE for video and FTP users arising from the time spread VoIP allocations which might span TTIs initially allocated in full to video and FTP users. In other words, the same video or FTP UE might takes less number of RBs during single TTI while consuming the same power. It is worth mentioning, that all the obtained results are averaged over 50000 LTE frames. When attempting to increase the simulation length (in frames) the output results did not show any change, however, with a dramatic increase in the computation time.

It can also be noticed from the results obtained in Figures 6.4 and 6.7 that the double stage jitter optimization for our proposed SRV schedulers (*i.e.*, $\Delta t^{\max}$ =20msec) is substantially boosting both of their EE and jitter performances compared to their single stage optimization arrangement (*i.e.*, $\Delta t^{\max} = \infty$). The reason is that adding the delay jitter parameter to the VoIP metric function in (6.1) makes it more sensitive to changes resulting in more aggressive utility requirements (*i.e.*, prioritization) to VoIP users compared to video and FTP users. This can be further illustrated by the improved delay performance obtained in Fig. 6.8. Therefore, by allowing jitter-dependent VoIP utility, the proposed schedulers strike better EE/delay jitter trade-off compared to the single stage jitter optimization (*i.e.*, jitter-independent utility). From another perspective, the SRV-PO scheme generally shows better performance (*i.e.*, EE/delay jitter trade-off) compared to the SRV-CO. This is due to the fact that the SRV-CO algorithm is performing jitter optimization with respect to the entire delay history of scheduled packets. Having in mind that some of the scheduled packets might have attained high jitter values, the jitter optimization of subsequent packets gets consequently negatively influenced. This is a problem that does not exist in the SRV-PO algorithm which only provisions the last scheduled packet departure time when optimizing the jitter for the following packet. This can be seen obviously in the double stage jitter optimization case where the SRV-PO algorithm is clearly outperforming the SRV-CO algorithm in terms of EE and jitter. However, in the case of single stage jitter optimization (*i.e.*, $\Delta t^{\max} = \infty$), some discrepancies are noted to be taking place. The SRV-PO scheme performs all the way better than the SRV-CO scheme in terms of EE as shown in Fig. 6.4. Nevertheless, in terms of jitter, as observed Fig. 6.7, it only outperforms at light load (*i.e.*, 10, 20 and 30 VoIP users). At moderate load (*i.e.*, 40, 50, 60 and 70 VoIP users) the performance is almost equal to the SRV-CO. Then at high load (*i.e.*, 80, 90 and 100 VoIP users), SRV-CO showed even better jitter performance than SRV-PO. The sudden jump in jitter in Fig. 6.7 (*i.e.*, at 80 VoIP UEs) could be seen happening in the delay depicted in Fig. 6.8 at the same point. This behavior conveys an overloaded system situation during which the effect of the inter-class distinguishing parameter β_h in prioritizing the VoIP traffic becomes minor compared to the utility of the growing number of heavy traffic video and FTP users. Hence, a lower priority will be rather given most of the time to the light VoIP traffic for the sake of heavy rates video and FTP traffics. The stable jitter performance trend illus-

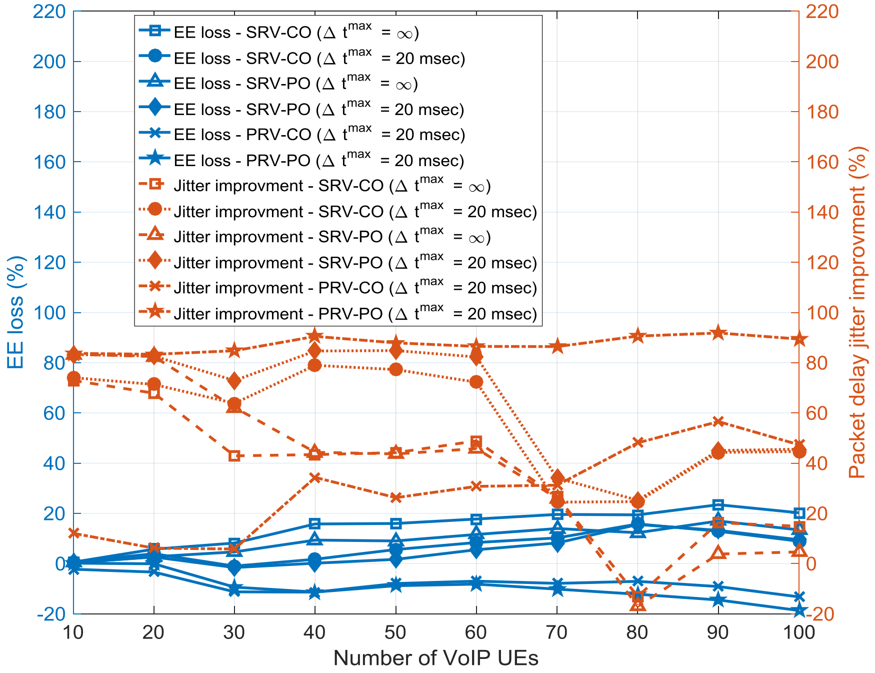


Figure 6.9: EE/delay jitter trade-off for VoIP UEs

trated in Fig. 6.7 for the double stage optimization arrangement, which adds more priority to VoIP users over others, of our proposed schedulers essentially supports the previous conclusion. To summarize all the previous discussions about the SRV schedulers results obtained in figures 6.4 and 6.7, Fig. 6.9 shows the EE loss versus the delay jitter improvement trade-off attained by the proposed algorithms, in their jitter-dependent and independent utility arrangements, with respect to the GRA-FCBPS scheduler performance. The fluctuations noticed in the results directly represent the relative jitter performance of our proposed schedulers with respect to the GRA-FCBPS scheduler as illustrated in Fig. 6.7.

For the video and FTP users, the results reported in figures 6.10 and 6.11 confirm that the improvement in the EE and delay jitter performances for VoIP users achieved by our proposed SRV schedulers does not harmfully affect the video and FTP users in terms of the achieved throughput. However, the drop in the attained throughput for our proposed SRV schedulers, especially in case of FTP users, pertains to the jitter allocation strategy, as previously discussed, which might be sometimes spectrally inefficient in favor of VoIP users. In terms of fairness, as expected all the results depicted in figures 6.12, 6.13 and 6.14 show comparable satisfactory intra-class fairness for the VoIP, video and FTP users,

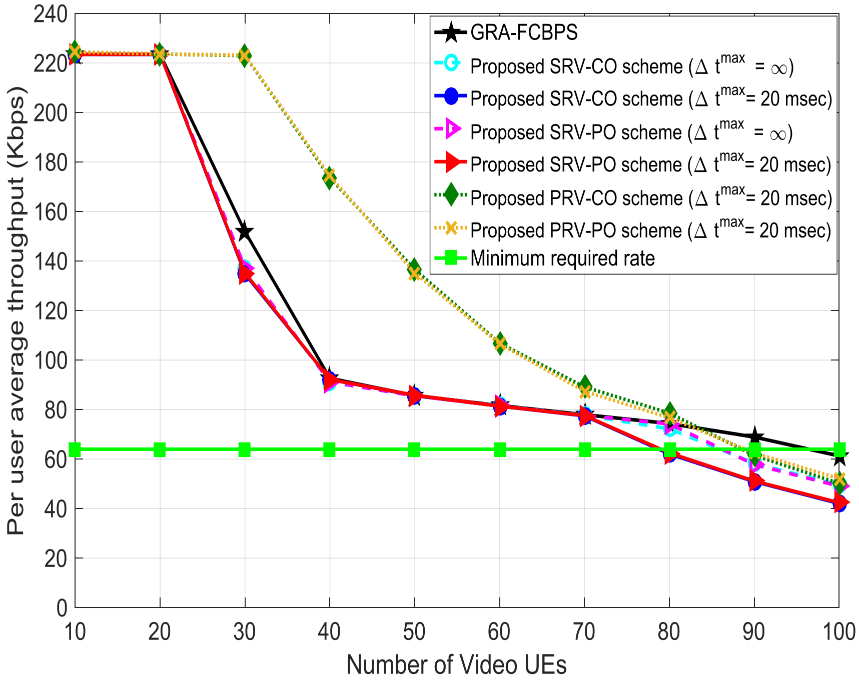


Figure 6.10: Average throughput for Video UEs

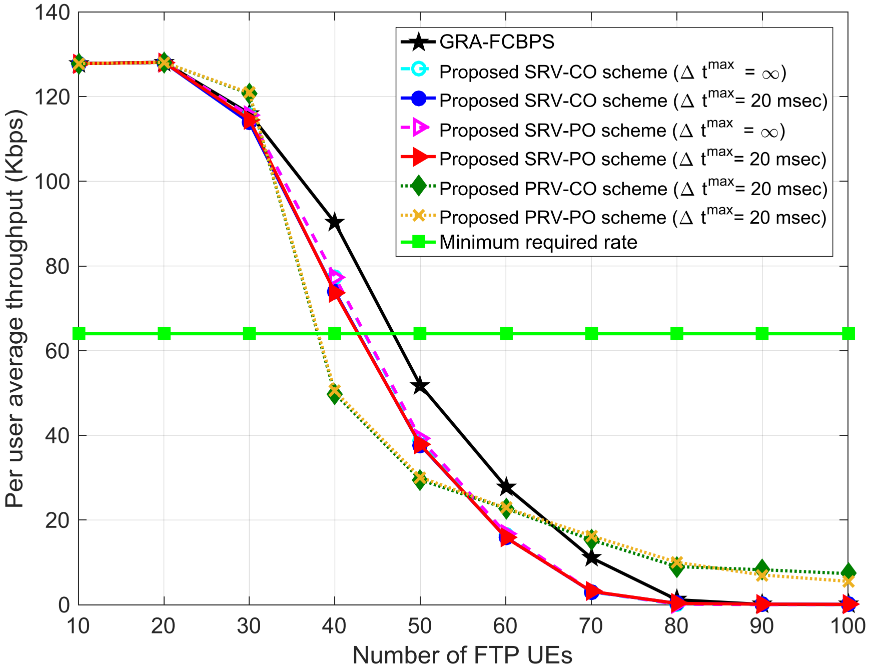


Figure 6.11: Average throughput for FTP UEs

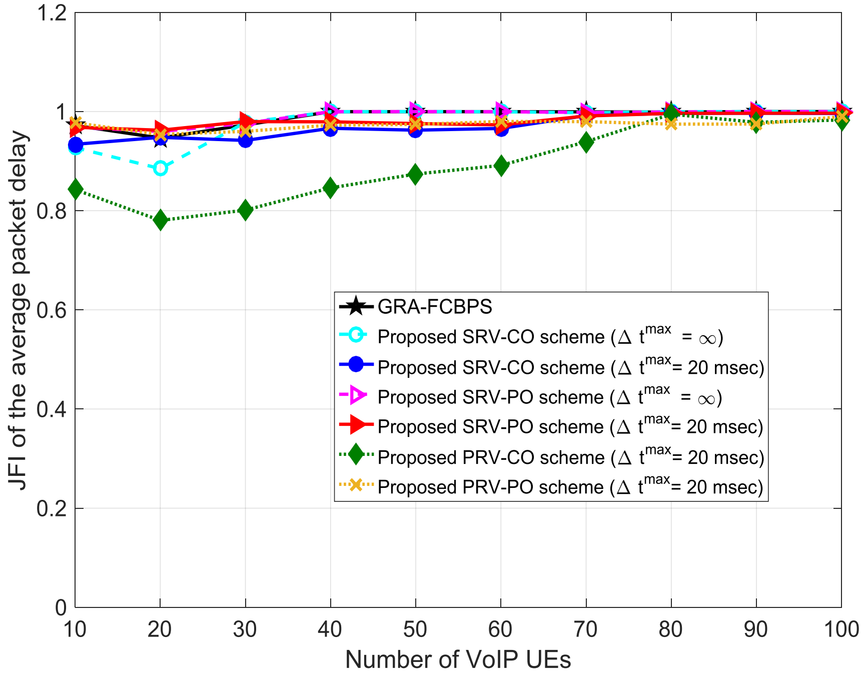


Figure 6.12: JFI for VoIP UEs

respectively. Since all the implemented schedulers are based on the utility-based strategy discussed in Section 6.2.1, the intra-class fairness was expected. The fairness is measured using the Jain's Fairness Index (JFI) [107]. For VoIP users, the JFI is calculated in terms of the average packet delay experienced by each user. For video and FTP, the JFI is calculated based on the users' average attained throughput. The fairness drop observed for FTP traffic (*i.e.*, lowest priority traffic) is due to the scarcity of the spectral resources to carry the FTP users traffic at high network load. As a result, some FTP users might starve compared to others. Hence, the fairness gets slightly affected.

When investigating the peak performance of our proposed scheduler in its predictive version (*i.e.*, PRV) supported by the cloud-based RT prediction (as discussed in Section 6.2) and the proposed sliding window mechanism (*i.e.*, explained in the previous section), we were able to obtain the following. For VoIP users, a considerable improvement for both the EE and delay jitter performances is achieved as illustrated in figures 6.4 and 6.7, respectively. The improvements are due to the increased scheduling time horizon (*i.e.*, 5 frames with sliding window of 2 frames) which allows the scheduler to better exploit the channel diversity gain over time and explore bigger solution space to solve the optimization prob-

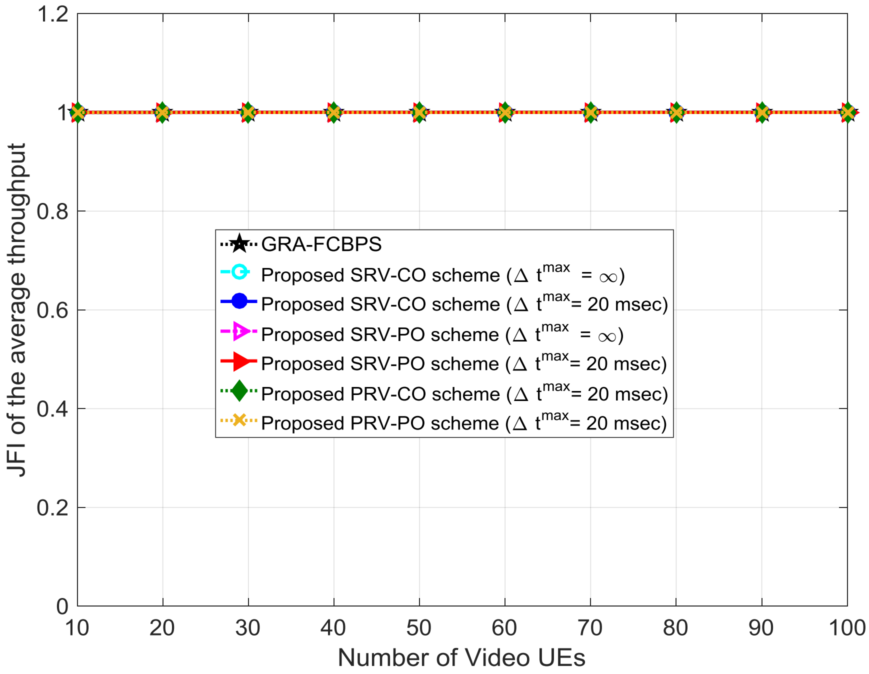


Figure 6.13: JFI for Video UEs

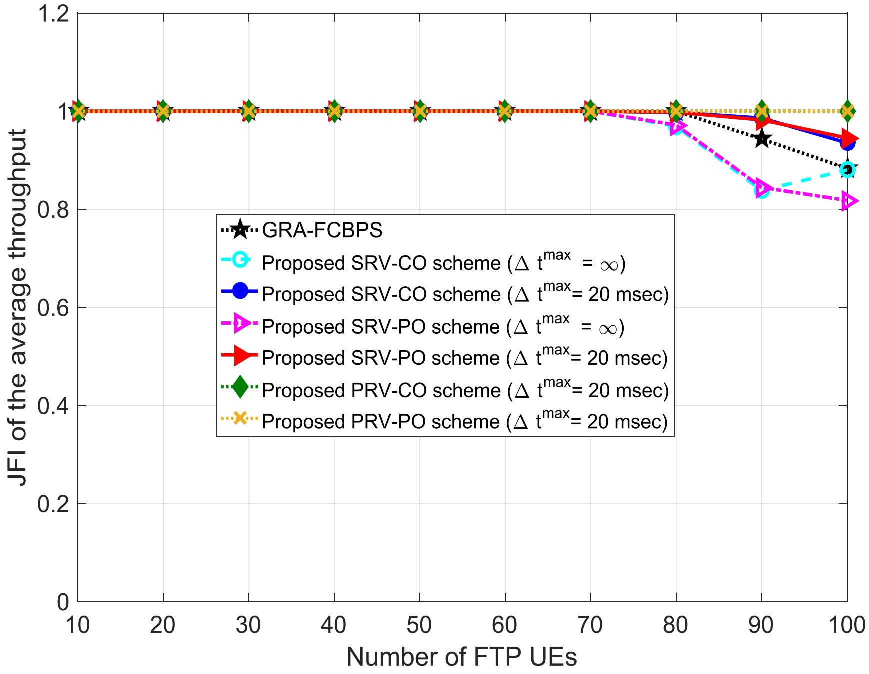


Figure 6.14: JFI for FTP UEs

lem more efficiently. As in the case of SRV schedulers, the PRV-PO algorithm showed better performance compared to its PRV-CO counterpart, however, with wider gap. The increased performance gap is regarded to magnifying the jitter relative inefficiency of the CO-based algorithm in case of the PRV scheduler. In particular, the PRV-CO has more freedom to inaccurately move its RB allocations to future frames to compensate for the delay jitter of the observed packet based on the whole history of scheduled packets. This intuition is supported by the relative large delay attained by the PRV-CO algorithm compared to the PRV-PO as depicted in Fig. 6.8. In terms of the EE/delay jitter trade-off, both algorithms show negative values for the EE loss percentage which implies an EE improvement, that is close to 20% as illustrated in Fig. 6.9. However, the PRV-PO algorithm is able to maintain nearly constant jitter improvement by approximately 90% relative to the GRA-FCBPS, whereas 60% at most is attained by the PRV-CO algorithm. Furthermore, the performance gap discussed above, between the PRV-PO and PRV-CO algorithms, is found also in the fairness among VoIP users. As demonstrated in Fig. 6.12, the redundant jitter mechanism for the PRV-CO scheduler has obviously affected the uniformity of the average packet delay experienced by each VoIP user leading to a drop in the JFI to 0.8. This drop is clearly found at light network load in which the redundancy of the CO jitter-efficient resource allocation algorithm is highly pronounced for each of the existing VoIP users. On the opposite side, the precise jitter mechanism employed in the PRV-PO algorithm leads to sustained fairness with JFI very close to 1 irrespective of the network load.

The performance gain for the PRV schedulers was not limited to VoIP users. Video and FTP users were also able to acquire boosted EE and throughput performances. The results obtained in figures 6.5 and 6.6 demonstrate dramatic increase in the EE for video and FTP UEs, respectively. However, at high network load, FTP UEs have experienced sudden drop in their EE performance due to their low inter-class priority compared to video UEs which relatively retain more resistance to the EE drop. From another perspective, the proposed PRV schedulers generally attain higher EE compared to the SRV and GRA-FCBPS schedulers. The EE results for video and FTP UEs can be directly justified with their corresponding throughput results obtained in figures 6.10 and 6.11. In Fig. 6.10, the video UEs experience as significant throughput increase as that for the EE in Fig. 6.5. As in the case of VoIP, this improvement is attributed to the channel diversity gain over future frames utilized by the PRV schedulers. In particular, the scheduler was able to find better TTI(s) to receive more average packets per each video UE while consuming the same circuit energy. As expected, the throughput increase for video UEs is directly translated as a decay in the attained throughput by the FTP users as portrayed in Fig. 6.11. This dependency is due to the higher priority for video users which require an average rate (*i.e.*, 224Kbps) approximately twice that required by FTP users (*i.e.*, 128Kbps), and the bounded cell capacity (*i.e.*, number of RBs/TTI) even at high spectral efficiency. Despite the throughput drop in moderately loaded network, the PRV schedulers were able to maintain improved throughput performance for FTP users compared to the SRV and GRA-FCBPS schedulers at light and high loaded network situations. Finally, the PRV schedulers were able to maintain high JFI among video users as illustrated in Fig. 6.13. Meanwhile, the same high fairness is also achieved by the PRV schedulers in case of FTP users as depicted in Fig. 6.14, which confirms the equal throughput improvement among all users during network congestion compared to the SRV and GRA-FCBPS schedulers.

6.7 Chapter Summary

In this chapter, we developed a framework for implementing a QoS-aware energy and jitter efficient scheduling methodologies in downlink OFDMA heterogeneous traffic cellular systems.

Firstly, we utilized our previously proposed C-RAN based predictive scheduling system to provide a more detailed model in case of heterogeneous traffic networks. Secondly, based on the popular utility-based inter-class prioritization scheme, we proposed a new metric function which captures the packet's delay and delay jitter QoS metrics. The function targeted optimal jitter performance within the packet's delivery delay threshold for the real-time traffic class. Thirdly, we formulated a BIP problem which optimizes the LTE UE's receiving EE for delay-sensitive, rate-sensitive and best-effort QoS classes concurrently. In the case of delay-sensitive class, the formulation was of a multi-objective structure having the delay jitter as the second objective besides the EE and constrained by the packet delay budget. For the rate-sensitive and best-effort classes, the formulation targeted only the EE as a single objective while setting a minimum rate constraint for the rate-sensitive QoS class. Fourthly, due to the inherent intractability of the optimal formulation, four different heuristic algorithms were proposed to find a sub-optimal solution for the problem with reasonable computational requirements. Two of the proposed algorithms belong to the traditional short range schedulers which work with a single frame time granularity (at most). The other two algorithms belong to the predictive scheduling class and were capable of allocating resources in multiple future frames by utilizing the model proposed in the first part of our work. All the proposed algorithms employed the jitter optimization in two stages. The first stage encompassed the proposed jitter-based utility function, whereas the second stage involved two proposed jitter-efficient resource allocation algorithms. Furthermore, a sliding window mechanism was proposed to allow the heuristic algorithms profit from the future predicted CSI (*i.e.*, provided by the pool of ray tracing engines residing in the cloud) without the need of implementing a traffic prediction mechanism.

To evaluate the performance of our proposed schedulers, extensive numerical simulations were conducted in comparison with existing schedulers. The first part of the results demonstrated the ability of our proposed short range schedulers to remarkably improve the delay jitter, for the real-time traffic, on the expense of the EE while maintaining the delay bounds. This is in addition to maintaining comparable EE and throughput performances for rate-sensitive and best-effort traffic types. In the second part, the predictive versions of our proposed scheduler were able to strike a dramatic improvement for the EE/delay-jitter trade-off, compared to existing schedulers and our short range schedulers, in case of the delay-sensitive traffic. Furthermore, the EE and throughput performances were also substantially improved for the rate-sensitive and best-effort traffics. The improvements were due to the predictive scheduler's capability of exploiting future channel conditions in making better decisions compared to traditional short range schedulers. Finally, since employing the utility-based prioritization scheme, all the proposed schedulers showed high degree of intra-class fairness for each of the considered traffic classes.

Chapter 7
Conclusion

The massive evolution in today's cellular technologies together with the myriad of available multimedia applications and services pose various challenges on wireless network designers such as maintaining the system's spectral and energy efficiencies at acceptable levels. What makes such challenges increasingly intricate is the rapid expansion in the wireless market volume and the number of cellular users with firm QoS requirements. Therefore, facing such challenges requires non traditional solutions. In this dissertation we proposed one promising agile solution utilizing a ray tracing (RT)-based predictive scheduling methodology, which we believe to be capable of addressing such challenges. The dissertation started by providing a solid understanding of the ray tracing problem and its implementation challenges. The study has further delivered a complete MATLAB ray tracer which could be used for benchmarking future hardware implementations or a simulator on its own. In the second part of the dissertation, the RT-based scheduling system model was proposed and heavily investigated in addressing the above challenges using optimization techniques. In the rest of this chapter, the work completed in this dissertation and its conclusions are summarized. The chapter concludes by recommendations for extending the conducted research.

7.1 Thesis Summary

An extensive survey on the ray tracing propagation prediction model was conducted in Chapter 2. The survey presented the ray tracing applications horizon, and the motivation behind the dissertation invoking the state-of-the-art prediction model as a potential solution for future cellular architectures. Implementation related topics were also provided such as theoretical description, algorithmic approaches and acceleration techniques. Finally, the survey concluded by introducing the typical architecture of a ray tracing engine and its implementation on MATLAB. The MATLAB ray tracer was then validated with a commercial propagation prediction tool and its accuracy was proven. The MATLAB ray tracer could serve as configurable simulator that can be easily altered or expanded according to the user's needs. Furthermore, it could be used to benchmark future software or hardware implementations.

In Chapter 3, the initial proposal for the RT-based predictive scheduling approach was presented in a simple TDMA system. The chapter's framework encompassed an analytical evaluation for the channel outage probability as a function of the link parameters (*e.g.*, rate and power) adaptation time horizon, and a throughput optimization problem. The analysis has given insights into the importance and motivation of utilizing the ray tracing prediction to adapt the transmission parameters frequently enough to maintain target channel outage probability compared to the traditional block fading assumption (*i.e.*, channel SNR is constant within a certain number of transmission blocks). The numerical results showed that the outage probability could be unfavorably high with long adaptation time horizons and high mobile speeds. The second part of the chapter has build upon the conclusions drawn from the outage analysis, in the first part, to propose an RT-based throughput efficient scheduler subject to fairness among users in TDMA systems. The optimization problem was formulated as a binary integer programming (BIP) problem. The complexity of the optimal scheduler was then addressed by a low complexity heuristic algorithm which showed a comparable performance with its optimal counterpart. Numerical simulations demonstrated remarkable improvement in the cell's average achieved throughput for our proposed scheduler compared to the state-of-the-art maximum throughput (MT) scheduler.

The simple predictive scheduling approach, proposed in Chapter 3, was further developed in Chapter 4 to fulfill the optimal energy efficient performance for the LTE user equipments (UEs) subject to QoS requirements. The work constituted three phases. In the first phase, a C-RAN-based predictive downlink scheduling system was proposed. The objective was to optimize the UE's receiver circuit wake-up intervals in longer time horizons, compared to the state-of-the-art LTE DRX mechanism, subject to effective bandwidth constraints. The long-term circuit operation time optimization was based on ray tracing predictions for future channel conditions. The problem formulation was then presented in the second phase as a multi-objective BIP optimization problem. Furthermore, the formulation has undergone rigorous analysis and a series of modifications until reached the final practical form which accounted only for the factors dominating the UE power consumption budget in the downlink. In the third phase, the computational complexity of the optimal scheduler was relaxed by designing a low complexity heuristic algorithm. Finally, extensive numerical simulations were conducted not only to compare our proposed scheduler with another existing scheduler but also to provide the performance bounds of our scheduler in two different channel environments. The first environment involved a fast fading QSBR channel which models high mobility situations. On the other hand, addressing low mobility scenarios, the second environment was based on a realistic ray tracing channel prediction experiment performed by a commercial propagation tool in the north of centretown of Ottawa city in Canada. In the fast fading environment, our scheduler was able to exploit the rapidly changing channel conditions to improve both the UEs' energy efficiency (EE) and the capacity of the system to serve more users (*i.e.*, spectral efficiency) by 62.47% and 60%, respectively. In the slow fading environment, however, the scheduler was able to only improve the EE due to the low channel diversity gain within the selected scheduling granularities by 56.6%.

The work presented in Chapter 5 was divided into two parts. Motivated by the few efforts discussed in the literature, in the first part, the delay jitter in communication networks has been rigorously studied. The study has provided an extensive analytical model for the delay jitter in the simple queuing system with single queue of infinite length and single server under different utilization levels and packets' service and interarrival statistics. Unlike traditional independent and identically distributed (iid) models, our analysis focused on the correlated nature of the service time intervals. The underlying objective of the analytical study was to enable jitter-efficient packet scheduling schemes in LTE networks. In the second part of the chapter, the insights gained in the first part were utilized to design two energy and jitter efficient heuristic schedulers for VoLTE applications. Unlike most published works which only considered the average packet delay as the only QoS metric for VoIP traffic, our framework has set the delay jitter as a second objective besides the EE subject to fixed delay budget in LTE systems. This is due to the significant impact for the delay jitter in determining the QoS levels achieved by real-time applications. Thus, the resource allocation optimization problem was presented as a multi-objective BIP formulation. Numerical results revealed a novel perspective for a trade-off between the LTE UE's EE and its delay jitter performance. The results showed that the proposed schedulers were able to strike a better trade-off between the EE and delay jitter compared to other existing schedulers.

Combining the cloud-based predictive scheduling model proposed in Chapter 4, and the delay jitter modelling and optimization conducted in Chapter 5, Chapter 6 presented a thorough study for optimizing the EE and the packet delay jitter for real-time applications in the downlink of heterogeneous traffic LTE networks. The chapter elaborated on the C-RAN predictive scheduling model proposed in Chapter 4 to provide an insight about utilizing it in LTE cellular networks serving various QoS requirements. The new model considered the widely used utility-based prioritization scheme to serve three distinct QoS classes, denoted as delay-sensitive, rate-sensitive and best-effort classes. For the delay-sensitive QoS class, a new metric function was proposed which combines the delay jitter and the average packet delay to shape the priority of the corresponding traffic connections in the utility pool. The underlying objective of the metric function was to enable optimal jitter performance for real-time traffic connections subject to fixed packet delay budget. The resource allocation problem was formulated as a BIP problem by setting the UE's receiving EE as a global objective for all of the considered traffic classes. However, the packet delay jitter was defined as a concurrent objective only for the delay-sensitive traffic connections to ensure high quality real-time performance. This is unlike most of the published works in the literature which insufficiently focused only on the average packet delay metric. The formulation captured the maximum permissible packet delay and the minimum acceptable data rate constraints for delay and rate-sensitive traffic types, respectively. To efficiently solve the optimal formulation, which was found to be intractable, four different heuristic algorithms were proposed. Two of the proposed algorithms solved the problem, as most conventional schedulers proposed in the literature, within a single LTE frame time granularity. The other two algorithms utilized our proposed cloud-enabled predictive scheduling system (*i.e.*, presented in the beginning of the chapter) to solve the problem in longer time horizon (*i.e.*, spanning multiple future LTE frames). The proposed heuristic algorithms utilized the initially proposed metric function, and employed two different jitter-efficient resource allocation schemes to optimize the packet delay jitter for the delay-sensitive traffic class. Moreover, to avoid limiting the performance of our proposed schedulers, a sliding window mechanism was proposed to alleviate the short term knowledge of the buffer state information (BSI) (*i.e.*, assuming no traffic prediction) compared to the RT-based long term knowledge about the channel state information (CSI). Finally, the numerical results showed that our proposed schedulers were able to remarkably increase the UE's EE for all of the considered traffic classes, substantially minimizes the delay jitter for real-time applications, boosts the throughput performance for the rate-sensitive and best-effort traffic types, and maintains intra-class fairness among users.

7.2 Future Work

The explosive expansion in multimedia services over mobile broadband systems and the ever-increasing mobile users number are leading the wireless world to an imminent major shift to the 5G technology. As a result, numerous potential research areas are on their way to deployment. This section proposes some research directions as an extension for the vision introduced in this dissertation towards agile wireless resource scheduling for future cellular technologies.

7.2.1 Interference-Aware Scheduling for LTE-A D2D Communication

Unlike the 4G, the 5G system's architecture will utilize the user's device intelligence to support direct device-to-device (D2D) connectivity, for faster and efficient data transfer, side-to-side with the regular cellular links (*i.e.*, base-station relayed) depending on the situation. The motivation behind this new architecture is to offload huge traffic volume from the core network which consequently reduces the energy consumed and cost of data transmission for network operators. The design of the 5G network will also exploit the benefits of reducing the cell size [5] for more efficient spatial reuse of the spectrum leading to higher data rates, lower delay and energy consumption.

The D2D communication is classified into inband D2D and outband D2D. The inband D2D works in the same licensed spectrum with the cellular systems, while the Outband uses the unlicensed industrial, scientific and medical (ISM) spectrum (*e.g.*, Wi-Fi, Bluetooth, ZigBee and Ultra Wideband systems). It is noted in the literature that the inband D2D, also known as underlay D2D, is of greater interest compared to the outband. This pertains to its high spectral efficiency as a result of sharing the system's spectral resources between D2D and cellular users. However, the interference co-ordination between D2D and cellular terminals turned out to be a challenging problem in underlay D2D. Few works [130, 131, 132] were found tackling the problem by employing particular resource and power allocation schemes (*e.g.*, interference-aware scheduling, fractional frequency reuse scheduling), spectrum sharing using cognitive techniques, and massive MIMO transmission techniques. Thus, studying the impact of the D2D transmissions (*e.g.*, in cases of fixed transmit power and fixed target SNR schemes) on the power margin of the cellular network, and the utilization of long-term traffic and channel predictions on existing interference-aware scheduling schemes are two potential topics to be researched.

7.2.2 Virtualization in Next Generation Radio Access Network

The term "virtualization" is currently receiving wide popularity in all sectors including industry, research and standardization organizations. This is due to the numerous benefits of deploying this technology [133]. From the business point-of-view, virtualization will reduce the networks' roll-out costs and operating expenses, increase energy savings, and maximize the infrastructure providers' (InPs) revenue. Furthermore, it will improve the QoS delivered to end-users by increasing the competition in the wireless market when allowing small investors to enter the business market. From the technical point-of-view, wireless network virtualization employs sharing single infrastructure which contains sliced wireless and physical resources among several mobile network operators (MNOs). InPs perform the slicing to allow each operator has its assigned slice to be able to serve its own subscribers. The significant gain of deploying this common shared architecture is that the MNOs will be able to share their entire spectral resources which results in maximized network efficiency, high energy savings and improved QoS.

Many challenges, which need to be further investigated, still face this ambitious technology. For instance, fulfilling service contracts between different MNOs and fairness among their users employing dynamic and efficient resource sharing schemes [134, 135] is crucial. These schemes should deal with different MNOs (*i.e.*, using different scheduling policies) under highly changing network situations (*e.g.*, traffic demands and relative channel conditions between different MNOs subscribers). In these situations, our studied predictive scheduling model could be utilized on a high level, across different MNOs, to enable efficient and fair spectral sharing using the knowledge of future channel conditions.

7.2.3 On The Application of Ray Tracing Prediction in Physical layer Security

Data communications security and privacy have always been addressed as a fundamental aspects for designing robust and reliable communication networks. Security involves preventing unauthorized interceptors (*e.g.*, eavesdroppers) from accessing sensitive information, either by tapping or modifying, transmitted over the channel while still delivering content to the legitimate partners. For this, state-of-the-art encryption algorithms have been developed over the years and widely implemented in the top layers of the network protocol stack (*e.g.*, transport layer). Those algorithms are commonly executed independent of the physical layer channel impairments and errors occurred to the transmitted data. On the other hand, securing data in the physical layer and specifically over wireless channels has also been a subject of interest since Wyner has reported his wiretap channel model in [136]. Based on Wyner's model, several research works [137, 138] have supported the idea that perfect secrecy in wireless data transmission can be accomplished when utilizing the channel capacity difference (*i.e.*, secrecy capacity), for quasi-static fading models, between legitimate receiver and eavesdropper. The key idea here is that, by knowing the CSI for the legitimate receiver and an estimate for the eavesdropper at the transmitter side, suitable wiretap codes can be selected accordingly to ensure transmission favorable to the legitimate receiver while weakening the eavesdropper's ability to decode the transmitted data.

It is worth noting that none of the published works has considered the wireless channel security in dynamic environments such as V2V and V2I, where the channel response is changing rapidly and significantly. These scenarios require frequent channel estimation to be able to adapt the transmission rate for the legitimate receiver's sake. As a result, utilizing high speed ray tracing engine for controlling the transmission rate on the fly, identifying the reception zones and measuring the secrecy capacity outage probability along the transmitter's route is an important topic to be considered.

References

[1] M. Kalil, A. Shami, and A. Al-Dweik, "QoS-aware power-efficient scheduler for LTE uplink," *IEEE Transactions on Mobile Computing*, vol. 14, no. 8, pp. 1672–1685, Aug. 2015.

[2] J. Schmittler, S. Woop, D. Wagner, W. J. Paul, and P. Slusallek, "Realtime ray tracing of dynamic scenes on an FPGA chip," in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, ser. HWWS '04, 2004, pp. 95–106.

[3] Smart Insights, Available at: http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/.

[4] C. X. Wang, F. Haider, X. Gao, X. H. You, Y. Yang, D. Yuan, H. M. Aggoune, H. Haas, S. Fletcher, and E. Hepsaydir, "Cellular architecture and key technologies for 5g wireless communication networks," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 122–130, Feb. 2014.

[5] S. Mumtaz, K. Saidul Huq, and J. Rodriguez, "Direct mobile-to-mobile communication: Paradigm for 5G," *IEEE Wireless Communications*, vol. 21, no. 5, pp. 14–23, Oct. 2014.

[6] H. Seo, K. D. Lee, S. Yasukawa, Y. Peng, and P. Sartori, "LTE evolution for vehicle-to-everything services," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 22–28, June 2016.

[7] V. J. Kotagi, R. Thakur, S. Mishra, and C. S. R. Murthy, "Breathe to save energy: Assigning downlink transmit power and resource blocks to LTE enabled IoT networks," *IEEE Communications Letters*, vol. 20, no. 8, pp. 1607–1610, Aug. 2016.

[8] A. Gomez-Andrades, P. Muñoz, E. J. Khatib, I. de-la Bandera, I. Serrano, and R. Barco, "Methodology for the design and evaluation of self-healing LTE networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6468–6486, Aug. 2016.

[9] P. Mach, Z. Becvar, and T. Vanek, "In-band device-to-device communication in OFDMA cellular networks: A survey and challenges," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 1885–1922, Fourthquarter 2015.

[10] S. Stefania, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution*. John Wiley & Sons Ltd., 2011.

[11] C. Cox, *An Introduction to LTE*. John Wiley & Sons Ltd., 2012.

[12] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Downlink packet scheduling in LTE cellular networks: Key design issues and a survey," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, pp. 678–700, Second 2013.

[13] R. Ruby, V. C. M. Leung, and D. G. Michelson, "Uplink scheduler for sc-fdma-based heterogeneous traffic networks with qos assurance and guaranteed resource utilization," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4780–4796, Oct. 2015.

[14] J. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2001.

[15] G. Song, Y. Li, and L. J. Cimini, "Joint channel- and queue-aware scheduling for multiuser diversity in wireless OFDMA networks," *IEEE Transactions on Communications*, vol. 57, no. 7, pp. 2109–2121, July 2009.

[16] R. Atawia, H. Abou-zeid, H. S. Hassanein, and A. Noureldin, "Joint chance-constrained predictive resource allocation for energy-efficient video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1389–1404, May 2016.

[17] Q. D. Vo and P. De, "A survey of fingerprint-based outdoor localization," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 491–506, Firstquarter 2016.

[18] A. S. Glassner, *An introduction to ray tracing*. San Diego, CA, USA: Academic Press, 1989.

[19] Remcom Inc., Available at: http://www.remcom.com/wireless-insite.

[20] K. R. Schaubach, N. J. Davis, and T. S. Rappaport, "A ray tracing method for predicting path loss and delay spread in microcellular environments," in *IEEE Vehicular Technology Conference*, May 1992, pp. 932–935 vol.2.

[21] D. Didascalou, T. M. Schafer, F. Weinmann, and W. Wiesbeck, "Ray-density normalization for ray-optical wave propagation modeling in arbitrarily shaped tunnels," *IEEE Transactions on Antennas and Propagation*, vol. 48, no. 9, pp. 1316–1325, Sep 2000.

[22] Z. Ji, B.-H. Li, H.-X. Wang, H.-Y. Chen, and T. K. Sarkar, "Efficient ray-tracing methods for propagation prediction for indoor wireless communications," *IEEE Antennas and Propagation Magazine*, vol. 43, no. 2, pp. 41–49, April 2001.

[23] Z. Zhang, R. K. Sorensen, Z. Yun, M. F. Iskander, and J. F. Harvey, "A ray-tracing approach for indoor/outdoor propagation through window structures," *IEEE Transactions on Antennas and Propagation*, vol. 50, no. 5, pp. 742–748, May 2002.

[24] G. L. Turin, F. D. Clapp, T. L. Johnston, S. B. Fine, and D. Lavry, "A statistical model of urban multipath propagation," *IEEE Transactions on Vehicular Technology*, vol. 21, no. 1, pp. 1–9, Feb 1972.

[25] M. Iskander and Z. Yun, "Propagation prediction models for wireless communication systems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 3, pp. 662–673, Mar. 2002.

[26] Y. Okumura, E. Ohmori, T. Kawano, and K. Fukuda, "Field strength variability in VHF and UHF land mobile service," *Review of the Electrical Communication Laboratory*, vol. 16, no. 9-10, pp. 825–873, Sept. 1968.

[27] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Transactions on Vehicular Technology*, vol. 29, no. 3, pp. 317–325, Aug 1980.

[28] "Propagation prediction models," *COST 231 Final Rep., ch.4*, pp. 17–21.

[29] V. Sampath, C. Despins, B. Sultana, W. Lippler, and G. Y. Delisle, "Comparison of statistical and deterministic indoor propagation prediction techniques with field measurements," in *IEEE Vehicular Technology Conference*, vol. 2, May 1997, pp. 1138–1142 vol.2.

[30] K. Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," *IEEE Transactions on Antennas and Propagation*, vol. 14, no. 3, pp. 302–307, May 1966.

[31] L. Nagy, "Comparison and application of FDTD and ray optical method for indoor wave propagation modeling," in *Proceedings of the Fourth European Conference on Antennas and Propagation*, Apr. 2010, pp. 1–3.

[32] J. Walfisch and H. L. Bertoni, "A theoretical model of UHF propagation in urban environments," *IEEE Transactions on Antennas and Propagation*, vol. 36, no. 12, pp. 1788–1796, 1988.

[33] M. F. Catedra and J. Perez, *Cell Planning for Wireless Communications*, 1st ed. Artech House, Inc., 1999.

[34] C. A. Balanis, *Antenna Theory: Analysis and Design*, 3rd ed. Wiley-Interscience, 2005.

[35] M. Catedra, J. Perez, F. Saez de Adana, and O. Gutierrez, "Efficient ray-tracing techniques for three-dimensional analyses of propagation in mobile communications:

application to picocell and microcell scenarios," *IEEE Antennas and Propagation Magazine*, vol. 40, no. 2, pp. 15–28, 1998.

[36] F. Agelet, A. Formella, J. Hernando Rabanos, F. de Vicente, and F. Fontan, "Efficient ray-tracing acceleration techniques for radio propagation modeling," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 6, pp. 2089–2104, Nov. 2000.

[37] H. Azodi, U. Siart, and T. Eibert, "A fast 3-D deterministic ray tracing coverage simulator including creeping rays based on geometry voxelization technique," *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 1, pp. 210–220, Jan. 2015.

[38] X. Meng, L.-X. Guo, Y.-W. Wei, and J.-J. Sun, "An accelerated ray tracing method based on the TSM for the RCS prediction of 3-D large-scale dielectric sea surface," *IEEE Antennas and Wireless Propagation Letters*, vol. 14, pp. 233–236, 2015.

[39] S. A. H. Tabatabaei, M. Fleury, N. N. Qadri, and M. Ghanbari, "Improving propagation modeling in urban environments for vehicular Ad Hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 705–716, Sept. 2011.

[40] W. Sun, E. G. Ström, F. Brännström, K. C. Sou, and Y. Sui, "Radio resource management for D2D-based V2V communication," *IEEE Transations on Vehicular Technology*, vol. 65, no. 8, pp. 6636–6650, Aug. 2016.

[41] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of IoT: Applications, Challenges, and Opportunities with China Perspective," *IEEE Internet Things Journal*, vol. 1, no. 4, pp. 349–359, Aug. 2014.

[42] I. Steinberg, E. Kaplan, M. Ben-David, and I. Gannot, "The role of skew rays in biomedical sensing," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 16, no. 4, pp. 961–966, July 2010.

[43] P. de Heras Ciechomski, M. Klann, R. Mange, and H. Koeppl, "From biochemical reaction networks to 3D dynamics in the cell: The zigcell3D modeling, simulation and visualisation framework," in *IEEE Symposium on Biological Data Visualization*, Oct. 2013, pp. 41–48.

[44] D. Lopez-Perez, A. Juttner, and J. Zhang, "Dynamic frequency planning versus frequency reuse schemes in OFDMA networks," in *IEEE Vehicular Technology Conference*, Apr. 2009, pp. 1–5.

[45] I. Tal, B. Ciubotaru, and G. M. Muntean, "Vehicular-communications-based speed advisory system for electric bicycles," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 4129–4143, June 2016.

[46] T. Alho, P. Hamalainen, M. Hannikainen, and T. D. Hamalainen, “Design of a compact modular exponentiation accelerator for modern FPGA devices,” in *World Automation Congress*, July 2006, pp. 1–7.

[47] A. Kanatas, I. Kountouris, G. Kostaras, and P. Constantinou, “A UTD propagation model in urban microcellular environments,” *IEEE Transactions Vehicular Technology*, vol. 46, no. 1, pp. 185–193, Feb. 1997.

[48] J. B. Keller, “Geometrical theory of diffraction,” *Journal of the Optical Society of America*, vol. 52, no. 2, pp. 116–130, Feb. 1962.

[49] Y. Tao, H. Lin, and H. Bao, “GPU-based shooting and bouncing ray method for fast RCS prediction,” *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 2, pp. 494–502, Feb 2010.

[50] G. Durgin, N. Patwari, and T. Rappaport, “An advanced 3D ray launching method for wireless propagation prediction,” in *IEEE Vehicular Technology Conference*, vol. 2, 1997, pp. 785–789 vol.2.

[51] S. Seidel and T. Rappaport, “Site-specific propagation prediction for wireless in-building personal communication system design,” *IEEE Transactions on Vehicular Technology*, vol. 43, no. 4, pp. 879–891, 1994.

[52] H. Fuchs, Z. M. Kedem, and B. F. Naylor, “On visible surface generation by a priori tree structures,” *Computer Graphics*, vol. 14, no. 3, pp. 124–133, july 1980.

[53] S. M. Rubin and T. Whitted, “A 3-dimensional representation for fast rendering of complex scenes,” *Computer Graphics*, vol. 14, no. 3, pp. 110–116, july 1980.

[54] M. E. Newell, R. G. Newell, and T. L. Sancha, “A solution to the hidden surface problem,” in *Proceedings of the ACM Annual Conference*, 1972, pp. 443–450.

[55] A. Goldsmith, *Wireless Communications*. New York, NY, USA: Cambridge University Press, 2005.

[56] N. Amitay, “Modeling and computer simulation of wave propagation in lineal line-of-sight microcells,” *IEEE Transactions on Vehicular Technology*, vol. 41, no. 4, pp. 337–342, Nov. 1992.

[57] K. Hammad, M. Mirahmadi, S. Primak, and A. Shami, “On a throughput-efficient look-forward channel-aware scheduling,” in *IEEE International Conference on Communications*, June 2015, pp. 6234–6239.

[58] P. Viswanath, D. Tse, and R. Laroia, “Opportunistic beamforming using dumb antennas,” *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1277–1294, June 2002.

[59] D. Dechene and A. Shami, "Energy-aware resource allocation strategies for LTE uplink with synchronous HARQ constraints," *IEEE Transactions on Mobile Computing*, vol. 13, no. 2, pp. 422–433, Feb. 2014.

[60] M. Kalil, A. Shami, A. Al-Dweik, and S. Muhaidat, "Low-complexity power-efficient schedulers for LTE uplink with delay-sensitive traffic," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4551–4564, Oct. 2015.

[61] M. Mirahmadi and A. Shami, "Traffic-prediction-assisted dynamic bandwidth assignment for hybrid optical wireless networks," *Computer Networks*, vol. 56, no. 1, pp. 244–259, Jan. 2012.

[62] N. C. Ericsson, A. Ahlen, S. Falahati, and A. Svensson, "Hybrid type-ii ARQ/AMS supported by channel predictive scheduling in a multi-user scenario," in *IEEE Vehicular Technology Conference*, vol. 4, 2000, pp. 1804–1811 vol.4.

[63] S. Primak, V. Kontorovich, and V. Lyandres, *Stochastic methods and their applications to Communications: SDE Approach*. Chichester: John Wiley & Sons, 2004.

[64] COST 207, "Digital land mobile radio communications," *Office for Official Publications of the European Communities, Final report*, 1989.

[65] G. Miao, N. Himayat, G. Li, and S. Talwar, "Low-complexity energy-efficient scheduling for uplink OFDMA," *IEEE Transactions on Communications*, vol. 60, no. 1, pp. 112–120, January 2012.

[66] G. Song and Y. Li, "Cross-layer optimization for OFDM wireless networks-part I: theoretical framework," *IEEE Transations on Wireless Communications*, vol. 4, no. 2, pp. 614–624, Mar. 2005.

[67] M. Zorzi and R. Rao, "Capture and retransmission control in mobile radio," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 8, pp. 1289–1298, Oct. 1994.

[68] D. Dechene and A. Shami, "Energy efficient resource allocation in SC-FDMA uplink with synchronous HARQ constraints," in *IEEE International Conference on Communications*, June 2011, pp. 1–5.

[69] K. Hammad, S. Primak, M. Kalil, and A. Shami, "QoS-aware energy-efficient downlink predictive scheduler for OFDMA-based cellular devices," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2016.

[70] Statista, Available at: http://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/.

[71] PewResearchCenter, Available at: http://www.pewinternet.org/fact-sheets/social-networking-fact-sheet/.

[72] D. Feng, C. Jiang, G. Lim, J. Cimini, L.J., G. Feng, and G. Li, "A survey of energy-efficient wireless communications," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 1, pp. 167–178, Feb. 2013.

[73] L. Venturino, A. Zappone, C. Risi, and S. Buzzi, "Energy-efficient scheduling and power allocation in downlink OFDMA networks with BS coordination," *IEEE Transactins on Wireless Communications*, vol. 14, no. 1, pp. 1–14, Jan. 2015.

[74] R. Gupta and E. Strinati, "Green scheduling to minimize base station transmit power and UE circuit power consumption," in *IEEE International Symposium Personal Indoor and Mobile Radio Communications*, Sept. 2011, pp. 2424–2429.

[75] C. Xiong, G. Li, Y. Liu, Y. Chen, and S. Xu, "Energy-efficient design for downlink OFDMA with delay-sensitive traffic," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 3085–3095, June 2013.

[76] J. Tang and X. Zhang, "Cross-layer-model based adaptive resource allocation for statistical QoS guarantees in mobile wireless networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, pp. 2318–2328, June 2008.

[77] S. Wang, W. Shi, and C. Wang, "Energy-efficient resource management in OFDM-based cognitive radio networks under channel uncertainty," *IEEE Transactions on Communications*, vol. 63, no. 9, pp. 3092–3102, Sept. 2015.

[78] F. Chu, K. Chen, and G. Fettweis, "Green resource allocation to minimize receiving energy in OFDMA cellular systems," *IEEE Communications Letters*, vol. 16, no. 3, pp. 372–374, Mar. 2012.

[79] C. Bontu and E. Illidge, "DRX mechanism for power saving in LTE," *IEEE Communications Magazine*, vol. 47, no. 6, pp. 48–55, June 2009.

[80] C.-L. I, J. Huang, R. Duan, C. Cui, J. Jiang, and L. Li, "Recent progress on C-RAN centralization and cloudification," *IEEE Access*, vol. 2, pp. 1030–1039, 2014.

[81] B. Al-Manthari, H. Hassanein, N. Ali, and N. Nasser, "Fair class-based downlink scheduling with revenue considerations in next generation broadband wireless access systems," *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, pp. 721–734, June 2009.

[82] A. Jensen, M. Lauridsen, P. Mogensen, T. Sørensen, and P. Jensen, "LTE UE power consumption model: For system level energy and performance optimization," in *IEEE Vehicular Technology Conference*, Sept. 2012, pp. 1–5.

[83] H.-Y. Kim, Y.-J. Kim, J.-H. Oh, and L.-S. Kim, "A reconfigurable SIMT processor for mobile ray tracing with contention reduction in shared memory," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 4, pp. 938–950, Apr. 2013.

[84] M. Morelli and U. Mengali, "A comparison of pilot-aided channel estimation methods for OFDM systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 3065–3073, Dec. 2001.

[85] J. Del Peral-Rosado, J. Lopez-Salcedo, G. Seco-Granados, F. Zanier, and M. Crisci, "Achievable localization accuracy of the positioning reference signal of 3GPP LTE," in *International Conference on Localization and GNSS*, June 2012, pp. 1–6.

[86] W. Wang, V. Lau, and M. Peng, "Delay-optimal fronthaul allocation via perturbation analysis in cloud radio access networks," in *IEEE International Conference on Communications*, June 2015, pp. 3999–4004.

[87] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. MA: Athena Scientific, 1996.

[88] S.-B. Lee, I. Pefkianakis, A. Meyerson, S. Xu, and S. Lu, "Proportional fair frequency-domain packet scheduling for 3GPP LTE uplink," in *IEEE INFOCOM*, April 2009, pp. 2611–2615.

[89] K. Hammad, A. Moubayed, A. Shami, and S. Primak, "Analytical approximation of packet delay jitter in simple queues," *IEEE Wireless Communications Letters*, vol. PP, no. 99, pp. 1–1, 2016.

[90] K. Hammad, S. Primak, A. Moubayed, and A. Shami, "Investigating the energy-efficiency/delay jitter trade-off for VoLTE in LTE downlink," *Submitted to the IEEE Transactions on Vehicular Technology*.

[91] F. Houeto and S. Pierre, "Characterization of jitter and admission control in multi-service networks," *IEEE Communications Letters*, vol. 8, no. 2, pp. 125–127, Feb. 2004.

[92] B. Oklander and M. Sidi, "Jitter buffer analysis," in *Proceedings of the International Conference on Computer Communications and Networking*, Aug. 2008, pp. 1–6.

[93] W. Matragi, K. Sohraby, and C. Bisdikian, "Jitter calculus in ATM networks: multiple nodes," *IEEE/ACM Transations on Networking*, vol. 5, no. 1, pp. 122–133, 1997.

[94] D. Wen, C. Xue-fen, L. Zi-chuan, and Z. Yu-hong, "Queuing theory based analysis for packet jitter of mixed services," *The Journal of China Universities of Posts and Telecommunications*, vol. 21, no. 3, pp. 71–76, 2014.

[95] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. Boston, MA: McGraw-Hill, 1991.

[96] M. Abramowitz and I. Stegun, Eds., *Handbook of Mathematical Functions*. New York: Dover, 1965.

[97] M. S. Alouini and A. J. Goldsmith, "Capacity of rayleigh fading channels under different adaptive transmission and diversity-combining techniques," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 4, pp. 1165–1181, July 1999.

[98] S. Primak, V. Lyandres, and V. Kontorovich, "Markov models of non-Gaussian exponentially correlated processes and their applications," *Physical Review E*, vol. 63, no. 6 I, Article ID 061103, May 2001.

[99] A. Sniady, M. Sonderskov, and J. Soler, "VoLTE performance in railway scenarios: Investigating VoLTE as a viable replacement for GSM-R," *IEEE Vehicular Technology Magazine*, vol. 10, no. 3, pp. 60–70, Sept. 2015.

[100] R2-074933, "Introduction of CS voice over HSPA," 3GPP RAN-WG2.

[101] C. Courcoubetis and R. Weber, "Effective bandwidths for stationary sources," *Probability in the Engineering and Informational Sciences*, vol. 9, no. 02, pp. 285–296, 1995.

[102] A. L. Stolyar and K. Ramanan, "Largest weighted delay first scheduling: Large deviations and optimality," *The Annals of Applied Probability*, no. 1, pp. 1–48, 02.

[103] M. J. Neely and S. Supittayapornpong, "Dynamic markov decision policies for delay constrained wireless scheduling," *IEEE Transactions on Automatic Control*, vol. 58, no. 8, pp. 1948–1961, Aug. 2013.

[104] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Communications Magazine*, vol. 39, no. 2, pp. 150–154, Feb. 2001.

[105] B. Sadiq, R. Madan, and A. Sampath, "Downlink scheduling for multiclass traffic in LTE," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, no. 14, pp. 1–18, 2009.

[106] K. Yong-Seok, "Capacity of VoIP over HSDPA with frame bundling," *IEICE transactions on communications*, vol. 89, no. 12, pp. 3450–3453, 2006.

[107] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," *DEC Research Report TR-301*, Sept. 1984.

[108] K. Hammad, A. Shami, S. Primak, and A. Moubayed, "QoS-aware energy and jitter-efficient downlink predictive scheduler for heterogeneous traffic LTE networks," *Submitted to the IEEE Transactions on Mobile Computing*.

[109] M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of things in the 5G era: Enablers, architecture, and business models," *IEEE*

Journal on Selected Areas in Communications, vol. 34, no. 3, pp. 510–527, Mar. 2016.

[110] P. Demestichas, A. Georgakopoulos, K. Tsagkaris, and S. Kotrotsos, “Intelligent 5G networks: Managing 5G wireless/mobile broadband,” *IEEE Vehicular Technology Magazine*, vol. 10, no. 3, pp. 41–50, Sept. 2015.

[111] T. Wang, G. Li, J. Ding, Q. Miao, J. Li, and Y. Wang, “5G spectrum: is china ready?” *IEEE Communications Magazine*, vol. 53, no. 7, pp. 58–65, July 2015.

[112] J. A. Stankovic, “Research directions for the internet of things,” *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, Feb. 2014.

[113] M. Altamimi, A. Abdrabou, K. Naik, and A. Nayak, “Energy cost models of smartphones for task offloading to the cloud,” *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 3, pp. 384–398, Sept. 2015.

[114] C. Y. Wong, R. S. Cheng, K. B. Lataief, and R. D. Murch, “Multiuser OFDM with adaptive subcarrier, bit, and power allocation,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 10, pp. 1747–1758, Oct 1999.

[115] M. Bohge, J. Gross, A. Wolisz, and M. Meyer, “Dynamic resource allocation in OFDM systems: an overview of cross-layer optimization principles and techniques,” *IEEE Network*, vol. 21, no. 1, pp. 53–59, Jan 2007.

[116] R. A. Berry and R. G. Gallager, “Communication over fading channels with delay constraints,” *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1135–1149, May 2002.

[117] J. Liu, W. Chen, and K. B. Letaief, “Joint channel and queue aware scheduling for wireless links with multiple fading states,” in *IEEE International Conference on Communications in China*, Nov 2015, pp. 1–6.

[118] X. Zhong and C. Z. Xu, “Energy-efficient wireless packet scheduling with quality of service control,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 10, pp. 1158–1170, Oct 2007.

[119] M. I. Poulakis, A. D. Panagopoulos, and P. Constantinou, “Channel-aware opportunistic transmission scheduling for energy-efficient wireless links,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 192–204, Jan 2013.

[120] L. Lei, D. Yuan, C. K. Ho, and S. Sun, “Resource scheduling to jointly minimize receiving and transmitting energy in OFDMA systems,” in *International Symposium on Wireless Communications Systems*, Aug 2014, pp. 187–191.

[121] X. Wang, G. B. Giannakis, and A. G. Marques, “A unified approach to qos-guaranteed scheduling for channel-adaptive wireless networks,” *Proceedings of the IEEE*, vol. 95, no. 12, pp. 2410–2431, Dec 2007.

[122] H. Varian, *Intermediate Micoreconomics: A Modern Approach*, 6th ed. W.W. Norton and Company, 2003.

[123] M. Gidlund and J. C. Laneri, “Scheduling algorithms for 3GPP long-term evolution systems: From a quality of service perspective,” in *IEEE International Symposium on Spread Spectrum Techniques and Applications*, Aug 2008, pp. 118–123.

[124] Y. Zaki, T. Weerawardane, C. Gorg, and A. Timm-Giel, “Multi-QoS-aware fair scheduling for LTE,” in *IEEE Vehicular Technology Conference*, May 2011, pp. 1–5.

[125] D. C. Verma, H. Zhang, and D. Ferrari, “Delay jitter control for real-time communication in a packet switching network,” *IEEE Conference on Communications Software, Proceedings of TRICOMM ’91*, pp. 35–43, Apr. 1991.

[126] C. Rosado-Sosa and I. Rubin, “Jitter compensation scheduling schemes for the support of real-time communications,” *IEEE International Conference on Communications*, vol. 2, pp. 885–890, June 1998.

[127] F. P. Zhang, O. W. W. Yang, and B. Cheng, “Performance evaluation of jitter management algorithms,” *Canadian Conference on Electrical and Computer Engineering*, vol. 2, pp. 1011–1016, May 2001.

[128] B. Rong, Y. Qian, M. H. Guiagoussou, and M. Kadoch, “Improving delay and jitter performance in wireless mesh networks for mobile IPTV services,” *IEEE Transactions on Broadcasting*, vol. 55, no. 3, pp. 642–651, Sept. 2009.

[129] P. Goyal and H. M. Vin, “Generalized guaranteed rate scheduling algorithms: A framework,” *IEEE/ACM Transactions on Networking*, vol. 5, no. 4, pp. 561–571, Aug. 1997.

[130] S. Mumtaz, K. M. S. Huq, A. Radwan, J. Rodriguez, and R. L. Aguiar, “Energy efficient interference-aware resource allocation in LTE-D2D communication,” in *IEEE International Conference on Communications*, June 2014, pp. 282–287.

[131] P. Janis, V. Koivunen, C. Ribeiro, J. Korhonen, K. Doppler, and K. Hugl, “Interference-aware resource allocation for device-to-device radio underlaying cellular networks,” in *IEEE Vehicular Technology Conference*, Apr. 2009, pp. 1–5.

[132] Z. Zhou, M. Dong, K. Ota, R. Shi, Z. Liu, and T. Sato, “Game-theoretic approach to energy-efficient resource allocation in device-to-device underlay communications,” *IET Communications*, vol. 9, no. 3, pp. 375–385, Feb. 2015.

[133] M. I. Kamel, L. B. Le, and A. Girard, “LTE wireless network virtualization: Dynamic slicing via flexible scheduling,” in *IEEE Vehicular Technology Conference*, Sept. 2014, pp. 1–5.

[134] A. Moubayed, A. Shami, and H. Lutfiyya, “Wireless resource virtualization with device-to-device communication underlaying LTE network,” *IEEE Transactions on Broadcasting*, vol. 61, no. 4, pp. 734–740, Dec. 2015.

[135] M. Kalil, A. Shami, and Y. Ye, “Wireless resources virtualization in LTE systems,” in *IEEE Conference on Computer Communications Workshops*, Apr. 2014, pp. 363–368.

[136] A. D. Wyner, “The wire-tap channel,” *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, Oct. 1975.

[137] J. Barros and M. R. D. Rodrigues, “Secrecy capacity of wireless channels,” in *IEEE International Symposium on Information Theory*, July 2006, pp. 356–360.

[138] J. P. Vilela, M. Bloch, J. Barros, and S. W. McLaughlin, “Wireless secrecy regions with friendly jamming,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 256–266, June 2011.

Curriculum Vitae

Name:	Karim Ahmed Hammad
Post-secondary Education and Degrees:	2012-2016 Ph.D Electrical and Computer Engineering Western University London, Ontario, Canada
	2000-2005 B.Sc., 2005-2009 M.Sc. Electronics and Communications Engineering Arab Academy for Science, Technology and Maritime Transport Cairo, Egypt
Related Work Experience	2013-2016 Teaching Assistant Western University London, Ontario, Canada
	ES 1036 - Programming Fundamentals for Engineers ECE 4437 - Communication Theory ECE 2274 - Electric Circuits ECE 4436 - Networking: Principles, Protocols, and Architectures ECE 2208 - Electrical Measurements and Instrumentation
Honours and Awards	Western Graduate Research Scholarship (WGRS), and Western Graduate Research Assistance (WGRA) The Department of Electrical and Computer Engineering Western University, London, Ontario, Canada, 2012-2016
Publications	
[J1]	K. Hammad, S. Primak, M. Kalil, and A. Shami, "QoS-Aware Energy-Efficient Downlink Predictive Scheduler for OFDMA-based Cellular Devices," IEEE Transactions on Vehicular Technology, 2016.

[J2] K. Hammad, A. Moubayed, A. Shami, and S. Primak, "Analytical Approximation of Packet Delay Jitter in Simple Queues," IEEE Wireless Communications Letters, 2016

[J3] K. Hammad, S. Primak, A. Moubayed, and A. Shami, "Investigating the Energy-Efficiency/Delay Jitter Trade-off for VoLTE in LTE Downlink," *Submitted to the IEEE Transactions on Vehicular Technology*

[J4] K. Hammad, A. Shami, S. Primak, and A. Moubayed, "QoS-Aware Energy and Jitter-Efficient Downlink Predictive Scheduler for Heterogeneous Traffic LTE Networks," *Submitted to the IEEE Transactions on Mobile Computing*

[C1] K. Hammad, M. Mirahmadi, S. Primak, and A. Shami, "On a Throughput-Efficient Look-Forward Channel-Aware Scheduling," IEEE International Conference on Communications (ICC), June 2015