Emulating a Software Defined LTE Radio Access Network Towards 5G

Inês da Silva Santos

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. António José Castelo Branco Rodrigues
Prof. Maria Paula Dos Santos Queluz Rodrigues
Prof. Pedro Manuel de Almeida Carvalho Vieira

Examination Committee

Chairperson: Prof. José Eduardo Charters Ribeiro da Cunha Sanguino
Member of the Committee: Prof. Pedro Joaquim Amaro Sebastião
Supervisor: Prof. António José Castelo Branco Rodrigues

June 2018

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

Firstly, I would like to thank my supervisors Professor António Rodrigues, Professora Maria Paula Queluz and Professor Pedro Vieira, which contributed greatly to the development of this work, being always available to help.

I want to thank to Celfinet for the opportunity of doing this project in a professional environment and to all my team members not only for the help but for fun activities that we shared. A special thanks to Eng. Rúben Boralho for the support and patience, and for taking the time to help me.

I also want to thank Instituto de Telecomunicações for providing me with the required means for the completion of this dissertation.

To my family, specially my parents, for the love and caring, for having always supported and valued my work, and for all the incredible opportunities they provided me.

Finally, I am really grateful to have met amazing people during my journey in IST, who taught me that university is much more than work, and with whom I have shared the best moments of those years. Filipa Oliveira, Rute Santos and Vera Korchevnyuk, thank you!

Abstract

This thesis evaluates the performance of a Software Defined Network (SDN) and the respective OpenFlow (OF) protocol, when supporting a Long Term Evolution (LTE) Radio Access Network (RAN). The main objective is to understand the impact of the new network technologies and the new network architecture, using a SDN based LTE network (when compared to conventional LTE networks), in a solution towards Fifth Generation (5G) implementation. Several simulations were performed using the SDN emulator Mininet-WiFi platform, which was adapted to produce use cases regarding a SDN based LTE topology. This platform uses WiFi as radio technology. However, and since this is an open source platform, changes were introduced in order to include a LTE RAN parametrization based on LTE technology standards. The latency and throughput are the used Quality of Service (QoS) performance metrics.

The throughput results led to a 54 % and 158 % spectral efficiency increase regarding the ITU requirement for urban and rural environments respectively, for a LTE Radio Access system reference from 3GPP. Latency results were also obtained for the same reference scenario achieving, in the urban environment, average values of 21 - 22 ms, close to the 3GPP requirement for LTE of 20 ms; on the other hand, for the rural environment the average latency values were around 122 ms. The latency was also analyzed according to the SNR and the packet size.

The latency was then measured according to the *drive-tests* scenario, in order to compare the emulator results with real network measurements. The emulator presented better latency results than the *drive-tests* despite the different latency curve behaviour. At last, the latency was measured only in the backhaul/core network where the SDN influence is notorious, and, for the air interface, reference values from 3GPP were used. The obtained results were compared with 3GPP reference values, an ITU case study and *drive-tests*. The average latencies obtained presented 62.15 % and 63.77 % latency decrease regarding the 3GPP and ITU results respectively. According to the drive-tests, the emulator reached 40.18 % lower latency.

Keywords: SDN, C-RAN, 5G, OpenFlow, throughput, latency, RTT, User Plane.

Resumo

Esta tese avalia o desempenho de uma *Software Defined Network* (SDN) e respectivo protocolo OpenFlow (OF), ao suportar uma rede de acesso rádio LTE. O principal objetivo é entender o impacto das novas tecnologias e arquiteturas de rede, usando uma rede LTE baseada em SDN (quando comparada com redes LTE convencionais), numa solução para a implementação de redes de quinta geração (5G). Várias simulações foram realizadas usando o emulador SDN Mininet-WiFi, o qual foi adaptado para produzir casos de uso relativos a uma topologia LTE baseada em SDN. Esta plaforma usa WiFi como tecnologia rádio; no entanto, e uma vez que se trata de uma plantaforma *open source*, alterações foram introduzidas para incluir uma parametrização LTE na rede de acesso rádio baseada nos padrões da tecnologia LTE. A latência e o *throughput* são as métricas de desempenho de Qualidade de Serviço (QoS) usadas.

Os resultados de *throughput* conduziram a um aumento de eficiência espectral de 54 % e 158 % para ambiente urbano e rural respectivamente, relativamente ao requisito da ITU, para um sistema de rede de acesso LTE referência do 3GPP. Resultados de latência foram também obtidos para o mesmo cenário de referência alcançando, em ambiente urbano, valores médios de 21 - 22 ms, próximos do requisito do 3GPP para LTE de 20 ms; por outro lado, para ambiente rural, os valores médios de latência foram cerca de 122 ms. A latência foi também analisada de acordo com o SNR e com o tamanho dos pacotes.

A latência foi depois medida de acordo com o cenário de *drive-tests* de forma a comparar os resultados do emulador com medições de rede reais. O emulador apresentou melhor resultados de latência do que os *drive-tests* apesar do diferente comportamento da curva de latência. Por fim, a latência foi medida apenas na rede *backhaul*/*core* onde a influência do SDN é notória. Para a interface sem fios foram usados valores de referência do 3GPP. Os valores obtidos foram comparados com os valores de referência do 3GPP, com um caso de estudo da ITU e com *drive-tests*. As latências médias obtidas apresentaram um decréscimo na latência de 62,15 % e 63,77 % relativamente aos resultados do 3GPP e da ITU respetivamente. Relativamente aos *drive-tests*, o emulador alcançou 40,18 % menos latência.

Palavras-chave: SDN, C-RAN, 5G, OpenFlow, *throughput*, latência, Plano do utilizador.

Contents

Acknowledgments . . . v

Abstract . . . vii

Resumo . . . ix

List of Tables . . . xiii

List of Figures . . . xv

List of Software . . . xvii

List of Symbols . . . xix

Acronyms . . . xxi

1 Introduction 1

1.1 Motivation . . . 1

1.2 Objectives . . . 2

1.3 Thesis Outline . . . 2

1.4 Publications . . . 3

2 State of the art 5

2.1 Long-Term Evolution . . . 5

2.1.1 Architecture . . . 5

2.1.2 Radio Interface . . . 7

2.1.3 Protocol architecture for E-UTRA . . . 10

2.1.4 Similarities with WiFi . . . 10

2.2 Software Defined Networks . . . 13

2.3 Network Function Virtualization . . . 19

2.4 C-RAN . . . 23

3 LTE over SDN Implementation 25

3.1 Emulator Description . . . 25

3.2 LTE modules implementation . . . 30

3.2.1 LTE Radio Frequency (RF) scenario . . . 30

3.2.2 Modulation and coding scheme . . . 32

3.2.3 RRHs placement - grid implementation . . . 34

3.3 Network Tests and Assessment . . . 36

3.3.1 Assumptions 36
3.3.2 Network Tests 37
3.3.3 Hypotheses Assessment 41

4 Result Analysis **43**
4.1 Throughput 43
4.2 Latency 49
4.3 Use case - *Drive-tests* 55
4.3.1 Propagation Model calibration 55
4.3.2 Latency analysis 60
4.3.3 End-to-End Latency 63
4.4 Use Case - Energy saving 67

5 Conclusions **71**
5.1 Summary 71
5.2 Future Work 73

References **75**

A LTE Modulation and Coding Scheme **81**

B 3GPP U-Plane latency **83**
B.1 ITU U-Plane latency analysis 83

List of Tables

2.1 Number of resource blocks for each bandwidth. 8

2.2 Downlink peak bit rates in LTE (adapted from [10]). 9

2.3 Uplink peak bit rates in LTE (adapted from [10]). 10

3.1 RAN parameters from [39]. 31

3.2 RSRP intervals defined in the emulator. 35

3.3 U-Plane radio network latency analysis (estimated average) [44]. 38

3.4 U-plane latency analysis (estimated average) [44], [11]. 39

4.1 Throughput fitting results for an urban environment. 44

4.2 Bandwidth efficiency for 10 MHz LTE system downlink [45]. 44

4.3 Antenna scheme in [45]. 45

4.4 Throughput fitting results for a rural environment. 46

4.5 Regression fitting for higher values of SNR in Rural areas. 47

4.6 Average spectral efficiency comparison, between the emulator and the ITU requirement. . 47

4.7 Latency fitting coefficients for Urban areas 50

4.8 Latency fitting coefficients for rural areas. 51

4.9 Latency comparison between emulator's tests. 53

4.10 RF Conditions [48]. 54

4.11 Scenario 1 - *Drive-tests*' parameters. 56

4.12 Scenario 1 - Propagation model parameters. 57

4.13 Scenario 2 - *Drive-tests*' parameters. 59

4.14 Scenario 2 - Propagation model parameters. 60

4.15 RTT comparison between the emulator values and the values from 3GPP [11]. 65

4.16 RTT comparison between the emulator values and the values from ITU [51]. 66

5.1 RTT reduction regarding the reference [11] from 3GPP. 72

A.1 4-bit CQI Table [40] 81

B.1 U-Plane latency analysis (estimated average in downlink) according TR 125 912 and S1-U Transfer Delay [51] 83

B.2 U-Plane latency analysis (estimated average in uplink) according TR 125 912 and S1-U Transfer Delay [51] 84

List of Figures

1.1 Data traffic per smartphone (GB per month) from Ericsson [1]. 1

2.1 LTE architecture. 6

2.2 OFDMA vs SC-FDMA (adapted from [7]). 7

2.3 Difference between OFDM and OFDMA (adapted from [8]). 8

2.4 Resource block representation (extracted from [7]). 9

2.5 E-UTRAN architecture overview (adapted from [11]). 11

2.6 U-plane protocol stack (adapted from [11]). 11

2.7 LTE WLAN connection (adapted from [14]). 12

2.8 Connection protocols. 12

2.9 SDN overview (adapted from [17]). 14

2.10 OpenFlow methodology [19]. 15

2.11 Flow entry details (adapted from [20] and [19]). 15

2.12 Flow Table example [21]. 16

2.13 Controller integration with the S/P-GW (adapted from [22]). 17

2.14 4G LTE User Plane Protocols Stack with OpenFlow (adapted from [23]). 17

2.15 5G User Plane Protocols Stack with OpenFlow (adapted from [23]). 18

2.16 Gateway architectures [24]. 19

2.17 NFV framework (adapted from [25]). 20

2.18 NFV over a SDN network. 21

2.19 General architecture of a Software Defined Network (SDN) and virtualization-based mobile network (adapted from [27]). 22

2.20 C-RAN LTE mobile network (adapted from [31]). 24

2.21 Power consumption of China Mobile (adapted from [32]). 24

3.1 Mininet-Wifi architecture (adapted from [37]). 26

3.2 Topology implemented in Mininet-WiFi. 27

3.3 Mininet-Wifi architecture and components. 27

3.4 POX terminal window. 28

3.5 Network elements' flow tables in Mininet-WiFi. 29

3.6 Cells layout. 30

3.7 Received power according to distance. 32
3.8 MCS Theoretical curves. 33
3.9 RRHs' placement. 36
3.10 *Iperf* test. 37
3.11 LTE Control plane and User plane latency. 38
3.12 U-Plane end-to-end latency components (adapted from [11]). 39
3.13 End-to-End (E2E) *Ping* test. 40
3.14 Wired *Ping* test. 40

4.1 Spectral efficiency obtained for the urban area in the emulator. 44
4.2 Throughput fitting curves comparison between emulator's fit and Shannon equations from [45]. 45
4.3 Difference between emulator's fit and Shannon equations from [45] 46
4.4 Spectral efficiency obtained for the rural area in the emulator. 47
4.5 Spectral efficiency curves for urban and rural environments obtained in the emulator. . . . 48
4.6 Bar plot representation of the average spectral efficiencies. 48
4.7 Latency fitting curves according to packet size for the urban area. 49
4.8 Latency fitting curves comparison for the urban area. 50
4.9 Latency fitting curves according to packet size for the rural area. 51
4.10 Latency fitting curves comparison for the rural area. 52
4.11 Latency fitting curves for both Urban and Rural areas. 52
4.12 Latency comparison according to [47]. 54
4.13 Received power curves - Scenario 1. 57
4.14 Scenario 1 - Footprint representation. 58
4.15 Algorithm explanation. 58
4.16 Received power curve - Scenario 2. 59
4.17 Scenario 1 - Latency fitting curve and *drive-tests*' RTT. 61
4.18 Scenario 2 - Latency fitting curve and *drive-tests*' RTT. 61
4.19 RTT *drive-tests*. 62
4.20 Latency curve obtained from scenarios 1 and 2. 62
4.21 Latency curve obtained and RTT points from *drive-tests*. 63
4.22 Wired network test in Mininet-WiFi. 64
4.23 Latency according to the number of switches in the connection. 64
4.24 Emulator latency compared with the 3GPP S1-U latency. 65
4.25 RTT comparison between *drive-tests* and Emulator/3GPP. 67
4.26 Energy saving diagram. 68
4.27 Energy saving execution. 68

List of Software

- Microsoft Excel 2016
- Matlab 2016
- Mininet-Wifi - Emulator for Software-Defined Wireless Networks
- Google Maps
- Microsoft SQL Server 2014

List of Symbols

$\eta_{\mathrm{BW,r}}$	Bandwidth efficiency, for r antennas.
$\eta_{\mathrm{SNR,r}}$	SNR efficiency, for r antennas.
$\frac{\mathrm{k}}{\mathrm{n}}$	Code rate.
$\frac{\mathrm{R_b}}{\mathrm{B}}$	Spectral efficiency.
B	Bandwidth.
C	Shannon capacity.
f	Frequency.
$\mathrm{G_{MCS}}$	Coding gain.
$\mathrm{G_r}$	Gain of the receiving antenna.
$\mathrm{G_t}$	Gain of the transmitting antenna.
$\mathrm{h_{tx}}$	Transmission antenna height.
k	Boltzman's constant.
$\mathrm{L_p}$	Path loss.
M	Number of bits per symbol.
$\mathrm{N_{floor}}$	Noise floor.
$\mathrm{N_{Thermal}}$	Thermal noise.
$\mathrm{P_{b,target}}$	Bit error rate target.
$\mathrm{P_r}$	Received power.
$\mathrm{P_t}$	Transmitter output power.
$\mathrm{RTT_{Net}}$	Network Latency
$\mathrm{R^2}$	Coefficient of determination.
SNR	Signal-to-noise ratio (dB).

snr	Signal-to-noise ratio (linear units).
$\mathrm{T}_{\text{Backhaul}}$	Time for building connections between the eNB and the core network.
T_{Core}	Processing time taken by the core network.
T_{OWD}	Total one way delay.
T_{Prop}	Propagation delay.
T_{Queu}	Queuing delay.
$\mathrm{T}_{\text{Radio}}$	Packet transmission time between the eNB and the UE.
$\mathrm{T}_{\text{Transport}}$	Delay between the core network and the Internet/cloud.
$\mathrm{T}_{\text{Trans}}$	Transmission delay.
T	Temperature.

Chapter 1

Introduction

1.1 Motivation

The data traffic generated by smartphones is expected to increase significantly until 2022, where a large predominance of data compared to voice is observed. Figure 1.1 shows the data traffic expected in 2022.

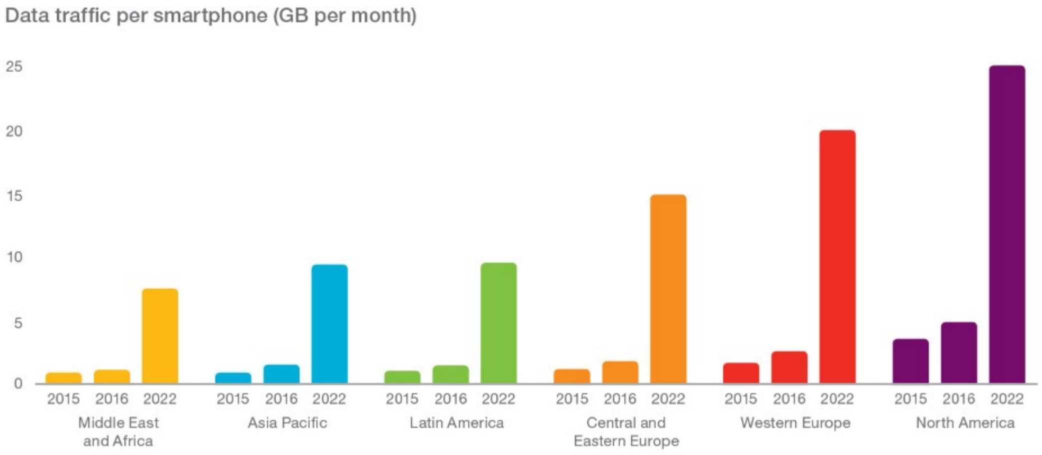


Figure 1.1: Data traffic per smartphone (GB per month) from Ericsson [1].

With the growing number of users, new network services and strict application network requirements it was necessary to think about new solutions based on intelligence and services capable of provide automation and flexibility. A Software-defined network via virtualization was the main idea proposed to reach this goal, combining the two view points, SDN and Network Function Virtualization (NFV).

The Fifth Generation (5G) of Mobile Communications currently being discussed and expected to be standardized and deployed until 2020 [2], aims to provide a wide range of new applications, not only because of the improvements in latency, throughput or capacity, but also due to an entire different way of thinking the network, in both architecture and implementation fields. In order to face these challenges, new solutions based on intelligence, automation and flexibility in control and interfaces management are essential. In this new mobile generation, the network architecture will suffer some adjustments concerning the radio access component, with a Centralized Radio Access Network (C-RAN), composed by base stations antennas (Remote Radio Heads, RRHs) located near the users and processing units (Baseband Units, BBUs) concentrated in a data centre. Along with the 5G, a new trend emerged and is expected to be a promising approach for mobile networks - Software Defined Networks (SDNs). SDN intends to minimize hardware constraints and to abstract lower level functions, by moving them to a standardized control plane, responsible for the management of the network behaviour, through Application Program Interfaces (APIs). From a software-based centralized control plane, network administrators can provide services to the network independently from the connected hardware components. SDN can be used to provide an overall framework which enables 5G to work across a control plane.

1.2 Objectives

The main goal of this thesis is to evaluate the network performance of the next generation networks, here based on concepts as SDN, NFV and C-RAN. To do so, a simplified SDN enabled 4G cellular network is considered, as a step towards 5G, to meet high users expectations and respect the network requirements. Its performance will be evaluated in terms of latency and throughput via network simulation in Mininet-WiFi [3], a Software Defined Wireless Network emulator that enables the testing of a LTE RAN over SDN technology. Modules will be implemented in the emulator to simulate the LTE RAN.

In order to prove the emulator validity, the simulation measurements obtained will be compared with LTE reference values and fitting curves will be developed. These fitting curves will be evaluated according to three metrics: the Root Mean Square Error (RMSE), the coefficient of determination (R^2) and the Pearson Correlation.

The validation of the results will be made in two ways:

- According to reference values from 3GPP and ITU;
- According to *drive-tests*, provided by Celfinet, a Portuguese telecommunications consulting company.

1.3 Thesis Outline

This thesis is organized in five chapters. The first chapter presents a brief introduction to the developed work, presenting the motivation behind it. Chapter 2 presents the state of the art, which consists in an overview of the current 4G network and an introduction to the emerging technologies addressed in this work: SDN, OpenFlow, NFV and C-RAN. Chapter 3 introduces the LTE Radio Frequency (RF) system scenario, which served as reference to setup the RAN along with the other modules implemented; this section also presents the network configuration implemented in Mininet-WiFi, along with this platform description and the performed network tests. In Chapter 5, the throughput and latency results are presented and compared with typical LTE values. An use case is also presented, addressing field measurements through the use of drive-tests. At the end of this chapter, a second use case regarding energy savings is briefly addressed. In Chapter 6 some conclusions are drawn relatively to the performance results obtained. Future work is also described in this chapter.

1.4 Publications

One paper was submitted and accepted under this thesis scope, to the 12th International Conference on Communications (COMM), at Bucharest, Romania, in June of 2018:

- Santos I., Vieira P., Borralho R., Queluz M.P., Rodrigues A., ”Emulating a Software Defined LTE Radio Access Network Towards 5G”, 12th International Conference on Communications (COMM), Bucharest, Romania, June 2018.

Chapter 2

State of the art

2.1 Long-Term Evolution

The 3rd Generation Partnership Project (3GPP) released the Long Term Evolution standard, often called Forth Generation (4G) in 2008 as an all IP System Architecture Evolution (SAE).

2.1.1 Architecture

In Figure 2.1, the generic architecture of a Long-Term Evolution (LTE) network is presented, comprising the services, the core network Evolved Packet Core (EPC), the access part Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and the User Equipment (UE).

The access network consists on base stations, the eNBs (evolved NodeBs), and the connections between them through X2-interfaces, resulting in a flat architecture called the E-UTRAN. In E-UTRAN the eNBs provide Evolved Universal Terrestrial Radio Access (E-UTRA) user plane and control plane protocols towards the user equipment, and have base station functions managing radio resources that were executed by the Radio Network Controller (RNC) in Third Generation (3G) networks.

The core network in this topology is the EPC and it is responsible for the overall control of the UE and the establishment of carriers, and is connected to the access network by the S1-interfaces.

The main logical nodes of the EPC explained in [4] and [5], are the following:

- The Serving Gateway (S-GW) works as a local mobility anchor, enabling communications when the user moves from one eNB to another, therefore handles frequent changes in user's location and stores a large amount of states since all user IP packets are transferred through the Serving Gateway. Also, it performs some administrative functions and tunnels traffic to the Packet data network Gateway (P-GW).

- The P-GW enforces QoS policies, monitors traffic to perform billing and is responsible for IP address allocation for the UE. The P-GW also connects the EPC to other cellular and packet data networks, including the Internet.

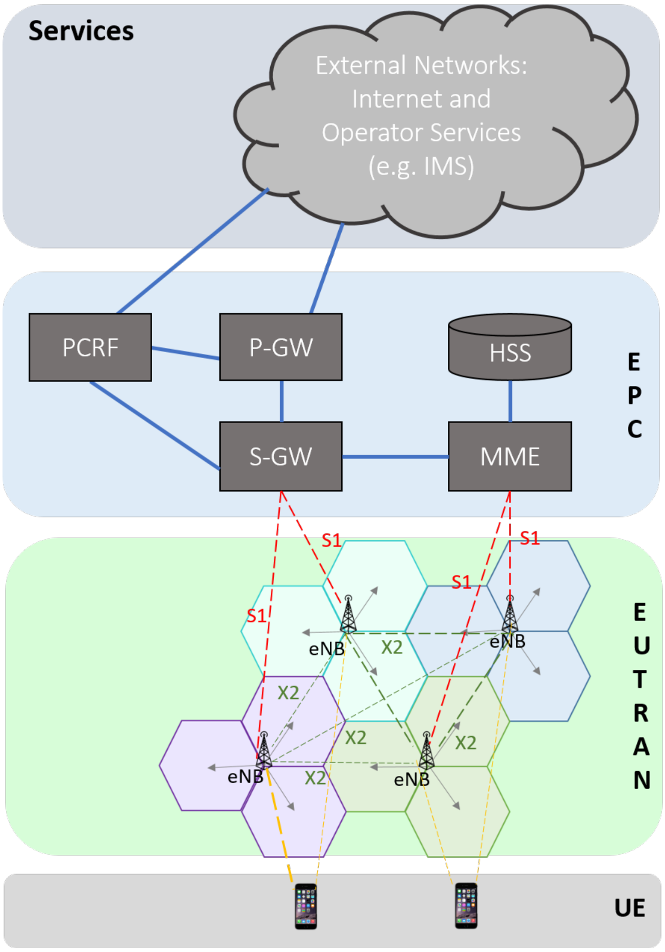


Figure 2.1: LTE architecture.

- The Mobility Management Entity (MME) is the main control plane element in the EPC and is the node that processes the signalling between the UE and the EPC.

- The Policy Control and Charging Function (PCRF) not only manages the flow-based charging in the P-GW, but is also the node responsible for providing the QoS authorization (QoS class identifier and bit rates that decide how to treat each traffic flow), based on the user's subscription profile.

- The Home Subscriber Server (HSS) contains subscription information for each user (e.g., QoS profile or access restrictions for roaming) and holds dynamic information such as the identity of the MME to which the user is currently attached or registered.

Today's cellular network architectures present some limitations, as observed in [4]. For example, the data-plane functions, such as monitoring, access control, and QoS, centralized at the packet gateway, introduce scalability challenges and make the equipment very expensive. Other problem is the fact that interfaces configurations are vendor-specific which makes the equipment not flexible, and the existing control-plane protocols are overly complex.

2.1.2 Radio Interface

LTE supports two different duplex modes to separate the transmission directions between the user and the base station: Frequency Division Duplex (FDD) and Time Division Duplex (TDD).

In Europe, FDD is the widely adopted duplex mode, and the most relevant bands correspond to 800 MHz, 900 MHz, 1800 MHz and 2.6 GHz, being also adopted by Portugal's communications sector regulator (ANACOM) [6].

The LTE radio interface is based on two multiple access techniques, represented in Figure 2.2:

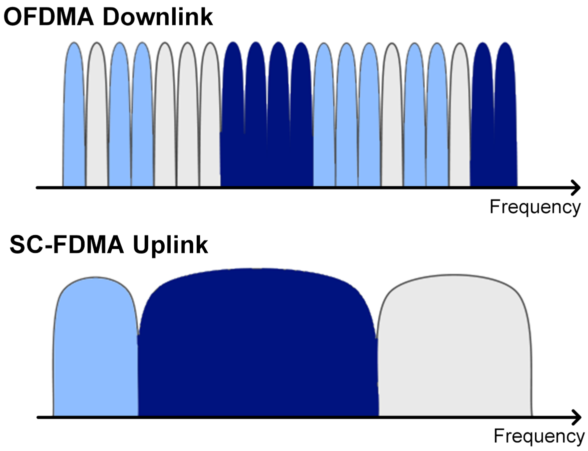
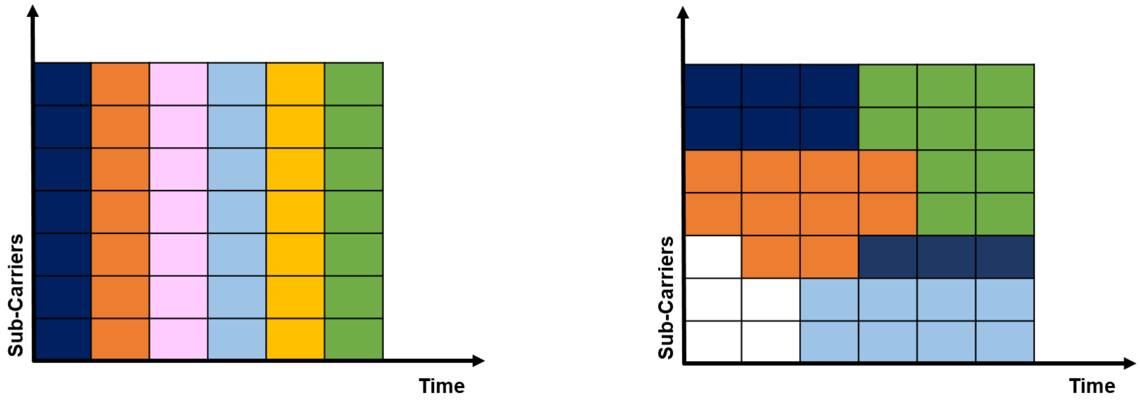


Figure 2.2: OFDMA vs SC-FDMA (adapted from [7]).

- Orthogonal Frequency Division Multiple Access (OFDMA) for Downlink (DL): a multi-carrier scheme that allocates radio resources to multiple users;

- Single Carrier Frequency Division Multiple Access (SC-FDMA) for Uplink (UL): data spreads across multiple subcarriers.

In DL, a OFDMA signal uses Orthogonal Frequency Division Multiplex (OFDM); however, the scheduling and assignment of resources is different. For LTE, OFDM splits the carrier frequency bandwidth into many small subcarriers, spaced by 15 kHz, and then modulates each individual subcarrier using the QPSK, 16-QAM or 64-QAM digital modulation formats. The OFDMA assigns to each user the bandwidth needed for their transmission. Unassigned subcarriers are off, thus reducing power consumption and interference [8]. These two techniques are presented in Figure 2.3.



(a) OFDM

(b) OFDMA

Figure 2.3: Difference between OFDM and OFDMA (adapted from [8]).

A group of twelve subcarriers (occupying a total of 180 kHz) is called a Resource Block (RB), and depending on the carrier bandwidth the system supports a specific number of resource blocks presented in Table 2.1. The smallest unit of resource is the Resource Element (RE), which consists of one subcarrier for a duration of one OFDM symbol. A resource block is thus comprised of 84 resource elements in the case of the normal cyclic prefix length, and 72 resource elements in the case of the extended cyclic prefix [7].

The resource structure is shown in Figure 2.4 for the normal cyclic prefix length.

Table 2.1: Number of resource blocks for each bandwidth.

Bandwidth [MHz]	1.4	3	5	10	15	20
Number of sub-carriers	72	180	300	600	900	1200
Number of RBs	6	15	25	50	75	100

In the UL, LTE uses a pre-coded version of OFDM called SC-FDMA. For UL, the use of OFDMA is not ideal because of its high Peak to Average Power Ratio (PAPR) when the signals from multiple subcarriers are combined [9], [8]. For this reason, SC-FDMA was selected, since the terminal can handle these challenges with more efficiency.

In LTE the Multiple Input Multiple Output (MIMO) operation emerged, which allows to increase of the peak data rate by a factor of 2 or 4 for a 2x2 or 4x4 antenna configuration, respectively.

Peak bit rates are directly related to the RB characteristics and bandwidth. Table 2.2 and Table 2.3 show the peak bit rates in DL and UL, respectively. Concerning the modulation, LTE uses both Quadrature Phase Shift Keying (QPSK) and Quadrature Amplitude Modulation (QAM).

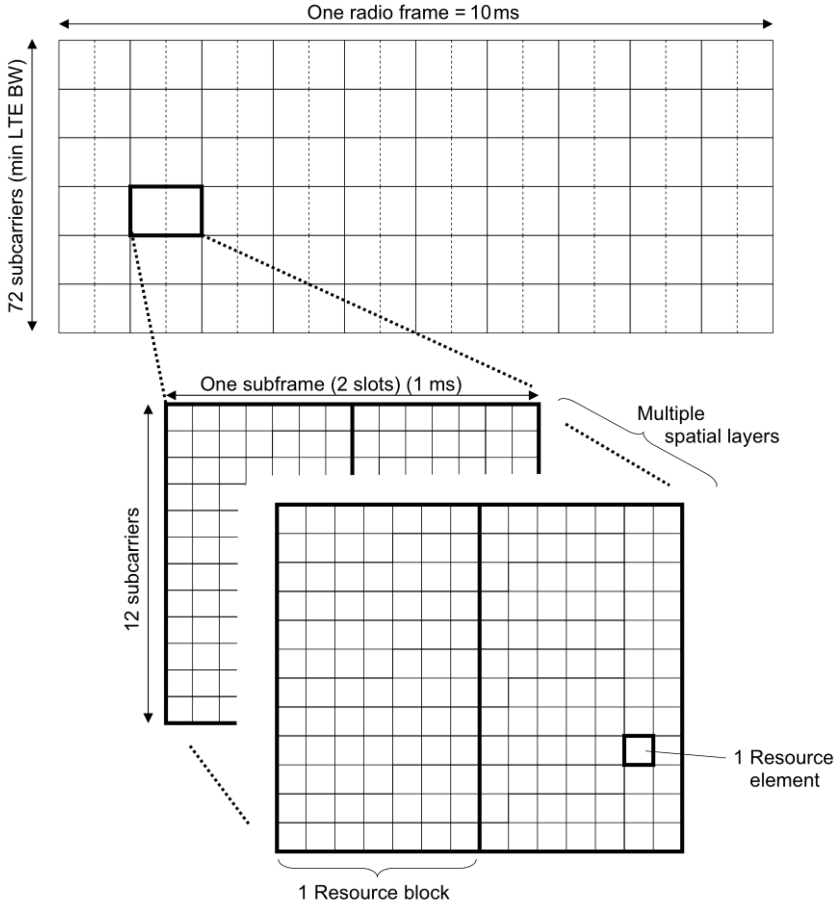


Figure 2.4: Resource block representation (extracted from [7]).

Table 2.2: Downlink peak bit rates in LTE (adapted from [10]).

Downlink peak bit rates in LTE [Mbit/s]								
Resource Blocks			Bandwidth [MHz]					
Modulation and coding	Bits per symbol	MIMO	1.4	3	5	10	15	20
QPSK 1/2	1	-	1.0	2.5	4.2	8.4	12.6	16.8
16QAM 1/2	2	-	2.0	5.0	8.4	16.8	25.2	33.6
16QAM 3/4	3	-	3.0	7.6	12.6	25.2	37.8	50.4
64QAM 3/4	4.5	-	4.5	11.3	18.9	37.8	56.7	75.6
64QAM 1/1	6	-	6.0	15.1	25.2	50.4	75.6	100.8
64QAM 3/4	9	2x2	9.1	22.7	37.8	75.6	113.4	151.2
64QAM 1/1	12	2x2	12.1	30.2	50.4	100.8	151.2	201.6
64QAM 1/1	24	4x4	24.2	60.5	100.8	201.6	302.4	403.2

Table 2.3: Uplink peak bit rates in LTE (adapted from [10]).

Uplink peak bit rates in LTE [Mbit/s]							
Resource Blocks		Bandwidth [MHz]					
Modulation and coding	Bits per symbol	1.4	3	5	10	15	20
QPSK 1/2	1	1.0	2.5	4.2	8.4	12.6	16.8
16QAM 1/2	2	2.0	5.0	8.4	16.8	25.2	33.6
16QAM 3/4	3	3.0	7.6	12.6	25.2	37.8	50.4
16QAM 1/1	4	4.0	10.1	16.8	33.6	50.4	67.2
64QAM 3/4	4.5	4.5	11.3	18.9	37.8	56.7	75.6
64QAM 1/1	6	6.0	15.1	25.2	50.4	75.6	100.8

2.1.3 Protocol architecture for E-UTRA

According to 3GPP [11], the E-UTRAN consists of eNBs, providing the E-UTRA User Plane (U-Plane) (Radio Link Control (RLC)/Medium Access Control (MAC)/Physical (PHY)) and Control Plane (C-Plane) Radio Resource Control (RRC) protocol terminations towards the UE. The eNBs are connected to the Access Gateway (aGW) via the S1 interface. The Figure 2.5 gives an overview of the E-UTRAN architecture where white boxes depict the functional entities of the C-Plane, and blue boxes depict the functional entities of the U-Plane [11].

Focusing on the U-Plane, Figure 2.6 shows the U-Plane protocol stack for E-UTRAN, where:

- RLC and MAC sublayers (terminated in eNB on the network side) perform functions, such as scheduling, Automatic Repeat Request (ARQ) and Hybrid automatic repeat request (HARQ).

  In ARQ, when we have a ’bad’ package, the system simply discards it, and asks for a retransmission (of the same package). The HARQ is the use of conventional ARQ along with an Error Correction technique called ’Soft Combining’, which no longer discards the received bad data (with error).

  HARQ is essentially a combination of Forward Error Correction (FEC) with Automatic Retransmission reQuest, in an optimal manner. Hybrid ARQ schemes are commonly used to provide a reliable communication over noisy wireless channels. HARQ is able to compensate for link adaptation errors and provide a finer granularity of coding rate thus giving a better throughput performance.

- PDCP sublayer (terminated in aGW on the network side) performs for the U-plane functions: Header Compression, Integrity Protection (to be determined during WI phase) and Ciphering.

2.1.4 Similarities with WiFi

Both WiFi and LTE, uses OFDM as the underlying modulation method. Since both technologies use OFDM signals, the decoding scheme will be similar in both cases. However, in the downlink, LTE

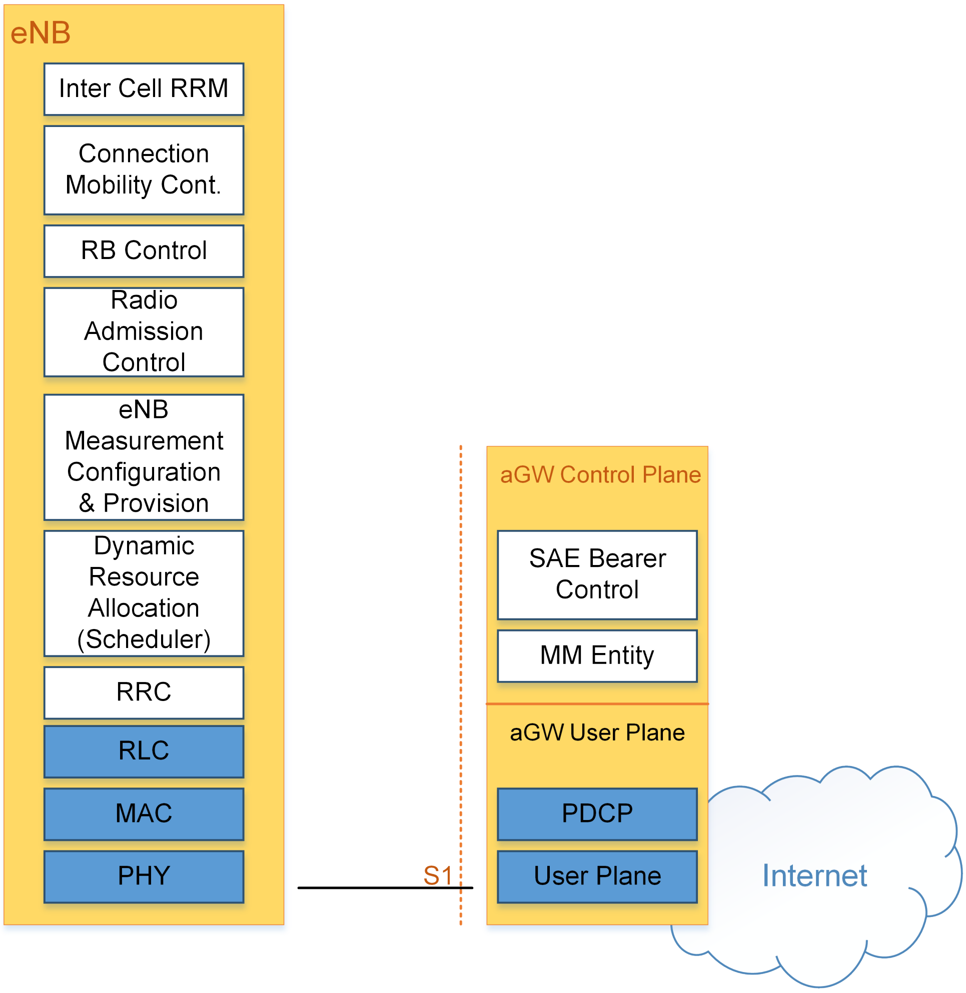


Figure 2.5: E-UTRAN architecture overview (adapted from [11]).

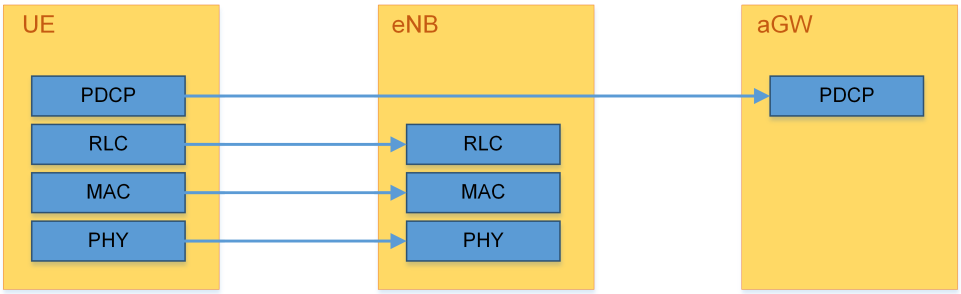


Figure 2.6: U-plane protocol stack (adapted from [11]).

uses OFDMA as the multiplexing scheme and WiFi uses Carrier Sense Multiple Access (CSMA) as the multiplexing method.

Regarding the coexistence between both technologies, 3GPP suggests a LTE-WLAN aggregation; however a specific solution is not defined for this use case. In the aggregation architecture, a Wireless Local Area Network (WLAN) Access Point (AP) will be similar to a secondary eNB in the user plane [12], sharing the same core network.

In terms of the core network, both WiFi and LTE are all IP based networks and present a similar topology, as presented in Figure 2.7, so they can be approximated by the connection of Ethernet switches. As mentioned in [13], EPC also supports the integration with non-3GPP radio technologies such as WLAN with which it shares the core network based on IP. The E-UTRAN is connected through the S-GW and WLAN by the evolved packet data gateway (ePDG), with both providing extra security functionalities.

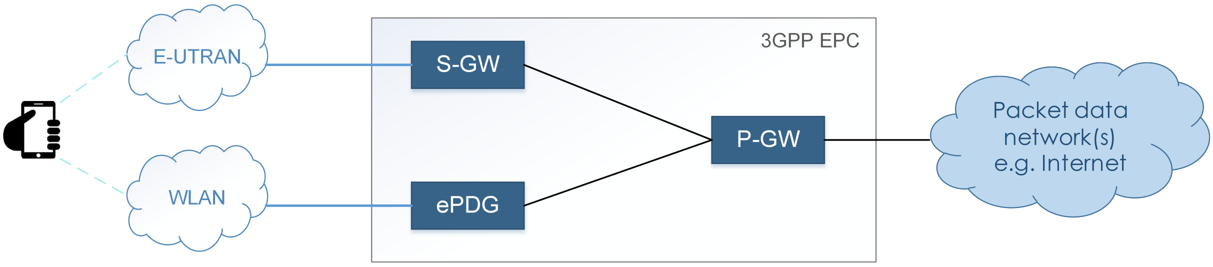
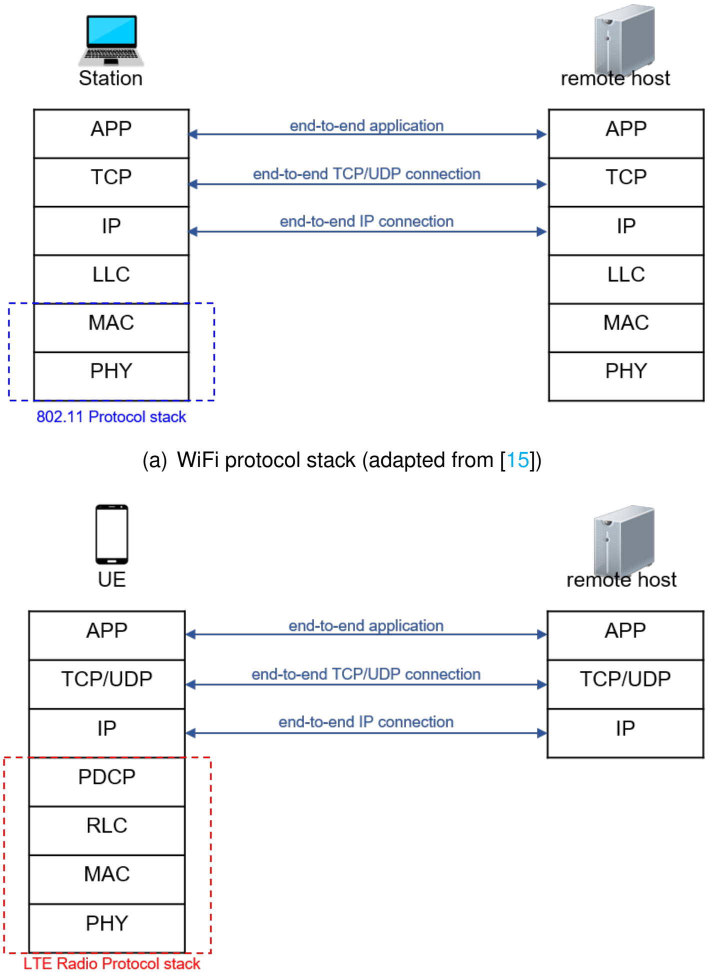


Figure 2.7: LTE WLAN connection (adapted from [14]).

Concerning the networks protocols, Figure 2.8 shows the two technologies protocol stacks.



(a) WiFi protocol stack (adapted from [15])

(b) LTE protocol stack (adapted from [16])

Figure 2.8: Connection protocols.

In WiFi, the Logical Link Control (LLC) layer provides an interface between the MAC layer and the higher layers, and performs flow and error control.

In LTE, it is the PDCP/RLC/MAC/PHY that is responsible for the IP packets transmission between the

UE and the eNB, over the air interface. The Application (APP) layer, the Transmission Control Protocol (TCP) layer, transport layer, and the IP layer allow the end-to-end communications, between the mobile terminal and the remote host (i.e., the internet).

2.2 Software Defined Networks

According to the Open Networking Foundation (ONF)[1], SDN is considered one of the most promising technologies to realize virtual networks, where network control is decoupled from forwarding actions and is directly programmable. Network intelligence is logically centralized in the software-based SDN controller, constituted by multiple physical and virtual components, but still behaves like a single node.

SDN follows four key principles:

- **Separation of control and data planes**

  The control plane migrates from the network forwarding devices to a separate unit, the controller, with the ability to change the forwarding behaviour of these devices. This requires changes in vendors hardware, that will become cheaper.

- **Logical and efficient centralized control**

  The control of the entire network is managed from a single logical point, simplifying the network design and operations. The components no longer need to understand and process protocols and just accept SDN controller's instructions. This global network view, enables the controller to adapt network policy in terms of routing and forwarding in a faster way.

- **Open Interfaces**

  For the network to be flexible and adaptable, it is essential that interfaces are open. This factor enables the development of software that can control the traffic exchange, along with possible inspection and modification of the traffic flow, and open to innovation.

- **Programmability**

  Introducing programmability principles, the controller starts to treat the network as a single programmable entity instead of a set of devices that have to be configured individually. This measure meets the principle of network virtualization.

The SDN architecture is presented in Figure 2.9 and is divided in three layers.

In [17], the **Application layer** represents the functional applications of the network as services provided by the network operator, like access control, traffic/security monitoring and energy-efficiency. In other words, it can be seen as an end-user business applications that consume SDN communications services. The **Control layer** consists in logically centralized controllers that supervise the network forwarding behaviour through an open interface. The **Infrastructure layer** involves the physical network elements and devices that provide packet switching and forwarding, presented in Figure 2.9 in three

[1] user-driven organization dedicated to accelerating the adoption of SDN and NFV

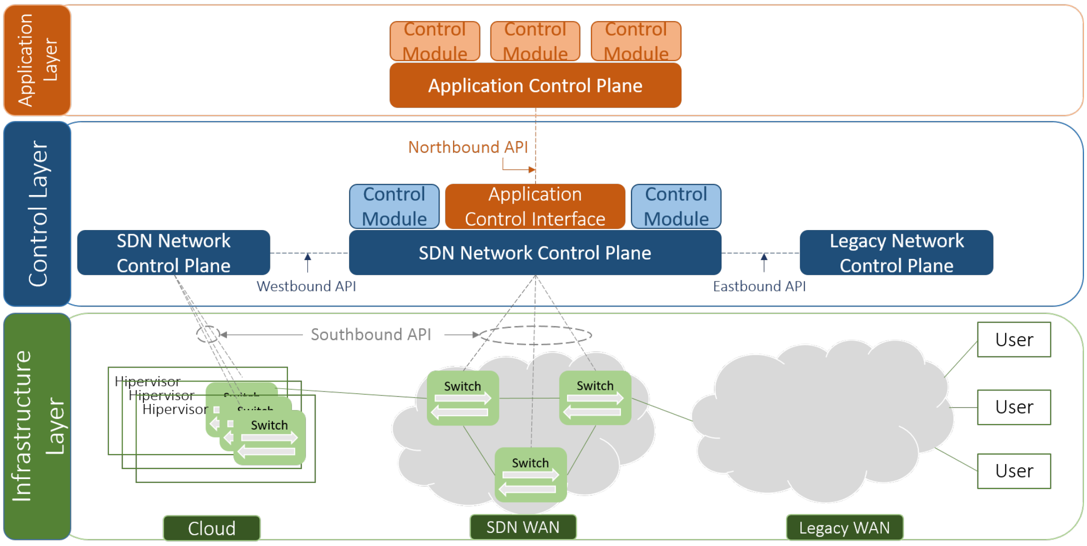


Figure 2.9: SDN overview (adapted from [17]).

Autonomous Systems (AS): a conventional IP or legacy access network, an SDN-based legacy Wide Area Network (WAN)[2], and an SDN-based data center network (cloud). This last layer also represents the data plane.

Westbound Application Programming Interface (API) is defined as an information conduit between SDN control planes of different network domains, allowing the exchange of network state information to influence routing decisions of each controller. On the other side, there is the *Eastbound API* which work as the communication channel within the control planes of non-SDN domains.

The *Northbound API* represents the software interfaces between the software modules of the controller and the SDN running applications that consume the communication services. These software interfaces are open source-based.

The *Southbound API* is especially important since it is the one responsible for the communication between the control and data planes, and so, it will be explained in more detail. The most known, and whose analysis will be addressed, is OpenFlow.

OpenFlow

The OpenFlow specification is controlled and defined by the nonprofit Open Network Foundation (ONF) [18], which is managed by a board of directors from seven companies that own and operate some of the largest networks in the world (Deutsche Telekom, Facebook, Google, Microsoft, Verizon, Yahoo, and NTT). The same reference defines OpenFlow as the protocol that provides software-based access to the flow tables that instruct switches and routers, on how to direct network traffic. Using these flow tables, administrators can quickly change network layout and traffic flow.

An OpenFlow Switch, presented in Figure 2.10, consists of one or more flow tables and a group table, which performs packet lookups and forwarding, and an OpenFlow channel to communicate with

[2] old network, which is rarely used today and not part of the TCP/IP protocol suite;mostly proprietary to individual vendors.

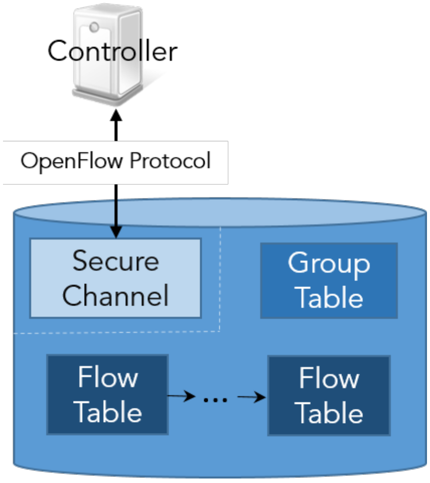


Figure 2.10: OpenFlow methodology [19].

an external controller. This controller manages the switch via OpenFlow protocol, being able to add, update and delete flow entries in the flow tables. According to [18], a flow could be a TCP connection, packets with the same MAC address or IP address, packets with the same Virtual Local Area Network (VLAN) tag, or packets arriving from the same switch port.

The flow entries that constitute the flow tables, presented in Figure 2.11 and described in [19], are:

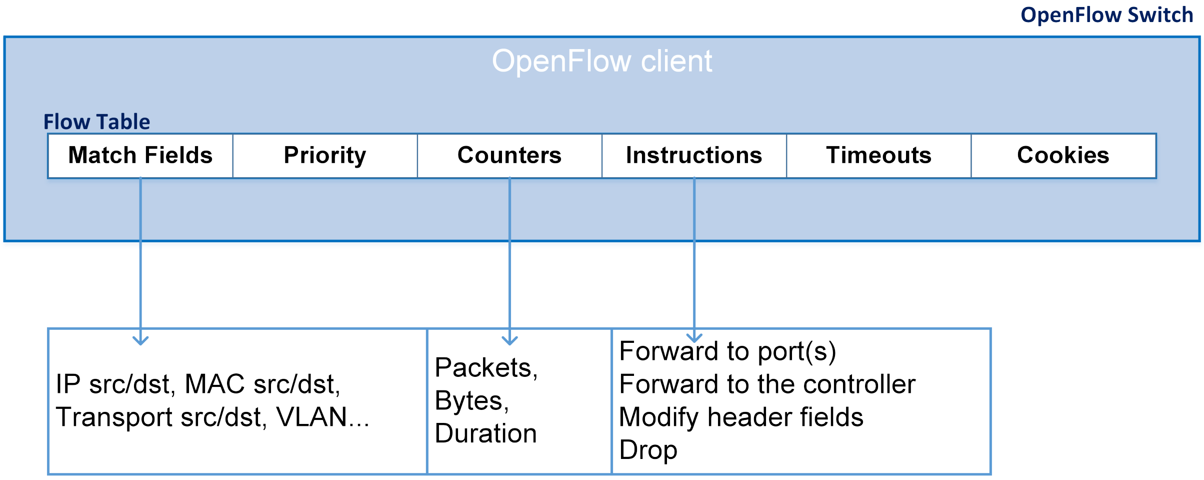


Figure 2.11: Flow entry details (adapted from [20] and [19]).

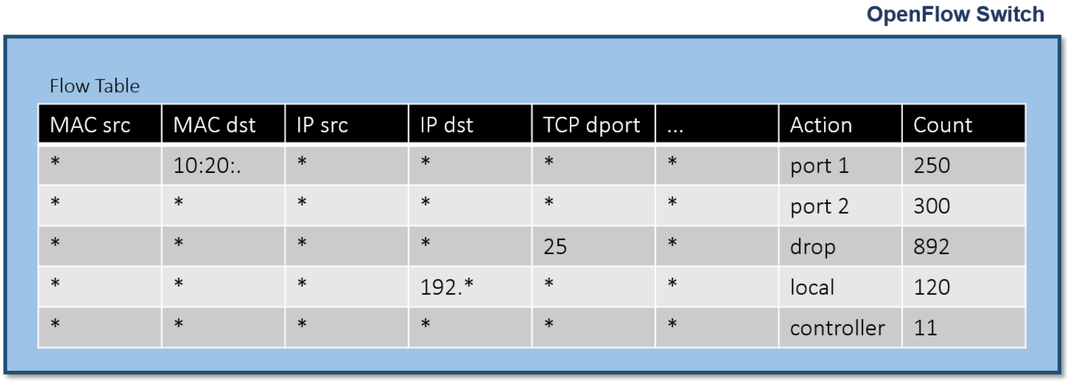
- **Match fields:** Ingress port and packet headers, and optionally metadata specified by a previous table. It contains information such as VLAN ID, source and destination ports, IP address, and Ethernet source and destination.
- **Priority:** match precedence of the flow entry.
- **Counters:** to update for matching packets, keeping track of the number of packets and bytes for each flow and the elapsed time since flow initiation.
- **Instructions:** to modify the action set or pipeline processing. It specifies how the packets in a flow

will be processed, and its actions can be: Forwarding the packet to a given port/ports; Drop the packet; and Forward the packet to the controller.

- **Timeouts:** maximum amount of time or idle time before flow is expired by the switch.

- **Cookie:** opaque data value chosen by the controller, used to filter flow statistics, flow modification and flow deletion. Not used when processing packets and include the number of packets, number of bytes and the time since the last packet matched the flow (for each type of flow).

Figure 2.12 presents an example of a filled flow table.


OpenFlow Switch

Flow Table

MAC src	MAC dst	IP src	IP dst	TCP dport	...	Action	Count
*	10:20:.	*	*	*	*	port 1	250
*	*	*	*	*	*	port 2	300
*	*	*	*	25	*	drop	892
*	*	*	192.*	*	*	local	120
*	*	*	*	*	*	controller	11

Figure 2.12: Flow Table example [21].

Matching starts at the first flow table, and may continue to additional flow tables. Flow entries match packets and if a matching entry is found, the instructions associated with the specific flow entry are executed. If no match is found in a flow table, the packet may be forwarded to the controller over the OpenFlow channel, dropped or may continue to the next flow table, depending on the flow entry. A group table consists of group entries that offers additional methods of forwarding (e.g., multicast, broadcast or link aggregation). A group entry consists of a group identifier, a group type, counters, and a list of action buckets containing a set of actions to be executed.

Software Defined Mobile Networks

Integrating SDN into mobile networks towards the Software Defined Mobile Network (SDMN) concept, can be done in several ways. One of them is the controller integration with the S/P-GW to control the transport network [22]. This option consists of decoupling the S/P-GW into the control and data planes. The control part of the S/P-GW (S/P-GWc) provides IP address allocation for the UE. The data plane of the S/P-GW (S/P-GWu) provides the GTP tunneling termination endpoint and the anchoring of the GTP tunnels, during the handover process. The control part of the S/P-GW is integrated with the SDN controller and performs data filtering. The rest of the network elements are not changed, and the MME interacts with the S/P-GWc. This SDN integration into the LTE architecture is represented in Figure 2.13.

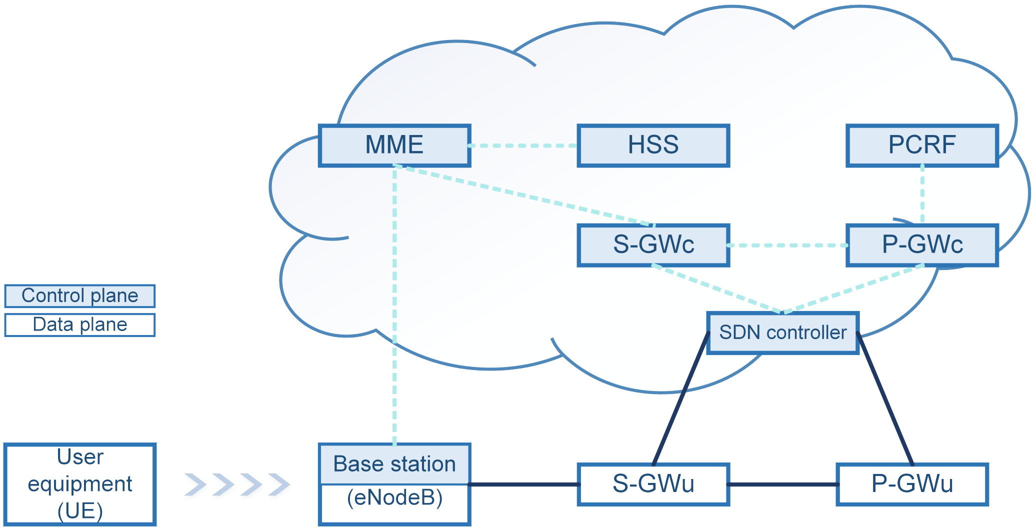


Figure 2.13: Controller integration with the S/P-GW (adapted from [22]).

In [22] an architecture is suggested where the applications manage the network. The OpenFlow switches are placed as the first aggregation element to which eNBs are connected. The eNB will terminate the protocol stack over the air interface and send all traffic from an user to an Ethernet VLAN using 802.1ad encapsulation. The second OpenFlow switch (mOFS) is required before the entry point to the Internet. The mOFS will tag and route the packets from the Internet to the right eNB and the right mobile device.

In this scenario, the same MME, when receiving the signalling from an SDN-based eNodeB through the S1-MME interface, will establish the connection with the termination SDN switch over L2.

In [23] a proposal to integrate OpenFlow in the LTE architecture is presented. The eNodeB now communicates directly with the P-GW (data plane) over OpenFlow switches (layer 2 switching) since the S-GW entity is removed from the architecture in order to reduce the stack horizontally. The request and response messages sent from the MME to the S-GW must be forwarded to the OpenFlow controller. This controller then communicates with the P-GW. The user plane protocol stack would be as as shown in Figure 2.14.

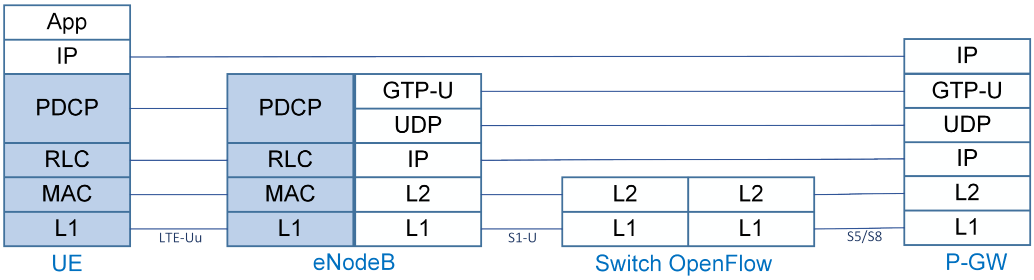


Figure 2.14: 4G LTE User Plane Protocols Stack with OpenFlow (adapted from [23]).

On the other hand, [23] foresees that it will be required to reduce the stack vertically by removing the GTP tunnels and performing pure layer 2 switching, i.e., by setting up flows with the OpenFlow

Controller to the different entities. Significantly reducing the number of protocol layers by two levels in the core network, will improve the network traversal time and the end-to-end delay of all communications. The new stack of protocols using OpenFlow is presented in Figure 2.15.

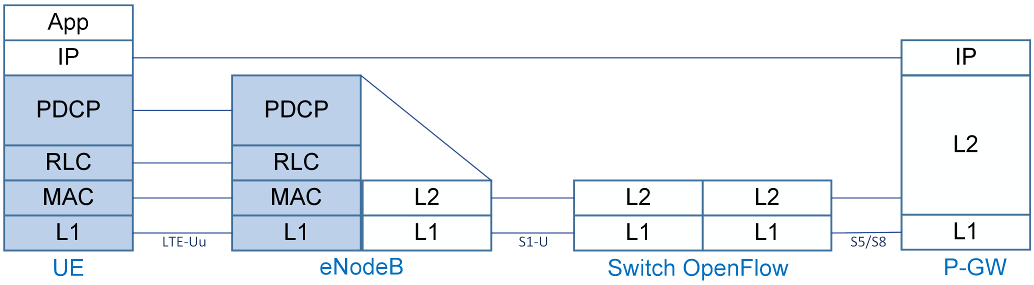


Figure 2.15: 5G User Plane Protocols Stack with OpenFlow (adapted from [23]).

Mobile Core gateway re-design

In [24] a mobile core gateway re-design is proposed, where the gateway's functions are allocated in virtualized and SDN decomposed deployments, in addition to the required transport network elements.

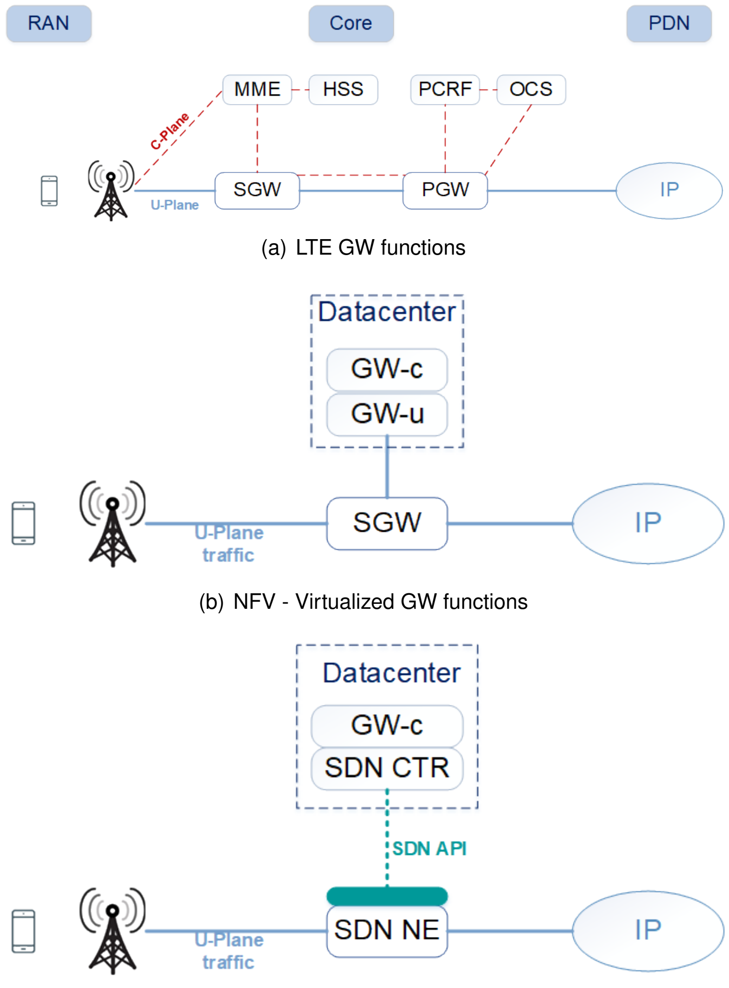
1. **Virtualized Gateway**

   Applying the concept of NFV to the current gateway architecture, results in moving the control plane (GW-c) and user data plane (GW-u) functions to the operator's datacenter as shown in Figure 2.16(b). Thus, a Network Element (NE) will be used, which can be either a typical transport switching element or a standard SDN network element, to direct the data traffic from the transport network to the intended data center. All further processing would be done by the virtualized gateway.

2. **SDN Decomposed Gateway**

   With SDN decomposed gateways, SDN controllers are extended with control-plane gateway functions as for example LTE signaling, and hosted within operator deployed datacenters. In this case, enhanced SDN network elements which resembles the previously mentioned dataplane gateway functions such as GTP tunneling or charging, which cannot be realized with current SDN network elements. In addition, an extended SDN API that should be adopted by the operator, to program and control those features.

These new gateways are presented in Figure 2.16(b) and 2.16(c), along with the typical LTE architecture, represented in Figure 2.16(a).



(a) LTE GW functions

(b) NFV - Virtualized GW functions

(c) SDN - Decomposed GW functions

Figure 2.16: Gateway architectures [24].

2.3 Network Function Virtualization

NFV is about using virtualization to implement network functions on commodity servers in data centres, rather than on proprietary hardware, running expensive proprietary software. It enables abstraction and sharing of infrastructure and radio spectrum resources. With NFV, network functions can be executed in various locations in the network as required, without the need for installation of new equipment.

In [25] several differences between NFV based networks and the current ones are introduced, such as:

- **Decoupling software from hardware**

  A network element is no longer the combination of both software and hardware entities, which makes the evolution of each part independent. This evolution allows the progression of software, and hardware, separately.

- **Flexible network function separation**

  The deployment of software from hardware allows the reassignment and sharing of network infrastructures, making the network more automated and being a support for cloud technologies.

This automation also helps the network operators to rapidly introduce new services faster in the network over the same physical platforms.

- **Dynamic operation**

  The network functions going into software components provide greater flexibility to scale the actual Virtual Network Functions (VNFs) according to the traffic, helping operators provisioning capacity.

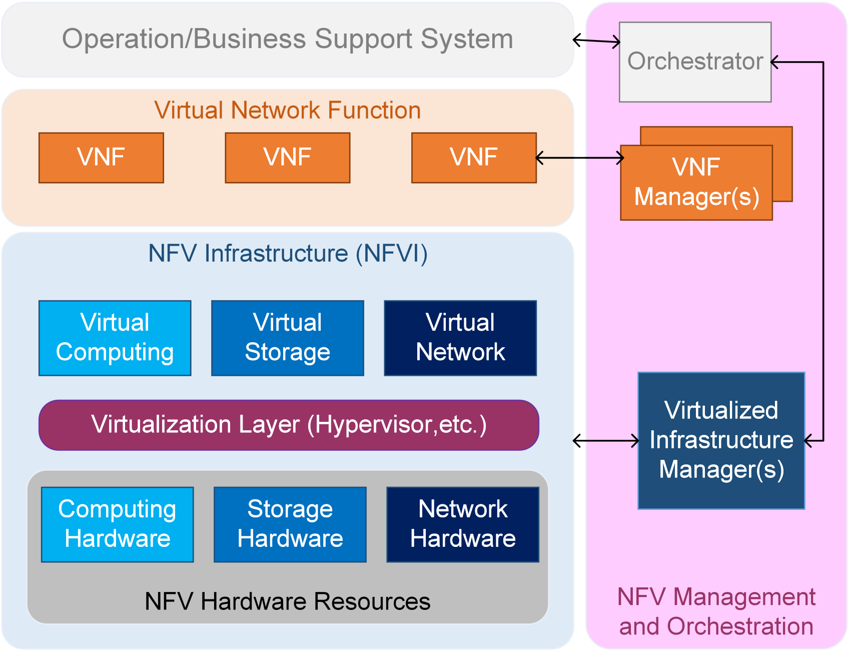


Figure 2.17: NFV framework (adapted from [25]).

The European Telecommunication Standards Institute (ETSI) Industry Specification Group for NFV formulated a framework [25], represented in Figure 2.17, divided in three domains:

- Virtualized Network Function: software implementation of a network function capable of running over the Network Function Virtualization Infrastructure (NFVI).

- Network Function Virtualization Infrastructure (NFVI): set of physical resources and the way they can be virtualized; supports the execution of the VNFs.

- NFV Management and Orchestration: platform responsible for the dynamic initiation and orchestration of VNF instances, as well as for the management of the NFVI.

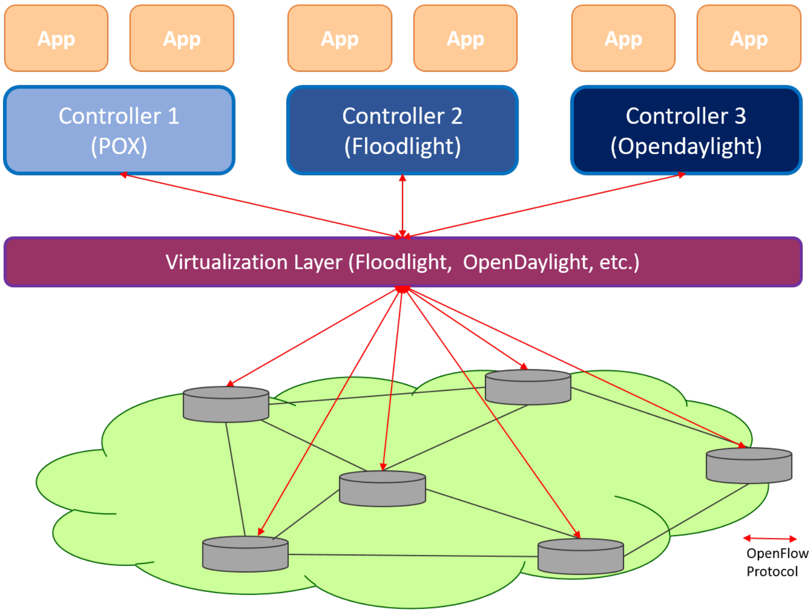

App
App
App
App
App
App
Controller 1 (POX)
Controller 2 (Floodlight)
Controller 3 (Opendaylight)
Virtualization Layer (Floodlight, OpenDaylight, etc.)
OpenFlow Protocol

Figure 2.18: NFV over a SDN network.

In SDN, with Openflow protocol and controllers such as Floodlight, OpenDaylight or POX, virtual networks can be created. These controllers allow the underlying Openflow-based physical network to be sliced into multiple isolated virtual networks, and to each slice, a specific controller is delegated. In Figure 2.18, the integration of NFV in a SDN network is represented, and comparing with Figure 2.17 some of the NFV domains can be seen as well as their place in the new and adapted SDN network.

In [26] some requirements for NFV and SDN integration in mobile networks are addressed:

- **Migration**: A technology to be part of 5G should provide a clear migration path with proper compatibility with the legacy systems. This allows the incremental updates of network elements in certain parts of the network.

- **Security**: Should be considered for all layers, in network functions as well as physical and virtual elements. Starting with the SDN controller that has access to the whole network architecture, to the actual nodes that perform network functions.

- **Monitoring**. Facilitates verification and validation of Service Level Agreements (SLA), managing performance (Quality of Service QoS) and user experience (Quality of Experience QoE), troubleshooting, and the assessment of optimizations and use of resources.

- **Service provisioning and optimization**: This assumption can be deployed in SDN networks by using control applications that have full view of the network configuration. This, together with status information provided by network monitoring and data collection systems, enables mobile network orchestrator applications to optimize services (e.g. latency) and/or resources usage easier than traditional networks, which need to rely on signalling.

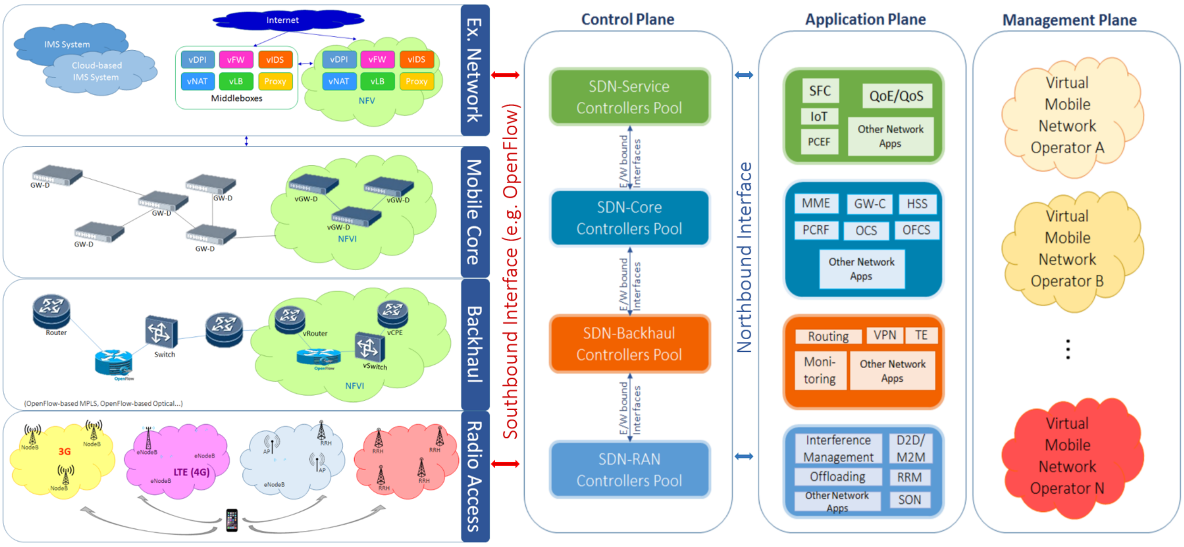


Figure 2.19: General architecture of a SDN and virtualization-based mobile network (adapted from [27]).

Figure 2.19 shows how SDN technology and virtualization are being used in mobile networks, through a generic mobile network, mapped into the SDN reference architecture addressed in [27]. The figure represents, in the vertical section, the reference model for the current mobile networks, which is divided in three parts: the Radio Access Network (RAN), the mobile backhaul and the mobile packet core. Horizontally, the four planes of a SDN architecture: the data plane, control plane, application plane and management plane, are represented. In this figure, the RAN part includes different access technologies such as Global System for Mobile Communications (GSM), Universal Mobile Telecommunication System (UMTS), LTE and WiFi, and all these radio access networks can be programmable and put under the supervision of a pool of SDN controllers. The backhaul network, connects the radio access network and the mobile core network. The equipment that constitutes this network, can be dedicated hardware-based and programmable switches, routers or virtualized switches that are realized as Virtual Machines (VMs) running on a cloud computing system or NFVI. The core network, in contrast to the traditional mobile core networks, is composed of simple network gateways called GW-Ds, which can either be dedicated hardware servers or software appliances, running on the NFVI environment. These GW-Ds act as an anchor point for intra-/inter-handover in the mobile network, and are a point of connection to external networks (e.g., Information Management System (IMS) and Internet). The core network can also be programmed with SDN-core controllers through open APIs. On the top of this architecture stands the external network, that is an IMS or Internet, which provides the services for UEs. The user data traffic needs to pass through a set of middleboxes that help to ensure security, optimize performance, and facilitate remote access, before flowing to the Internet. Some examples of middleboxes include firewalls, load balancers, and WAN optimization. By realizing the concept of NFV, these middleboxes can be deployed as software appliances running on an NFVI environment. For IMS, with the emergence of NFV, traditional IMS is being cloudified, with the migration of proprietary hardware-based systems within a dedicated network infrastructure, into software-based deployments in a cloud infrastructure.

Regarding the SDN architecture component, the four layers previously presented have particular functions. The first layer, data or user plane, enables user data traffic to be delivered through the RAN to the external network. The control layer is a pool of SDN controllers, each one deployed for every part of the network, that can be simplified to a master controller responsible for the control and management of the entire network. SDN controllers can also be deployed as either hardware-based or software-based controllers, running in a NFVI environment. The next layer, the application layer, consists of a series of network control functions placed on the top of the SDN controllers (e.g., interference management, radio resource management and offloading). At last, the management plane, refers to the operator's management, where the network virtualization technology allows multiple network operators to share the same underlying mobile network infrastructure.

2.4 C-RAN

The C-RAN is a centralized, cloud-based new cellular network architecture, that can be adopted by Mobile Network Operators (MNOs) to improve the quality of service with reduced Capital expenditure (CAPEX), operating expenditure Operational expenditure (OPEX), and energy consumption [28]. In a C-RAN, the baseband processing units of a group of base stations are combined into a central server, retaining radio front end at the cell sides. However, this requires connection links with delay of 250 μs to support 5G low latency services [29]. This latency requirement is only achievable with optimization techniques, such as fine-tuned real-time kernel for processing, as proposed in [30].

In [28], the ASTRI's C-RAN processing entities are dynamically grouped and allocated to different virtual machines (VMs). During low traffic periods, the system can perform live migration of the processing entities and can shut down some of the processing cores for power saving or release them for other processing. The software stack includes many of the key open source elements needed to develop optimized NFV software applications. These key elements include Linux, Data Plane, Development Kit (DPDK), Open vSwitch (OVS), OpenDaylight, and others.

Figure 2.20 presents an example of a C-RAN mobile LTE network. The fronthaul part spans from the RRHs sites to the BBU Pool. The backhaul connects the BBU Pool with the mobile core network. At a remote site, RRHs are co-located with the antennas. The RRHs are connected to the high performance processors in the BBU Pool through low latency, high bandwidth optical transport links [31]. Digital baseband are sent between a RRH and a BBU.

Energy savings

Energy in mobile network is spent on power amplifiers, supplying Remote Radio Head (RRH) and Baseband Unit (BBU) with power and air conditioning. In C-RAN, since the number of BBUs is reduced compared with traditional RAN, the electricity cost will be reduced as well. Moreover, in the lower traffic period, e.g. during the night, some BBUs in the pool and RRHs that are not transmitting can be switched off, not affecting the overall network coverage. Another important factor is the decrease of cooling resources, which takes 46% of cell site power consumption as seen in figure 2.21(b) . In [32], is estimated

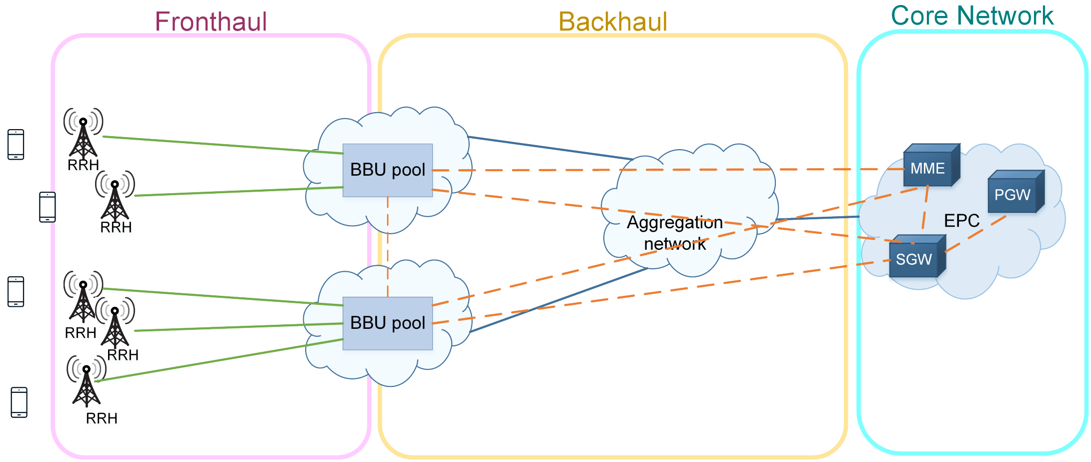
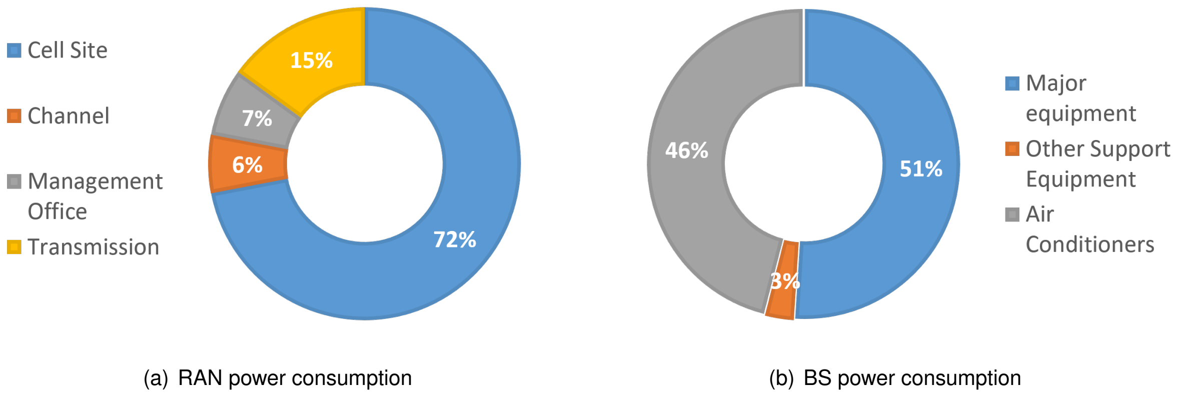


Figure 2.20: C-RAN LTE mobile network (adapted from [31]).

that C-RAN enables 67%-80% power savings compared with traditional RAN architecture, depending on how many cells one BBU pool covers.

In figure 2.21 the graphics show the power consumption of China Mobile [32]. We can see that the majority of power consumption is associated to the Base Stations (BS) in the radio access network (figure 2.21(a)). Inside the Base Station (BS), only half of the power is used by the RAN equipment, while the other half is consumed by air conditioning and other facility equipments.



(a) RAN power consumption

(b) BS power consumption

Figure 2.21: Power consumption of China Mobile (adapted from [32]).

In order to solve, or decrease, this consumption in the network fronthaul, switching hardware and flexible mapping of the connections between BBUs and RRHs are introduced, in order to optimize traffic and energy consumption in the BBU pool. In other way, this one-to-one mapping requires several BBUs to be active and generating frames, which consumes energy in the BBU pool. So, when the traffic load is low in a certain region, a single BBU is enough to cover that region, serving multiple RRHs, and the other BBUs may be switched off, reducing the energy consumption [33]. To do this, the software solutions are capable of turning off selected carriers on idle hours. The second way is the possibility of turnning off RRHs that are no longer transmitting, involving dynamic resource allocation as well as BBU-RRH assignment, depending on traffic load fluctuation during a time interval [34].

Chapter 3

LTE over SDN Implementation

This chapter presents the implemented SDN scenario as well as all the 3GPP LTE references used in order to simulate a LTE over SDN network in Mininet-WiFi, which is originally specified for WiFi. Furthermore, the LTE features implementation in the emulator and the SDN used mechanisms are detailed.

3.1 Emulator Description

Mininet-Wifi [3] emerged as a dependency of Mininet [35], an OpenFlow/SDN emulator capable of adding virtualized stations and access points, based on the standard Linux wireless drivers and the 80211_hwsim [36] wireless simulation driver. This emulator uses the SDN paradigm, which allows network administrators to specify the behaviour of the network in a logic centralized manner through the use of a controller that operates over the forwarding devices through the OpenFlow protocol.

The mobile network topology implemented in the emulator has two parts: the wireless network (representing a LTE RAN network combined with C-RAN concepts) and the wired network (constituted by SDN OpenFlow enabled switches responsible for the data plane functions). The control plane is the responsibility of the controller. The wireless network is composed by the base station antennas, here represented by **RRHs**, and the mobile users, the **UEs**.

The emulator runs in the Linux Operating System (OS), and the system memory in Linux can be divided into two distinct regions:

- The **kernel space** where the kernel (core of the operating system) executes and provides its services. In the kernel-space the module mac80211_hwsim is responsible for creating virtual wireless interfaces, in this case the RRHs and UEs. Moreover, in the kernel-space, Media Access Control Sublayer Management Entity (MLME) is realized in the UEs side. Some of the functions performed by MLME are authentication, association or sending and receiving beacons.

- The **user space** is a set of memory locations in which user processes run. A process is an executing instance of a program. In the user-space the *hostapd*[1] is responsible for the MLME functions in the RRH side.

Some important Linux utilities are used in Mininet-Wifi: the first two are *iw* and *iwconfig*, used for interface configuration and for getting information from wireless interfaces; besides them, another important utility is TC (Traffic control). It is an user-space utility program used to configure the Linux kernel packet scheduler, responsible for controlling the data rate, delay, latency and loss, applying these attributes in the virtual wireless interfaces of UEs and RRHs, representing with higher fidelity the behaviour of the real world networks.

In Figure 3.1 is presented, in a simplified way, the connections between the four main network components and their places in the network (RRHs, UEs, switches (S) and hosts (H)). The *ethx* and *wlanx* represent the wired and wireless interfaces respectively. The hosts, like the traditional Mininet hosts, are connected to switches or RRHs, so UEs are able to communicate with those hosts. The RRHs are configured as OpenFlow switches with BS capabilities.

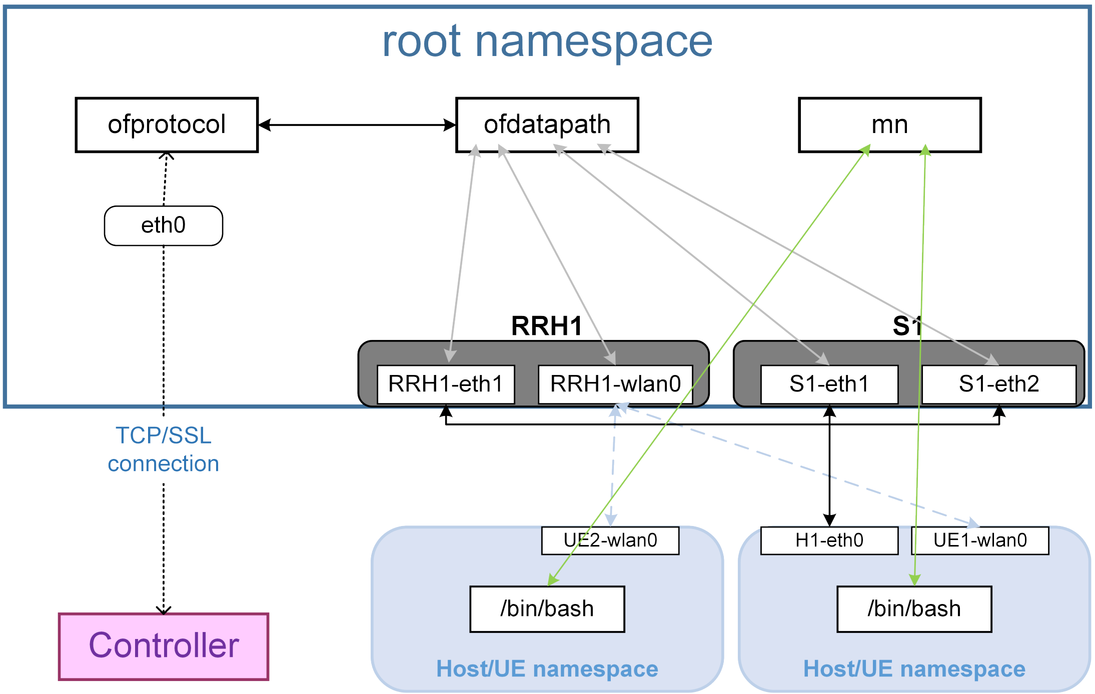


Figure 3.1: Mininet-Wifi architecture (adapted from [37]).

The root namespace, presented in Figure 3.1 includes the forwarding, OpenFlow enabled, elements represented by RRH1 and S1, and is connected to the controller, through the Openflow (OF) protocol, and to the UEs and Hs through the wireless and wired interfaces *wlan* and *eth* respectively. Similar to what happens in Mininet, the virtual network is created by placing hosts and now UEs in the Linux OS network namespaces interconnected through virtual Ethernet (veth) pairs.

According to the components available in the emulator, the topology of Figure 3.2 was built, containing the wireless and wired interfaces described previously in Figure 3.1.

[1] Hostapd (Host Access Point Daemon) user space software capable of turning normal wireless network interface cards into access points and authentication servers

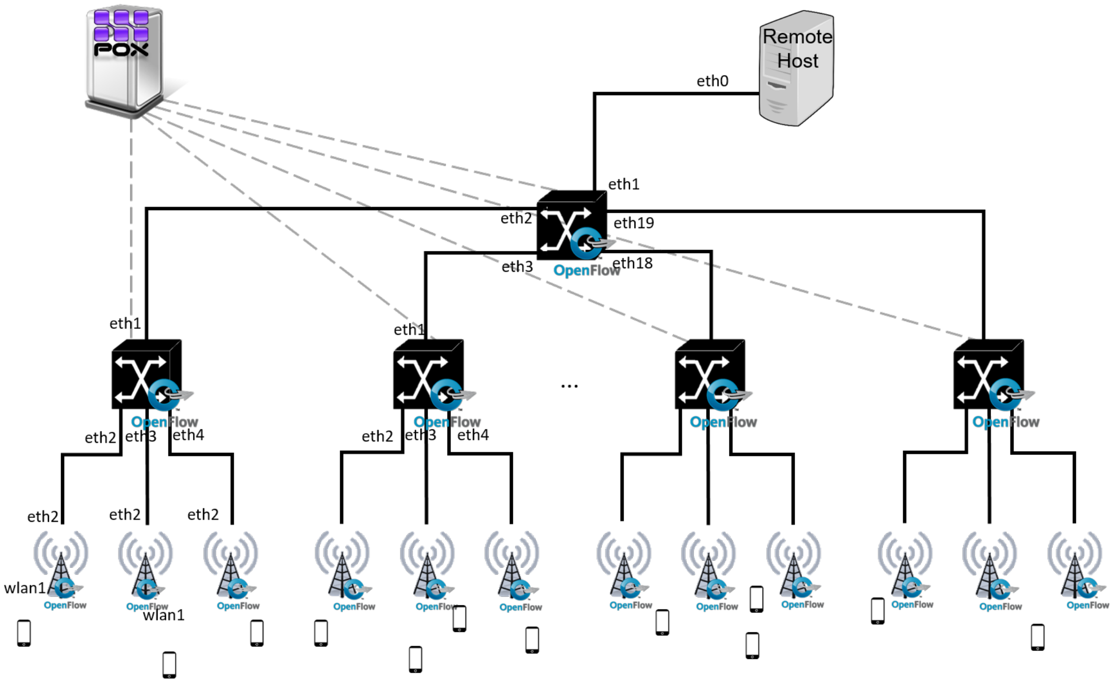


Figure 3.2: Topology implemented in Mininet-WiFi.

The radio access network is constituted by the OpenFlow enabled BS antennas, the RRHs, and the mobile users, the UEs. The backhaul consists on the forwarding switches, also OpenFlow enabled. In this context the node used as host simulates a server as an external network, in accordance with a LTE typical architecture, where the traffic can be directed. This is essentially used to test End-to-End services such as a connection between a UE and the Internet.

Network components

For the development of a mobile network, it was necessary to emulate the mobile network devices, RRHs and UEs. Regarding other network modules, it was used a remote controller running in a separate terminal and accessible through an IP address, an host, acting as a server, and OpenFlow virtual switches.

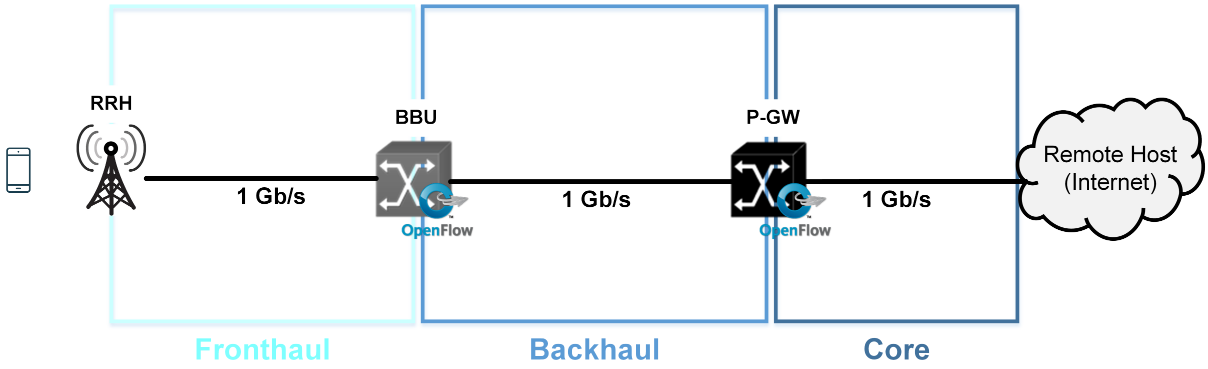


Figure 3.3: Mininet-Wifi architecture and components.

The topology implemented in the emulator represents a simplified version of a LTE network meeting the C-RAN concept described in Chapter 2, and it is presented in Figure 3.3. The RRHs represent simple base station antennas located at the cell sites. The BBUs, in this topology, are represented by OpenFlow virtualized switches. Each BBU serves 3 RRHs, and they are considered centralized since they can communicate with each other using the controller. The fronthaul link, between the RRH and the BBU is considered to be fiber, which seemed the most convenient choice due to fiber availability and applicability.

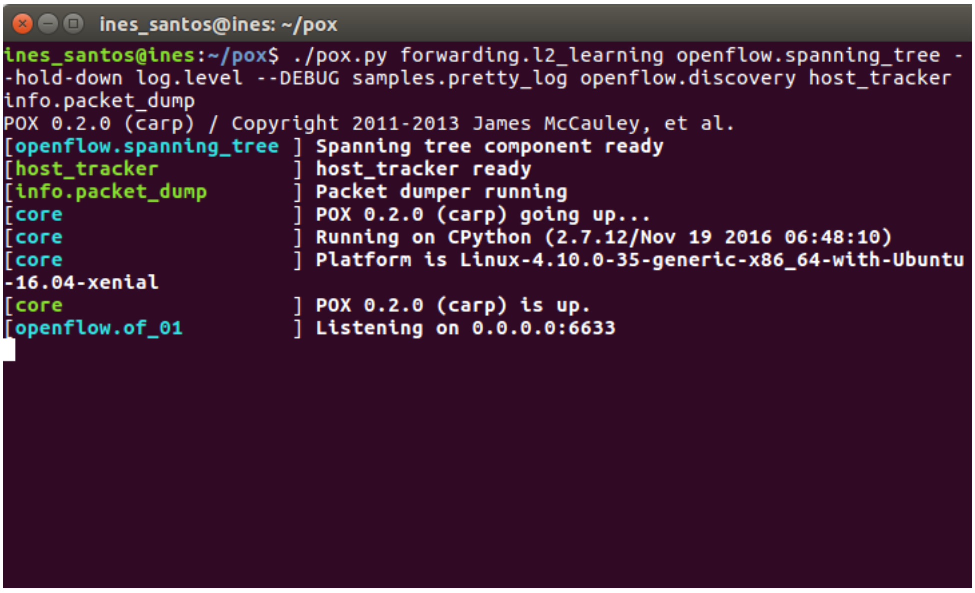
The EPC network, regarding the user plane connections responsible for the packet transport, is reduced to routing-enabled elements. The control part of these network elements is responsibility of the controller.

The topology previously described and implemented in the emulator has the main components:

- **Controller**

  A SDN controller is an application in software-defined networking (SDN) that manages flow control to enable intelligent content delivery.

  In Mininet it is possible to test a set of open-source controllers, considered as remote controllers, since they need to be connected to the Mininet through a specific IP address. From the range of open source controllers available, in addition to the Mininet reference controller, four external controllers were tested: POX, OpenDayLight, FloodLight and Ryu. In this thesis, the POX [38] controller was used, a simple-to-use SDN controller that is bundled with *Mininet* and whose components are Python programs that implement network functions and can be invoked when POX is started. It also creates network applications that performs switching based on Ethernet MAC addresses, and prevent loops in the network. The controller implements instructions in the switches to forward traffic, acting directly in their flow tables. Figure 3.4 shows the terminal window where POX is running during the program execution.



ines_santos@ines: ~/pox

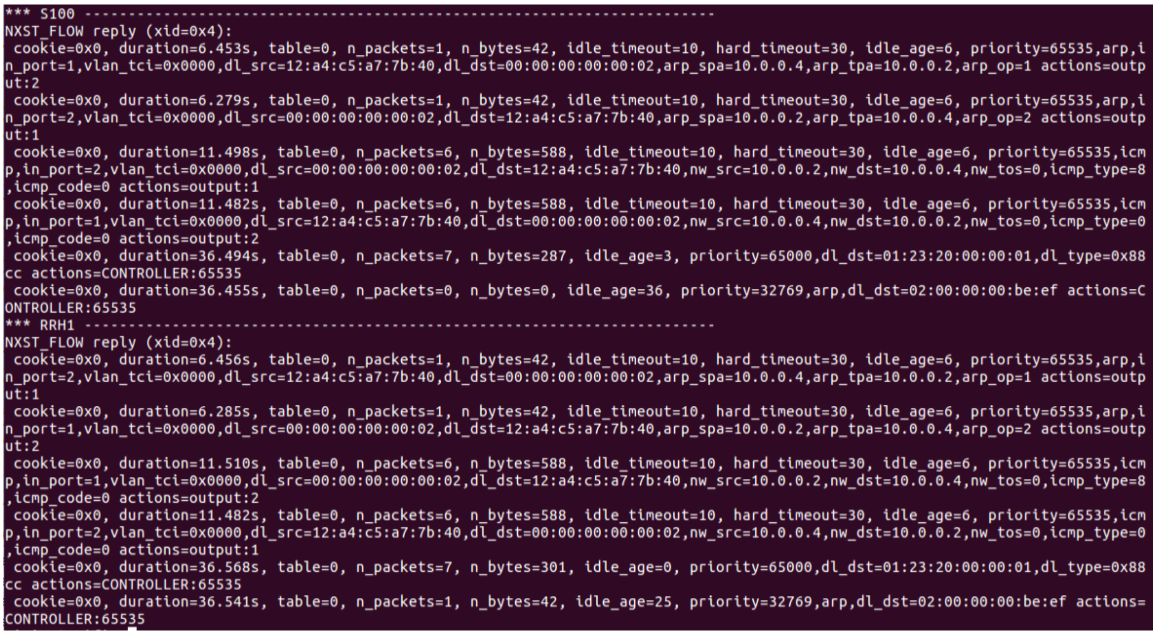
```
ines_santos@ines:~/pox$ ./pox.py forwarding.l2_learning openflow.spanning_tree -
-hold-down log.level --DEBUG samples.pretty_log openflow.discovery host_tracker
info.packet_dump
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.
[openflow.spanning_tree ] Spanning tree component ready
[host_tracker           ] host_tracker ready
[info.packet_dump       ] Packet dumper running
[core                   ] POX 0.2.0 (carp) going up...
[core                   ] Running on CPython (2.7.12/Nov 19 2016 06:48:10)
[core                   ] Platform is Linux-4.10.0-35-generic-x86_64-with-Ubuntu
-16.04-xenial
[core                   ] POX 0.2.0 (carp) is up.
[openflow.of_01         ] Listening on 0.0.0.0:6633
```

Figure 3.4: POX terminal window.

- **UEs** are the mobile users and they are connected to the RRHs through authentication (WPA_2) and association. For each UE an IP and MAC addresses are defined. In addiction to these addresses, the antenna gain and antenna height are also established.

- **RRHs** are the network elements responsible of maintaining the UEs connected to the network and assuring the communication between them and the controller, so they work as the connection point between the wireless network and the fixed wire network, and each RRH can serve multiple users within its range. When the UEs enter in other RRH's range the user starts to be served by the new one, according to the association rules, which can give priority to the signal quality or RRH load. The association rule used is defined before the program execution.

- **Open vSwitch** are the switches used in the network and are an open-source implementation of a distributed virtual multilayer switch. The main purpose of an Open vSwitch is to provide a switching stack for hardware virtualization environments, while supporting multiple protocols and standards used in computer networks. These elements are the ones responsible for introducing virtualization in the network. In this particular network, the switches work as OpenFlow switches, and it is possible to see their flow tables at any time during the program execution. If there are no packets exchanged, the flow tables are empty, otherwise each table presents the flow entries corresponding to the rules to be applied to each packet type that passes through the switch. This process can be seen with *dpctl dump-flows* command and it is represented in Figure 3.5. This snapshot was taken after a ping execution between the host and an UE, and represent the **S100** and **RRH1** flow tables, which are the network elements from the path within the host and the UE.



```
*** S100 -----------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=6.453s, table=0, n_packets=1, n_bytes=42, idle_timeout=10, hard_timeout=30, idle_age=6, priority=65535,arp,i
n_port=1,vlan_tci=0x0000,dl_src=12:a4:c5:a7:7b:40,dl_dst=00:00:00:00:00:02,arp_spa=10.0.0.4,arp_tpa=10.0.0.2,arp_op=1 actions=outp
ut:2
 cookie=0x0, duration=6.279s, table=0, n_packets=1, n_bytes=42, idle_timeout=10, hard_timeout=30, idle_age=6, priority=65535,arp,i
n_port=2,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=12:a4:c5:a7:7b:40,arp_spa=10.0.0.2,arp_tpa=10.0.0.4,arp_op=2 actions=outp
ut:1
 cookie=0x0, duration=11.498s, table=0, n_packets=6, n_bytes=588, idle_timeout=10, hard_timeout=30, idle_age=6, priority=65535,icm
p,in_port=2,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=12:a4:c5:a7:7b:40,nw_src=10.0.0.2,nw_dst=10.0.0.4,nw_tos=0,icmp_type=8
,icmp_code=0 actions=output:1
 cookie=0x0, duration=11.482s, table=0, n_packets=6, n_bytes=588, idle_timeout=10, hard_timeout=30, idle_age=6, priority=65535,icm
p,in_port=1,vlan_tci=0x0000,dl_src=12:a4:c5:a7:7b:40,dl_dst=00:00:00:00:00:02,nw_src=10.0.0.4,nw_dst=10.0.0.2,nw_tos=0,icmp_type=0
,icmp_code=0 actions=output:2
 cookie=0x0, duration=36.494s, table=0, n_packets=7, n_bytes=287, idle_age=3, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88
cc actions=CONTROLLER:65535
 cookie=0x0, duration=36.455s, table=0, n_packets=0, n_bytes=0, idle_age=36, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=C
ONTROLLER:65535
*** RRH1 -----------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=6.456s, table=0, n_packets=1, n_bytes=42, idle_timeout=10, hard_timeout=30, idle_age=6, priority=65535,arp,i
n_port=2,vlan_tci=0x0000,dl_src=12:a4:c5:a7:7b:40,dl_dst=00:00:00:00:00:02,arp_spa=10.0.0.4,arp_tpa=10.0.0.2,arp_op=1 actions=outp
ut:1
 cookie=0x0, duration=6.285s, table=0, n_packets=1, n_bytes=42, idle_timeout=10, hard_timeout=30, idle_age=6, priority=65535,arp,i
n_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=12:a4:c5:a7:7b:40,arp_spa=10.0.0.2,arp_tpa=10.0.0.4,arp_op=2 actions=outp
ut:2
 cookie=0x0, duration=11.510s, table=0, n_packets=6, n_bytes=588, idle_timeout=10, hard_timeout=30, idle_age=6, priority=65535,icm
p,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=12:a4:c5:a7:7b:40,nw_src=10.0.0.2,nw_dst=10.0.0.4,nw_tos=0,icmp_type=8
,icmp_code=0 actions=output:2
 cookie=0x0, duration=11.482s, table=0, n_packets=6, n_bytes=588, idle_timeout=10, hard_timeout=30, idle_age=6, priority=65535,icm
p,in_port=2,vlan_tci=0x0000,dl_src=12:a4:c5:a7:7b:40,dl_dst=00:00:00:00:00:02,nw_src=10.0.0.4,nw_dst=10.0.0.2,nw_tos=0,icmp_type=0
,icmp_code=0 actions=output:1
 cookie=0x0, duration=36.568s, table=0, n_packets=7, n_bytes=301, idle_age=0, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88
cc actions=CONTROLLER:65535
 cookie=0x0, duration=36.541s, table=0, n_packets=1, n_bytes=42, idle_age=25, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=
CONTROLLER:65535
```

Figure 3.5: Network elements' flow tables in Mininet-WiFi.

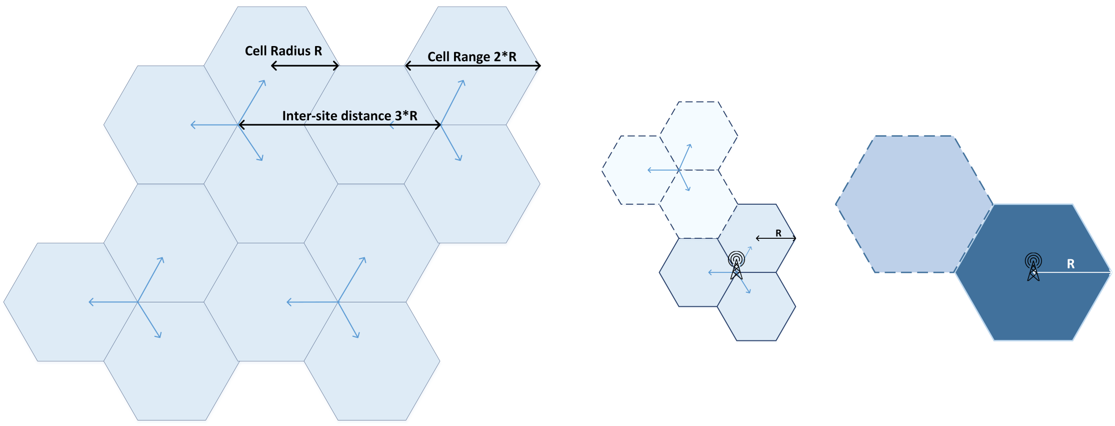
3.2 LTE modules implementation

In the wireless network, between the mobile user and the serving base station antenna, it was necessary to implement some modules in order to make the simulations be coherent with LTE. To do so, classes in Python were added, or modified, in the scripts that constitute the software, to support the addition of the LTE devices and their features.

3.2.1 LTE RF scenario

In order to test the SDN potential in LTE networks, a radio access network scenario from 3rd Generation Partnership Project (3GPP) [39] for LTE was used as reference. According to the reference scenario, the base stations with three sectors per site are placed in a hexagonal grid distanced of 3×R, where R represents the cell radius. The hexagonal grid is presented in Figure 3.6(a). The BS antenna radiation pattern used for each sector in 3-sector cell sites depends on the horizontal angle. However, the original BS antennas from the emulator are omnidirectional, radiating in all directions and not taking into account the horizontal angle between the UE and the BS antenna. So in order to use the implemented code to simulate sector antennas, it was considered the direction of maximum gain of the radiation pattern, and the transmission antenna gain value it was defined according to the same reference [39].

According to Figure 3.6(b), the cell radius used corresponds to the cell range of the reference scenario since omnidirectional antennas are used. The implemented antennas are considered with a gain equal to the maximum gain used for the sectorized antennas' planning. The interference was neglected and calculated manually. For the RAN implementation, and further network tests, the parameters presented in Table 3.1 were used, for both urban and rural scenarios.



(a) Cells layout in reference scenario (extracted from [39]).

(b) Reference scenario adaptation.

Figure 3.6: Cells layout.

Table 3.1: RAN parameters from [39].

Environment	Urban area	Rural area
Cell range	500 m	2000 m
Carrier frequency	2000 MHz	900 MHz
System bandwidth	10 MHz	10 MHz
BS antenna		
Antenna height	30 m	45 m
Antenna gain	15 dBi	15 dBi
Maximum BS power	46 dBm	46 dBm
User equipment		
Antenna height	1.8 m	
Antenna gain	0 dBi	
Maximum UE power	24 dBm	
UE noise figure	9 dB	

Beyond the radiation pattern, the transmission antennas in WiFi and LTE are different, so in order to use antennas as reliable as possible, a class was created in the emulator, in the script where the network devices were implemented, corresponding to the emulation of the LTE transmission antennas characteristics, the eNodeBs, with the features presented in Table 3.1.

Propagation Loss Models

The primary adopted LTE component was the implementation of the macro cell propagation model. For urban areas, the model is valid for the Non Line of Sight (NLOS) case only and describing the worst propagation case. This model is based on the 3GPP reference [39] considering a carrier frequency of 2000 MHz and a base station antenna height of 15 metres above average rooftop level. The propagation model is given by the following formula:

$$\mathrm{L_{urban[dB]}} = 128.1 + 37.6 \times \log_{10}(\mathrm{R_{[km]}}), \tag{3.1}$$

where R is the base station - user equipment distance.

In rural areas the Hata model from the same reference was used and, as for the urban case, is valid for the Non Line of Sight (NLOS) case only and describing the worst propagation case. Considering a carrier frequency of 900 MHz and a base station antenna height of 45 meters above ground the propagation model is given by:

$$\mathrm{L_{rural[dB]}} = 95.5 + 34.1 \times \log_{10}(\mathrm{R_{[km]}}), \tag{3.2}$$

where R is the base station - user equipment distance.

The UE received power, which is useful to estimate the SNR, using the above path loss formulas for each environment, can be calculated by:

$$P_{r[dB]} = P_{t[dBm]} + G_{t[dBi]} + G_{r[dBi]} - L_{p[dB]}, \tag{3.3}$$

where:

$\mathbf{P_{t[dBm]}}$: transmitter output power;

$\mathbf{G_{t[dBi]}}$: gain of the transmitting antenna;

$\mathbf{G_{r[dBi]}}$: gain of the receiving antenna;

$\mathbf{L_{p[dB]}}$: path loss.

The equation (3.3) translates the received power values presented in the graph of Figure 3.7, for urban and rural areas. In the same figure, the Pr is plotted for each distance between the UE and the RRH, and according to the parameters presented in Table 3.1. The values of the received power (Pr) were obtained in the emulator and plotted in Matlab®.

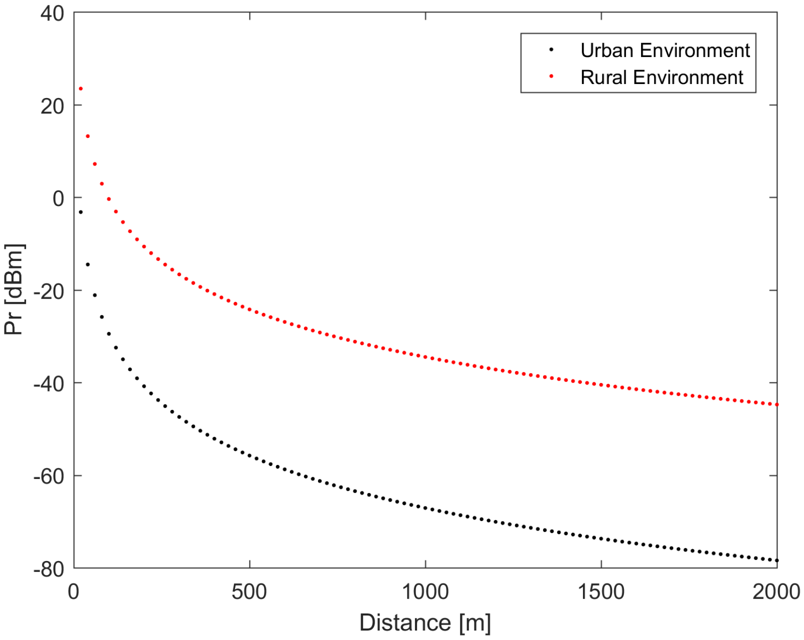


Figure 3.7: Received power according to distance.

3.2.2 Modulation and coding scheme

The Modulation and Coding Scheme (MCS) was used to determine the data rate of a connection, and even though WiFi and LTE present spectral efficiency depending on the modulation coding scheme, they give different values of spectral efficiency for each MCS. In WiFi, spectral efficiency depends on the technology (e.g., 802.11a, 802.11b, 802.11n, etc.), and because of that, it was necessary to implement the LTE coding scheme index distribution.

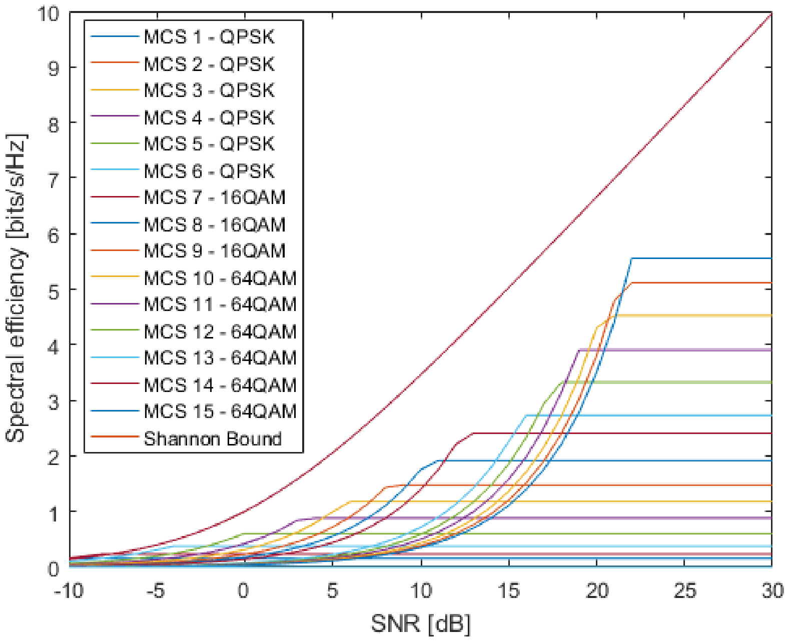


Figure 3.8: MCS Theoretical curves.

Using the LTE Channel Quality Indicator (CQI) Table presented in Appendix A, from 3GPP [40], along with Equation (3.4) from reference [41], the spectral efficiency was plotted according to the SNR. The MCS curves were obtained in Matlab® and are presented in the graph of Figure 3.8. The following equation was used for plotting:

$$\frac{R_b}{B} = \min\left(\frac{3\,\log_2 M}{2(M\text{-}1)(\mathrm{erfc}^{-1}(\frac{P_{b,target}\log_2 M}{2}))^2}\frac{n}{k}G_{MCS}\left(P_{b,target}\right)\mathrm{snr},\, \frac{k}{n}\,\log_2 M\right), \tag{3.4}$$

where M is the number of bits per modulation symbol (e.g., QPSK (M=4), 16-QAM (M = 16) and 64-QAM (M = 64)) , G_{MCS} is the coding gain , $\frac{k}{n}$ is the code rate and snr is the signal to noise ratio, in linear units. The Bit Error Rate (BER) target, $P_{b,\ target}$, is 10^{-3}, since, using SISO, the system does not guarantee a BER smaller than the 10^{-3} BER target for SNR less than -10 dB as described in [41].

The MCS curves give the maximum rate supported in each modulation,for each SNR interval. This interval can be taken from the graph of Figure 3.8 and was implemented in the emulator.

Moreover, since it is not possible to access the SNR values in the emulator, Equation (3.5) was used. It relates the SNR with a parameter, the Reference Signal Received Power (RSRP), capable of being converted into the Received Signal Strength Indicator (RSSI) measured at the receiving antenna:

$$\mathrm{SNR} = \frac{\mathrm{RSRP}}{N_{floor}}, \tag{3.5}$$

where the RSRP is defined as the linear average of the power contributions (in [W]) of the Resource Elements (REs). In [42], RSRP is expressed as a function of the RSSI according to:

$$\mathrm{RSRP}_{[dBm]} = \mathrm{RSSI}_{[dBm]} - 10 \times \log_{10}(12 \times \mathrm{N}), \tag{3.6}$$

where N is the number of Physical Resource Blocks (PRBs), per bandwidth.

The noise component was calculated as the noise floor of the receiver according with:

$$\mathrm{N}_{\text{floor[dBm]}} = \mathrm{N}_{\text{Thermal[dBm]}} + \mathrm{NF}_{\text{[dB]}}, \tag{3.7}$$

with the receiver Noise Figure (NF) presented in Table 3.1 and the thermal noise $\mathrm{N}_{\text{Thermal}}$ comes from the expression:

$$\mathrm{N}_{\text{Thermal [W]}} = \mathrm{k} \times T \times \mathrm{B}_{\text{subcarrier}}, \tag{3.8}$$

where k is the Boltzmann's constant, $1.38 \times 10^{-23}\mathrm{JK}^{-1}$, T is the temperature in Kelvin and $(\mathrm{B}_{\text{subcarrier}})$ is the bandwidth of a single sub-carrier equal to 15 kHz.

Using the expressions above (3.5) and (3.6) it is possible to convert the SNR values in RSSI as expressed in equation (3.9).

$$\mathrm{RSSI}_{\text{[dBm]}} = \mathrm{SNR} + 10 \times \log_{10}(12 \times \mathrm{N}) - \mathrm{N}_{\text{floor[dBm]}} \tag{3.9}$$

The RSSI corresponds to the UE received power measured over the entire bandwidth, and it can be accessed in the emulator terminal through the command *UE.params['RSSI']* while the program is running.

The implemented code in the emulator, assigned to each RSSI interval, obtained from the SNR values, a maximum supported data rate. Each RSSI value corresponds to a RSRP, which represents the UE received power. The RSRP intervals and corresponding efficiency are represented in Table 3.2. The maximum rate were established in an emulator script multiplying the efficiency by the channel bandwidth.

3.2.3 RRHs placement - grid implementation

In order to place all the RRHs equidistantly, a grid algorithm was implemented in the main script of the program, taking into account the cell range and the inter-cell distance, as described in Algorithm 1.

Table 3.2: RSRP intervals defined in the emulator.

RSRP [dBm]	Efficiency [bits/s/Hz]
≥ -76	5.6
< -76 and ≥ -78	5.1
< -78 and ≥ -80	4.5
< -80 and ≥ -81	3.9
< -81 and ≥ -83	3.3
< -83 and ≥ -85	2.7
< -85 and ≥ -87	2.4
< -87 and ≥ -89	1.9
< -89 and ≥ -91	1.5
< -91 and ≥ -93	1.2
< -93 and ≥ -95	0.9
< -95 and ≥ -97	0.6
< -97 and ≥ -99	0.4
< -99 and ≥ -101	0.3
< -101 and ≥ -103	0.2
< -103	0

Algorithm 1 RRH grid code

```
$nr_{rrhs} \leftarrow 19$
for $i \leftarrow 1 : nr_{rrhs}$ do
    $rrh_i = net.addAccessPoint('rrhi', position =' x, y, h'_{tx}, antennaGain, antennaHeight)$
    if $count_x = limit_x$ then
        $count_y \leftarrow count_y + 1$
        $x \leftarrow x_{init}$
        $y \leftarrow y_{init} + (2 \times count_y) \times tan(\frac{\pi}{3}) \times \frac{intercelldist}{2}$
        $count_x \leftarrow 0$
        $dir \leftarrow 0$
    else if dir = 1 then
        $x \leftarrow x + \frac{distance}{2}$
        $y \leftarrow y + tan(\frac{\pi}{3}) \times \frac{distance}{2}$
        $dir \leftarrow 0$
    else if dir = 0 then
        $x \leftarrow x + \frac{distance}{2}$
        $y \leftarrow y - tan(\frac{\pi}{3}) \times \frac{distance}{2}$
        $dir \leftarrow 1$
        $count_x \leftarrow count_x + 1$
    end if
end for
```

Figure 3.9 shows the placement of the 19 RRHs, according to the code developed. The figure presents also the UEs, placed in a generic order, serving only as an example of a possible distribution of users in the network.

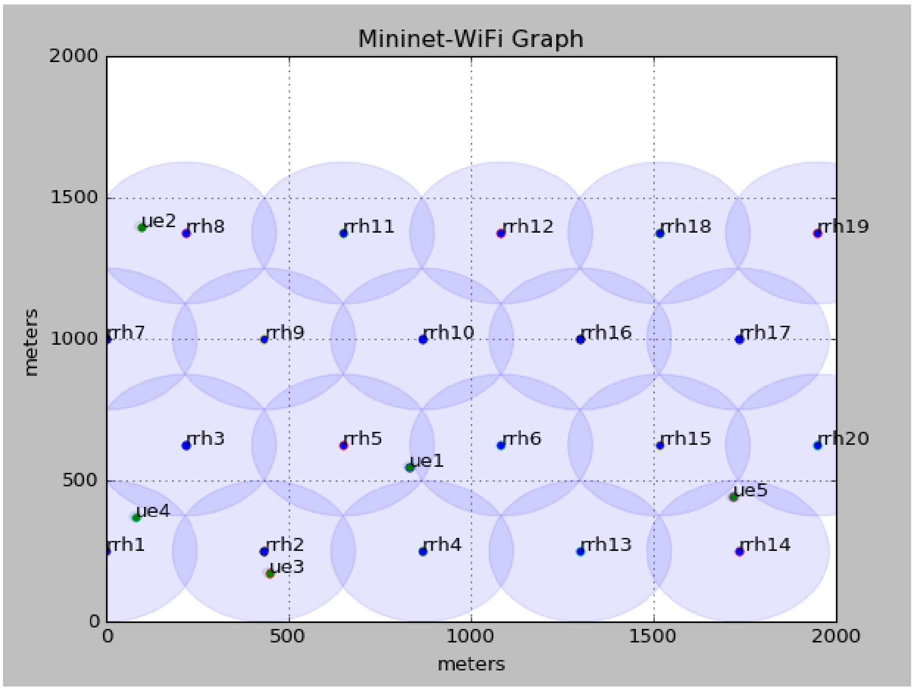


Figure 3.9: RRHs' placement.

3.3 Network Tests and Assessment

3.3.1 Assumptions

Considering the emulator' potential and limitations some assumptions were assessed:

- The BBUs, and the RRHs, are used as a representation to address the C-RAN concept for 5G networks. However, they do not execute explicit baseband, and digital, processing functions, respectively;
- The distance between RRHs and BBUs is not taken into account, since it was assumed fiber links with high capacity and all signal processing is done at the RRH level, so this possible limitation does not affect the network performance;
- The core network is used in terms of packet forwarding, and the S-GW and P-GW control functions are executed at the SDN controller level;
- The end-to-end TCP/UDP connection is assumed to be similar for WiFi and LTE, according to both technologies' protocol stack;
- The NFV component of this work is represented by the use of virtual switches.

3.3.2 Network Tests

In the simulations in Mininet-Wifi, several tests were performed in order to see, not only the SDN influence in mobile networks, but also to test the feasibility of this approach. The network performance is measured in two classic ways: the network throughput and the packet latency, or Round Trip Time (RTT).

Throughput

Iperf [43] was the tool used for throughput measurements, which runs as an application and generates traffic. It measures the throughput between two points of the network (e.g. a mobile user and a host acting as a server) through TCP packets exchange. This two points were selected in order to simulate an E2E LTE connection. In order to take some conclusions regarding the network performance, the throughput values were converted to spectral efficiency, dividing the simulated results by the bandwidth. Plotting the spectral efficiency values according to each SNR, it is possible to obtain a comparison with the Shannon bound curve.

The Shannon bound represents the capacity of an Additive White Gaussian Noise (AWGN) channel and therefore, the maximum theoretical spectral efficiency that can be achieved in LTE. It can be expressed as:

$$\mathrm{C}_{[\mathrm{bits/s/Hz}]} = \mathrm{B}_{[\mathrm{MHz}]} \times \log_2(1+\mathrm{snr}), \tag{3.10}$$

where C is the Shannon capacity, B the system bandwidth and snr the signal-to-noise ratio in linear units.

In order to the *iperf* measure the throughput between two points in one direction, it creates a data stream within a client and a server. The *iperf* output contains a time-stamped report with the amount of data transferred and the throughput measured. In this particular case the data stream is created from the host to the UE, corresponding to a downlink connection, as represented in Figure 3.10.

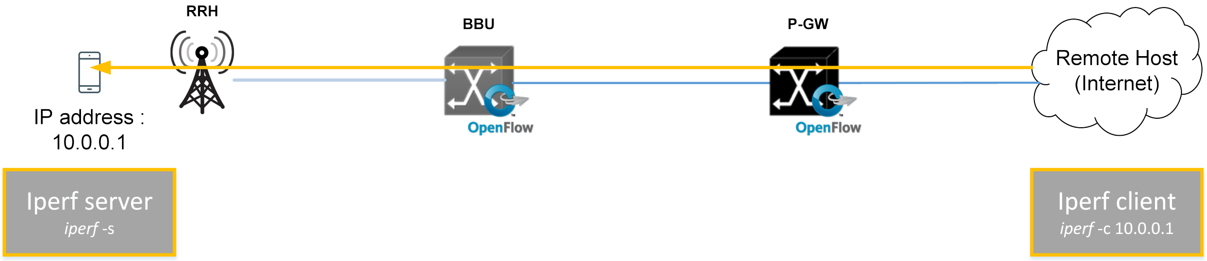


Figure 3.10: *Iperf* test.

Latency

In LTE there are two types of latency, the control plane latency and the user plane latency described in the diagram of Figure 3.11. In this thesis, the latency tests performed are only related to the user plane latency.

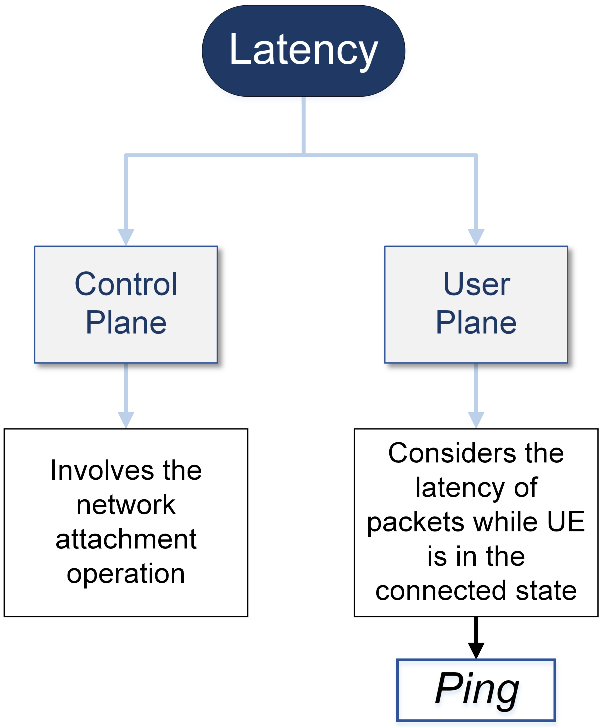

Latency
Control Plane
User Plane
Involves the network attachment operation
Considers the latency of packets while UE is in the connected state
Ping

Figure 3.11: LTE Control plane and User plane latency.

3GPP defines the U-Plane latency as: *"the one-way transit time between a packet being available at the IP layer in either the UE/RAN edge node and the availability of this packet at the IP layer in the RAN edge node/UE. The RAN edge node is the node providing the RAN interface towards the core network. This requirement, more specifically the exact definition of latency, may be revisited and further clarified once there is a 3GPP system end-to-end requirement agreed and the overall system architecture is settled, including the RAN and core network functional split"*.

3GPP defines the user plane latency in the radio part of the network according to the procedures presented in Table 3.3, for two HARQ scenarios.

Table 3.3: U-Plane radio network latency analysis (estimated average) [44].

Step	Description	Value (0% HARQ)	Value (30% HARQ)
0	UE wakeup time	Implementation dependent - Not included	Implementation dependent - Not included
1	UE processing delay	1 ms	1 ms
2	Frame Alignment	0.5 ms	0.5 ms
3	TTI for UL data packet	1 ms	1 ms
4	HARQ retransmission	0 ms	0.3 * 5 ms
5	eNB processing delay	1 ms	1 ms
	Total one way delay	**3.5 ms**	**5 ms**

In [11], the edge node is defined as the aGW and thus, the S1-U delay is specified separately in the U-Plane latency assessment. The assessment also assumes, in accordance with the requirement, unload conditions where scheduling delays and S1 transmission times are negligible.

So the LTE U-Plane delay is summarized in: node processing delays, TTI duration, radio frame alignment and S1-U delay. Figure 3.12 presents those components which are also described in Table 3.4.

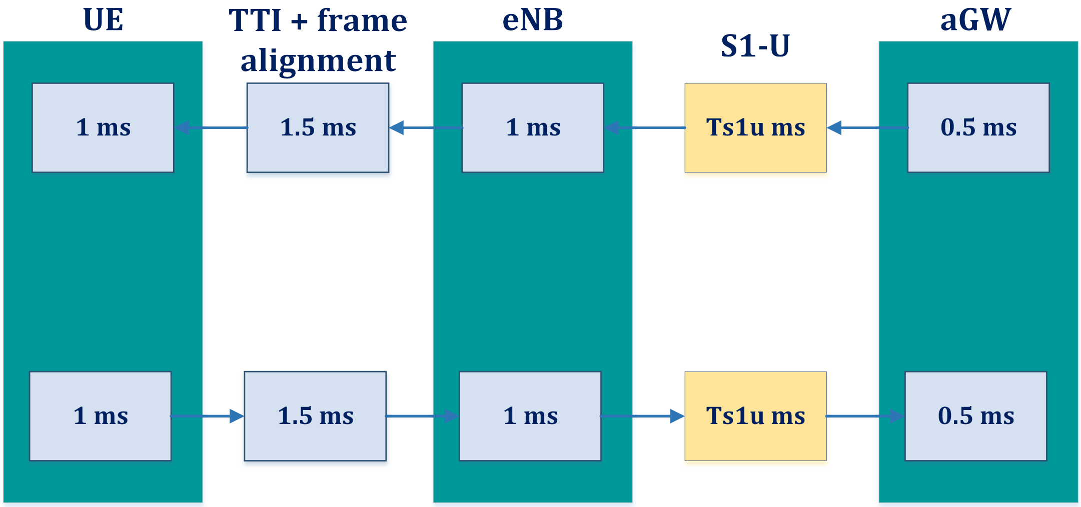


Figure 3.12: U-Plane end-to-end latency components (adapted from [11]).

Table 3.4: U-plane latency analysis (estimated average) [44], [11].

Step	Description	Value (30 % HARQ)
0	UE wakeup time	Implementation dependent - Not included
1	UE Processing delay	1 ms
2	Frame Allignment	0.5 ms
3	TTI for DL Data packet	1 ms
4	HARQ retransmission (30%)	0.3 * 5 ms
5	eNB Processing delay (Uu -> S1-U)	1 ms
6	S1-U transfer delay	Ts1u (1 ms - 15 ms)
7	aGW Processing Delay	0.5 ms
	Total one way delay	**5 ms + Ts1u**

More practically, in the U-Plane, the total delay of a packet transmission in a cellular network has contributions from the RAN, backhaul, core network, and data center/Internet [29]. The total one way transmission time of a current LTE system can be expressed as:

$$\mathrm{T_{OWD}} = \mathrm{T_{Radio}} + \mathrm{T_{Backhaul}} + \mathrm{T_{Core}} + \mathrm{T_{Transport}}, \tag{3.11}$$

where

$\mathbf{T_{Radio}}$ is the packet transmission time between the eNB and the UE and is mainly due to physical layer communications;

$\mathbf{T_{Backhaul}}$ is the time for building connections between the eNB and the core network (the EPC);

$\mathbf{T_{Core}}$ is the processing time taken by the core network;

$\mathbf{T_{Transport}}$ is the delay between the core network and the Internet/cloud.

Some network applications does not require high data rates, but rather very low latency. Latency is usually defined by RTT which impacts the Quality of Experience (QoE). The RTT can be given by:

$$\mathrm{RTT} = 2 \times \mathrm{T_{OWD}} \tag{3.12}$$

where $\mathrm{T_{RTT}}$ is the RTT and $\mathrm{T_{OWD}}$ is the One Way Delay.

The RTT measurements performed in the emulator are performed in two different ways:

1. In an E2E connection.

   The measurements to find the E2E latency of LTE networks for data applications are based on the *ping* Round Trip Time (RTT). The RTT measures the time between sending and receiving Internet Control Message Protocol (ICMP) messages (*ping* messages), between the terminal and the destination as shown in Figure 3.13. In this case, the test was executed between a mobile terminal and an IP host, which is configured to respond to ICMP Echo Requests.

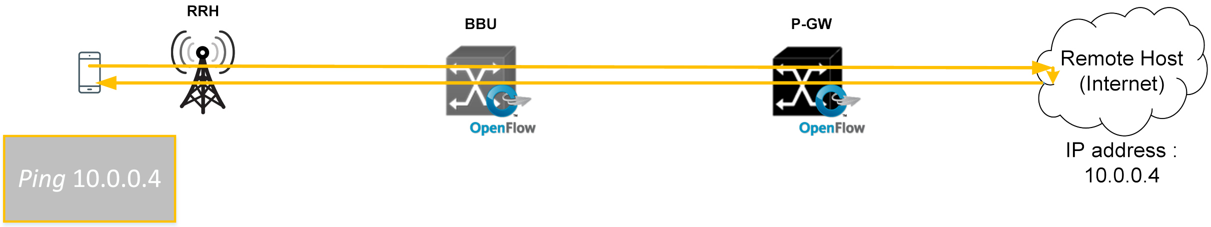


Figure 3.13: E2E *Ping* test.

2. In a wired connection.

   The *ping* test is executed in the same way as the previous test, but this time between the IP host and the base station antenna. This connection is represented in Figure 3.14

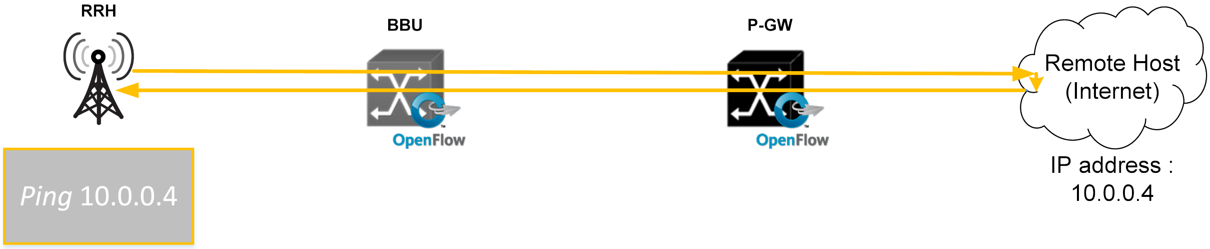


Figure 3.14: Wired *Ping* test.

3.3.3 Hypotheses Assessment

To assess the developed hypotheses, three metrics were considered: the RMSE, the R^2 and the Pearson correlation.

The RMSE measures the square root of the average of the square of the differences between the predicted and the original values. Expression (3.13) represents the RMSE, where y_i are the original and $\hat{y}_i$ the predicted values of the i^{th} set of parameters.

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{3.13}$$

The coefficient of determination, R^2, is given by:

$$\text{R}^2 = 1 - \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y}_i)^2} \tag{3.14}$$

where the numerator corresponds to the sum of squared errors of the regression model and the denominator is the sum of squared errors of the baseline model.

The Pearson Correlation measures the linear correlation between the predicted values and the original ones. This metric is given by (3.15), where y_i and $\hat{y}_i$ are the original and predicted values, respectively, of the i^{th} set of parameters. The $\bar{y}$ and $\bar{\hat{y}}$ represent the mean values of these metrics.

$$\text{R}_{\text{Pearson}} = \frac{\sum_{i=1}^{m}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{m}(\hat{y}_i - \bar{\hat{y}})^2}\sqrt{\sum_{i=1}^{m}(y_i - \bar{y})^2}} \tag{3.15}$$

Chapter 4

Result Analysis

This chapter presents the results of the performed tests in order to evaluate the LTE SDN based network performance. These results are then analyzed and compared to field measurements.

4.1 Throughput

The first performed test aimed to understand the network behaviour considers the throughput. The UE was placed manually in a position, corresponding to a certain SNR value, and a throughput measurement was done during 60 seconds, from which, minimum, maximum and mean throughput values were obtained. The same process was applied in several terminal positions, equally spaced, in order to scan the throughput within the SNR interval [-5;30] dB.

Urban environment

The graph of Figure 4.1 was obtained from the average throughput measurements at each position, divided by the corresponding channel bandwidth (10 MHz), resulting in the spectral efficiency.

In order to get a spectral efficiency curve (by fitting the experimental data), the following function, representing an attenuated and truncated form of the Shannon bound, was used:

$$\frac{\mathrm{R}_b}{\mathrm{B}}(\eta_{\mathrm{BW,r}}, \eta_{\mathrm{SNR,r}})_{[\mathrm{bits/s/Hz}]} = \eta_{\mathrm{BW,r}} \times \log_2(1 + \eta_{\mathrm{SNR,r}} \times \mathrm{snr}), \tag{4.1}$$

where snr represents the signal-to-noise ratio in linear units and the coefficients $\eta_{\mathrm{BW,r}}, \eta_{\mathrm{SNR,r}}$ are related to the bandwidth and snr efficiencies respectively.

According to (4.1) and using Matlab® to approximate the data points to a fitting curve capable of being compared with the Shannon bound equation, the coefficients presented in Table 4.1 were obtained.

The curve fitting resulted in 0.0783 dB of RMSE and 99.95% of correlation values, respectively, for the SNR range between -4 and 20 dB, which are solid results. For SNR values greater than 20 dB, the spectral efficiency seems to stabilize in the 5.5 Bits/s/Hz, which represents a value very close to the maximum rate for the highest supported modulation.

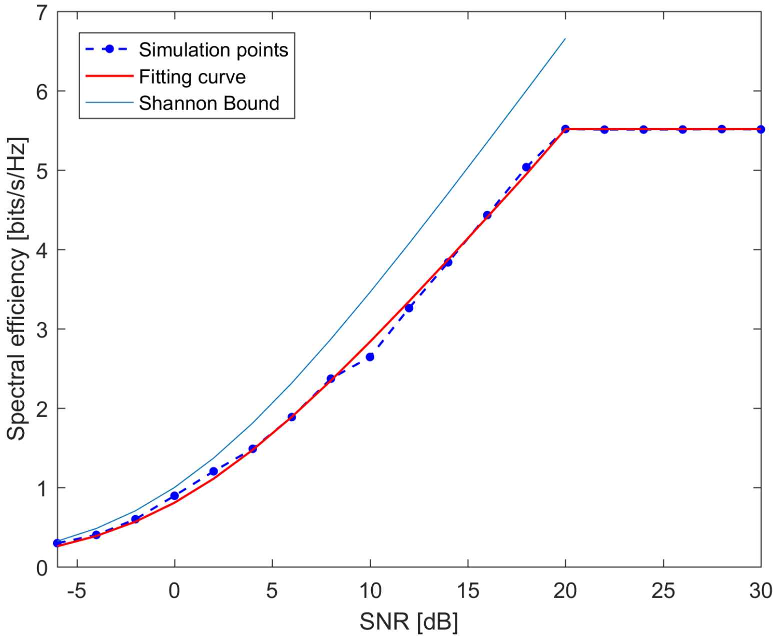


Figure 4.1: Spectral efficiency obtained for the urban area in the emulator.

Table 4.1: Throughput fitting results for an urban environment.

SNR	$\eta_{\text{BW,r}}$	$\eta_{\text{SNR,r}}$	RMSE	R^2	Correlation
[-4, 20] dB	0.83234	0.96005	0.0783	0.998	0.9991

In Figure 4.1 the Shannon bound is represented in light blue; since the LTE bandwidth efficiency is reduced by several factors, presented in Table 4.2, this capacity limit is never reached.

Table 4.2: Bandwidth efficiency for 10 MHz LTE system downlink [45].

Impairment	Link $\eta_{BW,r}$	System $\eta_{BW,r}$
BW efficiency	0.9	0.9
Cyclic Prefix	0.93	0.93
Pilot overhead	1	0.94
Dedicated and common control channels	N.a.	0.715
Total	**0.83**	**0.57**

Firstly, due to the Adjacent Channel Leakage Ratio (ACLR) requirements and practical filter implementation, the bandwidth occupancy is reduced to 90%. The overhead of the cyclic prefix is approximately 7% and the pilot assisted channel overhead estimation is approximately 6% for a single antenna transmission. The pilot overhead it is not included in the link performance bandwidth efficiency because it is used an ideal channel estimation. In [45], the authors simulated an AWGN channel with 10 MHz bandwidth in a Typical Urban (TU) environment and an overall link bandwidth efficiency of 83 % was found, where the $\eta_{SNR,r}$ was obtained by curve fitting. The same reference also presents the values which provides the best fit to the link adaptation curve, corresponding to a bandwidth and SNR efficiencies of 0.75 and 0.8 respectively.

The results in [45] are based on the antenna scheme listed in Table 4.3, for Typical Urban (TU) and an AWGN channel.

Table 4.3: Antenna scheme in [45].

Parameter	**Value**
Carrier Frequency	2 GHz
System Bandwidth	10 MHz
OFDM parameters	According to 3GPP TR 25.814, short cyclic prefix, 7 data symbols per sub-frame, 1 ms TTI
Channel estimation	Ideal
Antenna scheme	SISO

The two modified Shannon curves resulting from the bandwidth and SNR efficiencies compared in [45] are represented in Figure 4.2 along with the curve obtained in the emulator and the Shannon capacity bound.

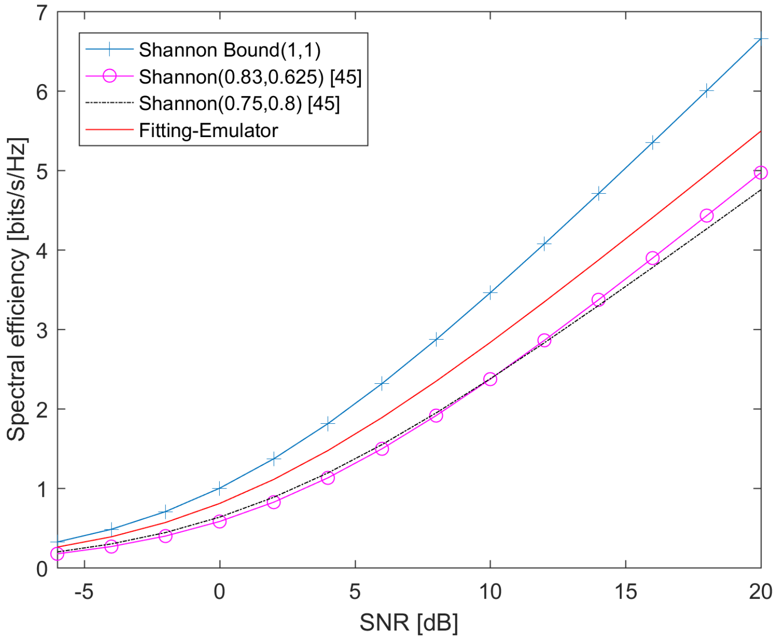


Figure 4.2: Throughput fitting curves comparison between emulator's fit and Shannon equations from [45].

In order to get some conclusions regarding the network performance, the graph of Figure 4.3 allows to evaluate the difference between the emulator values and the ones corresponding to the two adjusted Shannon curves. As one can see, the emulator constantly displays higher values of spectral efficiency, since the graph has no negative values. The maximum difference achieved for the curve with 83 % bandwidth efficiency was 0.46 dB and 0.94 dB regarding the curve which has 75 % bandwidth efficiency.

For the bandwidth considered (10 MHz), that difference corresponds to a difference in throughput of 4.6 Mbit/s and 9.4 Mbit/s (8.2 % and 16.8 %, respectively).

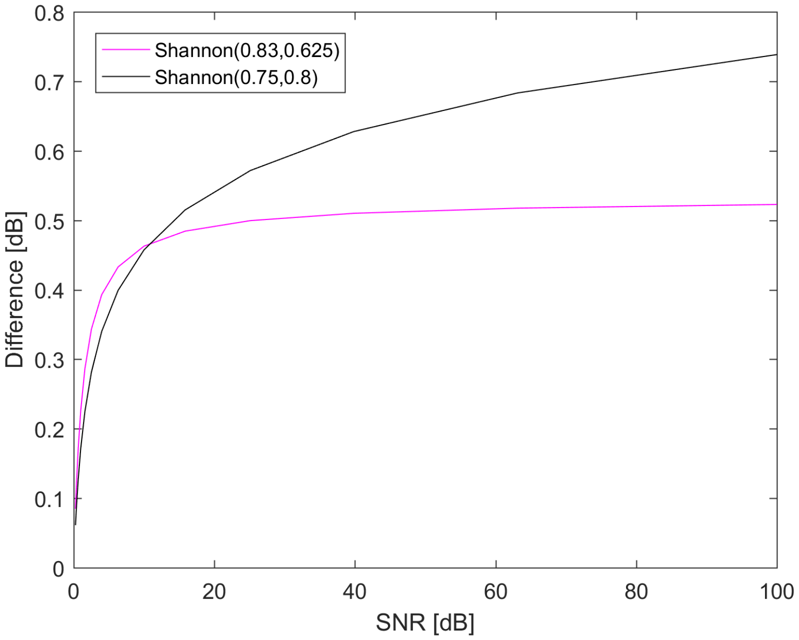


Figure 4.3: Difference between emulator's fit and Shannon equations from [45].

Rural Environment

The spectral efficiency values presented in the graph of Figure 4.4 were obtained from the average throughput measurements, as for the urban environment, but for the rural scenario described in Section 3.2.

According to (4.1) and using Matlab®, the spectral efficiency fit was obtained, and the coefficients were obtained for the SNR interval considered, and are presented in Table 4.4.

Table 4.4: Throughput fitting results for a rural environment.

SNR	$\eta_{BW,r}$	$\eta_{SNR,r}$	RMSE	R^2	Correlation
[-4, 20] dB	0.85722	0.4532	0.0682	0.998	0.9992

The throughput measurements in this environment does not stabilize for SNR values greater than 20 dB. Instead, for the highest modulation, which imposes a maximum spectral efficiency of 5.6 bits/s/Hz, the throughput, and consequently the spectral efficiency, increases linearly and can be approximated to a linear regression:

$$\frac{\mathrm{R}_b}{\mathrm{B}}\,[\text{bits/s/Hz}] = \mathrm{m} \times \mathrm{SNR} + \mathrm{b}, \tag{4.2}$$

where m is the function slope, and b is the y-axis interception.

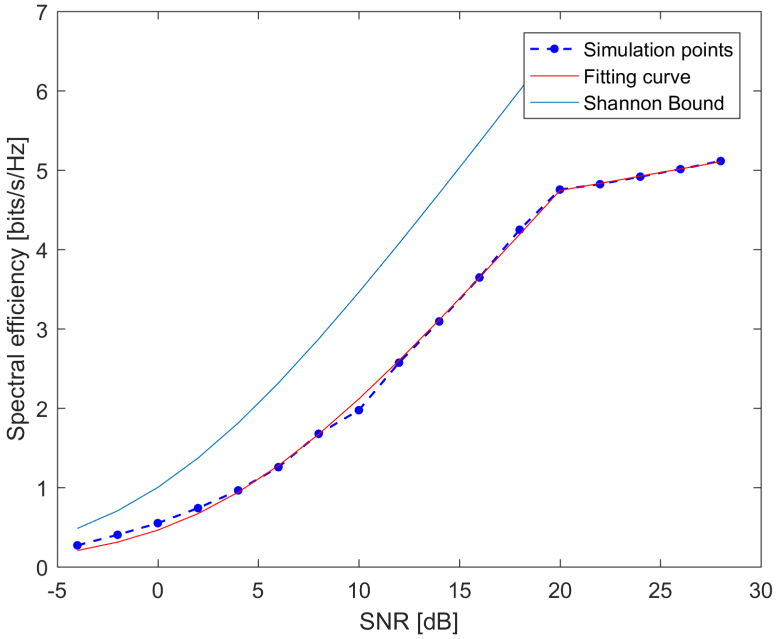


Figure 4.4: Spectral efficiency obtained for the rural area in the emulator.

The parameters obtained for the regression fitting are presented in Table 4.5, and the respective regression in Figure 4.4.

Table 4.5: Regression fitting for higher values of SNR in Rural areas.

SNR	m	b	R^2
[20, 30] dB	0.0454	3.8334	0.9973

Throughput comparison

Figure 4.5 presents the two throughput curves obtained for each environment in Mininet-Wifi.

In [7], the key radio access requirements set by International Telecommunication Union Radiocommunication Sector (ITU-R), for International Mobile Telecommunications (IMT) Advanced in different scenarios are defined. The spectral efficiency requirements for the urban and rural scenarios tested in this thesis, are presented in Table 4.6 according to [7].

Table 4.6: Average spectral efficiency comparison, between the emulator and the ITU requirement.

	Environment	Emulator	ITU requirement [7]
Average spectral efficiency [bits/s/Hz]	Urban Macro	**3.4**	**2.2**
	Rural Macro	**2.84**	**1.1**

The bar plot of Figure 4.6 was obtained through the Table 4.6 values. It can be concluded that, for this approach, using a SDN emulator and a LTE radio network, the spectral efficiency values are better than the ones imposed by international entities. For the urban scenario, the proposed solution presents

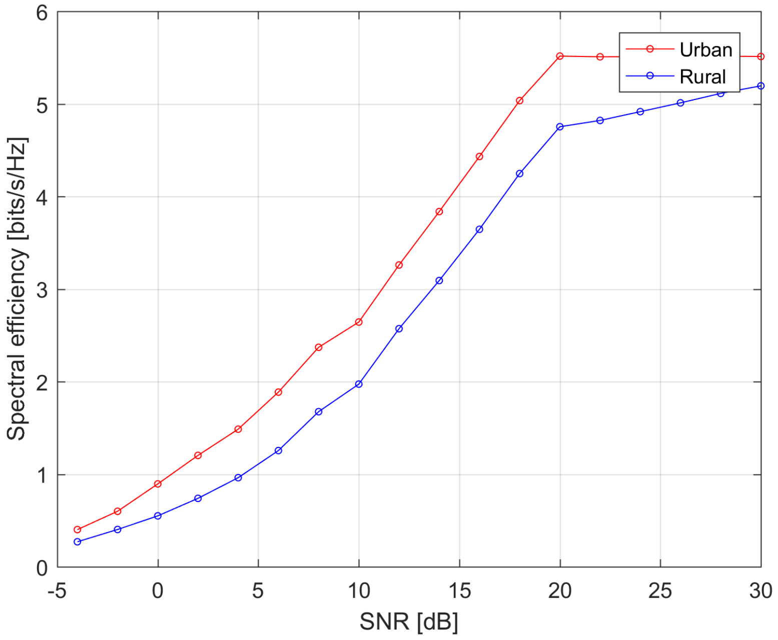


Figure 4.5: Spectral efficiency curves for urban and rural environments obtained in the emulator.

an increase of 54 % regarding the ITU requirement, and for the rural environment an increase of 158 % was achieved.

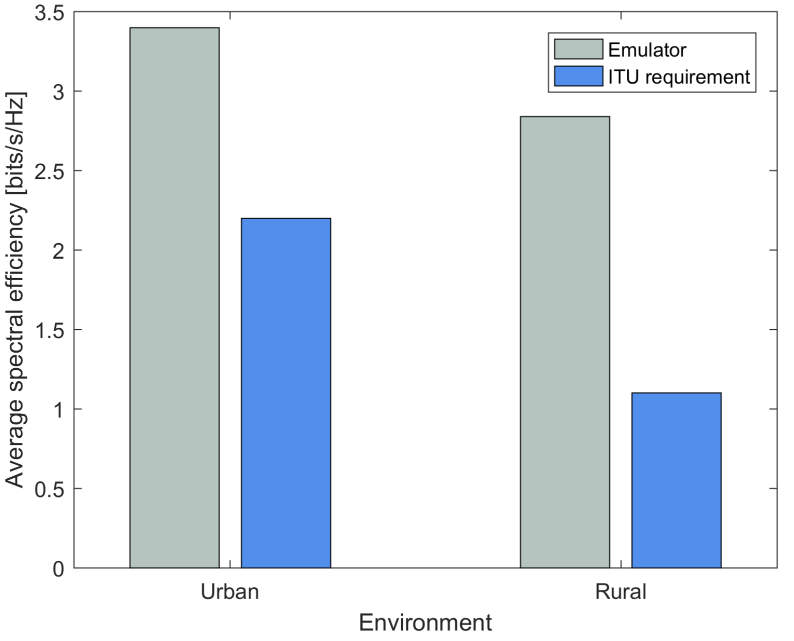


Figure 4.6: Bar plot representation of the average spectral efficiencies.

4.2 Latency

The second test addressed the E2E latency based on the *ping* test, which measures the RTT. As performed in the throughput test, a fitting curve was calculated in order to analyze how latency varies with SNR. The tests were performed for three packet sizes (32, 1000 and 1500 bytes), to see how this parameter affects the network performance. The same procedure was done for two difference environments : Urban and Rural.

Urban Environment

The latency was measured from the cell edge to the center in equal spaced time intervals, corresponding to the SNR values mapped in Figure 4.7. In the graphs of the aforementioned figure, the outliers were removed. These outliers happened periodically, corresponding to the OpenFlow hard timeout, starting in the first exchanged packet. The hard timeout is presented in the Openflow flow entries and, as described in Chapter 2, it represents the maximum amount of time a flow stays in the switch. After that time, the flow expires and a new rule needs to be sent from the controller to re-install the entry in the switch.

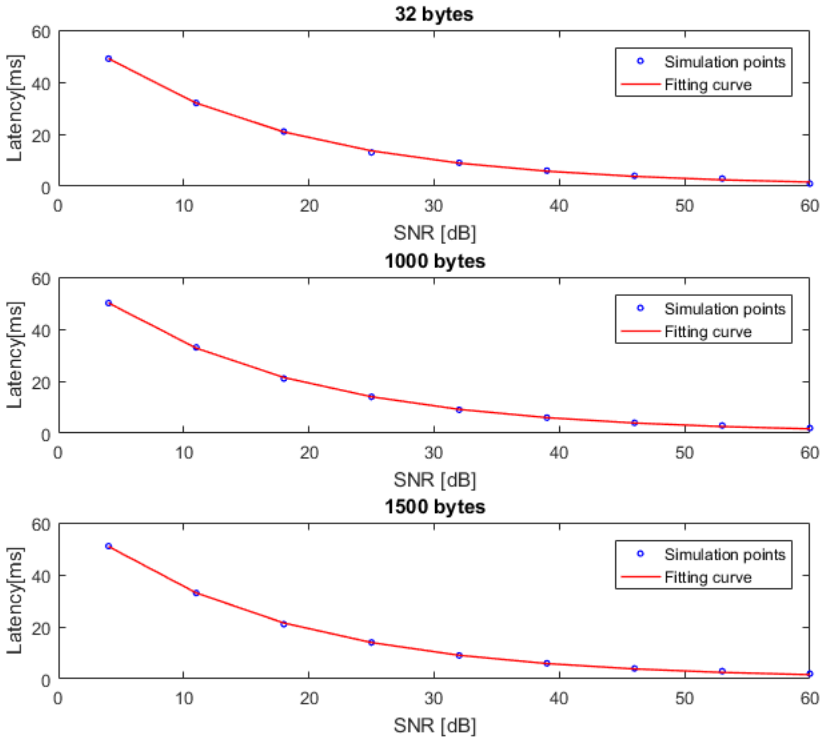


Figure 4.7: Latency fitting curves according to packet size for the urban area.

Looking at Figure 4.8, representing the three fitting curves in the same plot, it is possible to conclude that for packets of 32, 1000 and 1500 bytes, the latency variation is quite small and when the SNR is very high, the curves tend to the same maximum value. The small variation, for these package sizes, can be explained by the large difference between the channel bandwidth (and data rate) and the bytes being transmitted, causing the small increase of latency for 1500 bytes. For smaller values of SNR, since the modulation presents a smaller data rate, a large packet will take more time to be transmitted, than the smaller ones, being verified a difference of, approximately, 5 ms for the same SNR value. On the other hand, for larger data rates, the packet size will not affect the network latency, since the network throughput, the bits that can be transmitted per second presented in the Figure 4.1, is much higher than the packet size. The latency increases with decreasing SNR, since smaller SNR represents worse propagation conditions, more errors and consequently higher RTT for the packet to reach the destination and back.

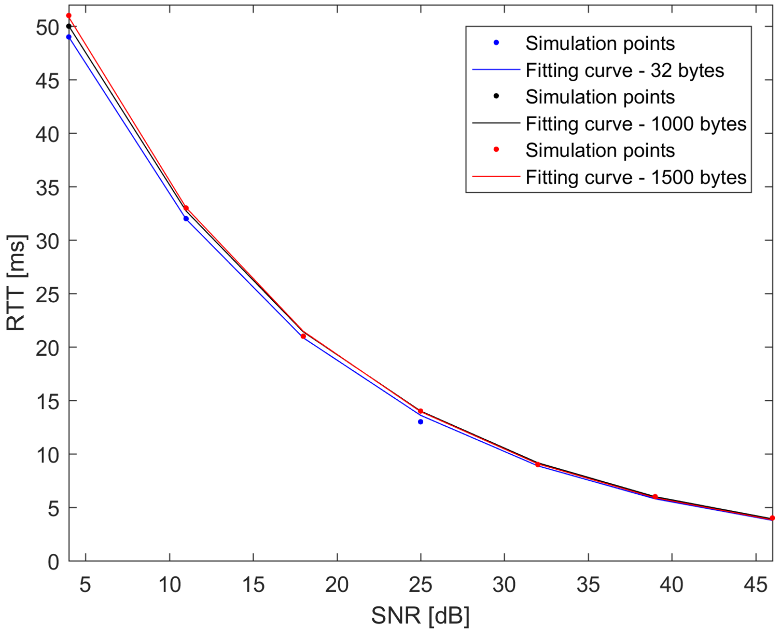


Figure 4.8: Latency fitting curves comparison for the urban area.

The fitting curves for latency, or RTT, presented in the graphs of Figures 4.7 and 4.8, dependent on SNR, resulted from equations of the form:

$$\mathrm{RTT}_{[\mathrm{ms}]} = \mathrm{b}_1\ \mathrm{e}^{\mathrm{b}_2 \times \mathrm{SNR}}, \tag{4.3}$$

where SNR represents the signal-to-noise ration in dB in each position. The coefficients b_1 and b_2 depends on the packet size, since it is the only different condition from one curve to the other. Their values are presented in Table 4.7.

Table 4.7: Latency fitting coefficients for Urban areas

Packet size (bytes)	b_1	b_2	RMSE	R^2	Correlation
32	54.232	-0.05751	0.445	0.999	0.9994
1000	59.576	-0.059027	0.474	0.999	0.9995
1500	58.854	-0.055837	0.694	0.997	0.9989

Rural Environment

The latency test for the rural scenario was performed in a similar way to what was done for the urban scenario. The measurements were performed from the cell edge to the center in equal spaced time intervals, corresponding to a specific SNR value, and are represented in Figure 4.9. In this case, looking to the graph of Figure 4.10, where the three fitting curves are plotted, it is possible to verify the similarity in the three packet sizes' curves. This similar behaviour can also be verified in the coefficients of the fitting curve, Table 4.8, which present very close values. So it can be concluded that for this environment, packet sizes until 1500 bytes, do not affect the latency performance. On the other hand, in this case, the measurements present very high values of latency, reaching the 200 ms. This may happen due to the large cell range characteristic of this type of environment, which affects the propagation delay.

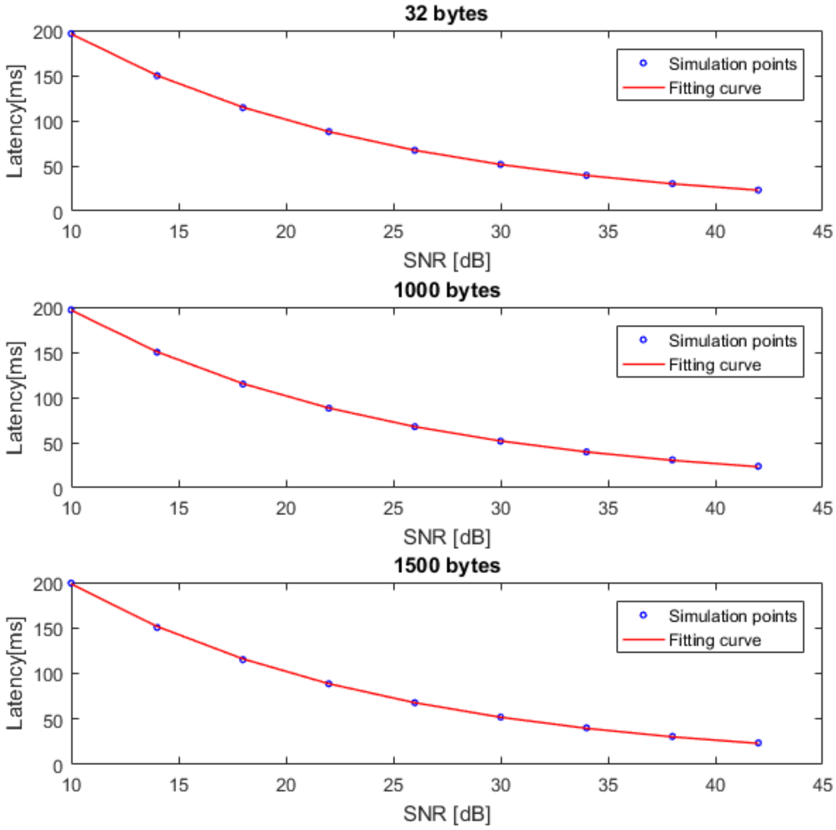


Figure 4.9: Latency fitting curves according to packet size for the rural area.

Table 4.8: Latency fitting coefficients for rural areas.

Packet size (bytes)	b_1	b_2	RMSE	R^2	Correlation
32	382.57	-0.066865	0.261	1	1
1000	382.6	-0.066656	0.415	1	1
1500	387.21	-0.066966	0.728	1	0.999

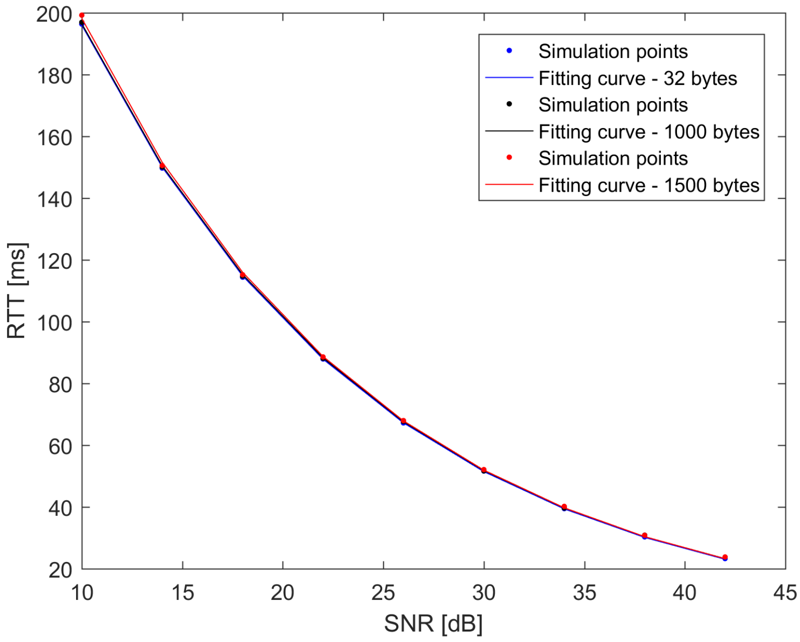


Figure 4.10: Latency fitting curves comparison for the rural area.

Latency comparison

In order to make some comparisons, all fitting curves from both environments and for the three packet sizes tested were plotted together. This graph is presented in Figure 4.11.

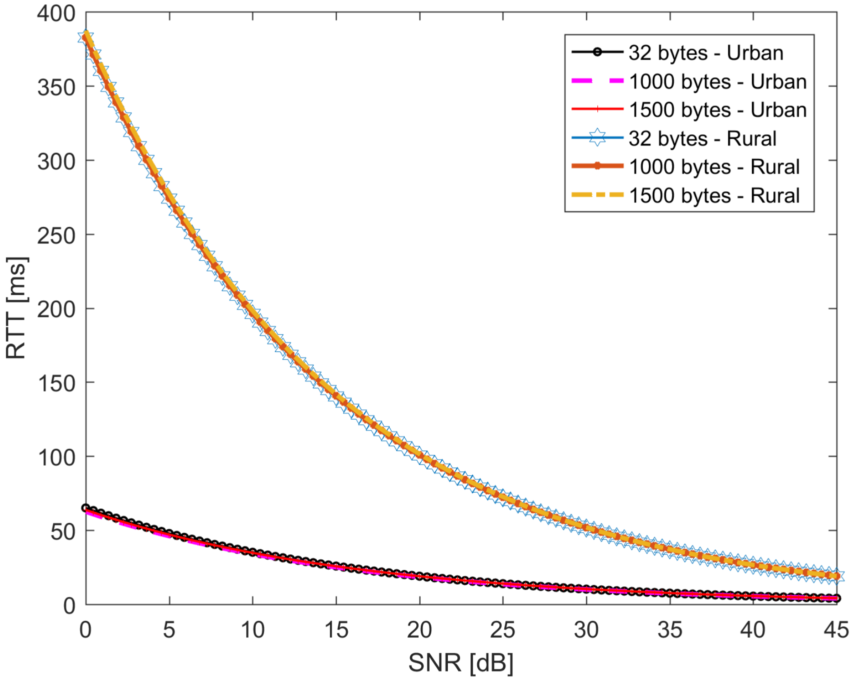


Figure 4.11: Latency fitting curves for both Urban and Rural areas.

Analyzing the curves obtained for the BS range defined in the reference scenario, and positioning the user according to an uniform distribution, it was achieved a minimum, average and maximum latencies, as shown in Table 4.9. Assuming a E2E latency requirement of 20 ms, according to ITU [46], the following results can be extracted:

Table 4.9: Latency comparison between emulator's tests.

Environment	Packet size	Minimum	Average	Maximum
Urban	32 bytes	4.014 ms	21.434	62.538 ms
	1000 bytes	4.165 ms	21.962 ms	63.746 ms
	1500 bytes	4.069 ms	22.128 ms	65.053 ms
Rural	32 bytes	18.878 ms	121.679	382.570 ms
	1000 bytes	19.058 ms	122.006 ms	382.600 ms
	1500 bytes	19.020 ms	123 ms	387.210 ms

- The minimum latency achieved for urban environments, for the three packet sizes, is around 80 % lower than the requirement; for rural environments is 5 % lower.
- For the average values, the obtained latency presents 1 or 2 ms more in urban scenarios; in the rural scenario the emulator reaches 100 ms more latency;
- For the worst signal conditions, in rural environments the latency obtained is three times higher than the requirement, and for rural environments nineteen times higher.

In general, for any communication network, the network latency is decomposed in three components:

$$\mathrm{RTT}_{\mathrm{Net[ms]}} = \mathrm{T}_{\mathrm{prop}} + \mathrm{T}_{\mathrm{trans}} + \mathrm{T}_{\mathrm{queu}}, \tag{4.4}$$

where the first component, the propagation delay ($\mathrm{T}_{\mathrm{prop}}$), is related to the distance travelled by the packets. The transmission delay ($\mathrm{T}_{\mathrm{trans}}$) is related with the size of the packets and the bandwidth of the channels where these packets are transmitted. The last component, the queueing delay ($\mathrm{T}_{\mathrm{queu}}$) is the time that a packet waits in a queue until it can be transmitted (also called the buffer).

Analyzing all latency results from Table 4.9, and taking (4.4) into account, some conclusions can be drawn:

1. The dependency between the latency and the SNR is directly related with the signal quality, and, consequently, with the distance between the UE and the BS. So, the exponential behaviour of the curve may be explained by the first component of the equation (4.4).
2. The packet size dependence, even slightly, is responsible for the second component of the previously mentioned equation. This small variation is expected since the wireless channel bandwidth, which is the one that limits the packet transmission, it is much larger than the size of the packets concerned that cross the network.
3. In the emulator there are no information regarding the queuing latency component. It is only possible to verify that, after a certain amount of time, if the packet cannot be transmitted it is discarded, and the packet loss is higher than zero.

In order to validate the emulator results, which addresses latency measurements and latency optimization in LTE, [47] was used as reference. It contains *ping* tests between an UE and an IP host (peer entity), very similar to what was implemented in the emulator. The tested connection includes an UE, an eNB, a router and two PCs, representing the terminal points, which execute the *ping* command. The RTT Key Performance Indicator (KPI) is measured according to cell position (RF conditions) and ICMP packet size, according to [47]. So, it is important to distinguish LTE RF conditions, and based on [48], the classification is done according to Table 4.10.

Table 4.10: RF Conditions [48].

		RSRP (dBm)	RSRQ (dBm)	SINR (dB)
RF conditions	**Excellent**	$\geq$ -80	$\geq$ -10	$\geq$ 20
	Good	-80 to -90	-10 to -15	13 to 20
	Mid Cell	-90 to -100	-15 to -20	0 to 13
	Cell Edge	$\leq$ -100	$<$ -20	$\leq$ 0

In [47] RTT values for the same three packet sizes tested in this thesis are presented, and they are valid for unloaded cells and good RF conditions, which can be compared with values obtained in the emulator. These values are presented in Figure 4.12 bar plot. The red line represents the average latency measured in the field.

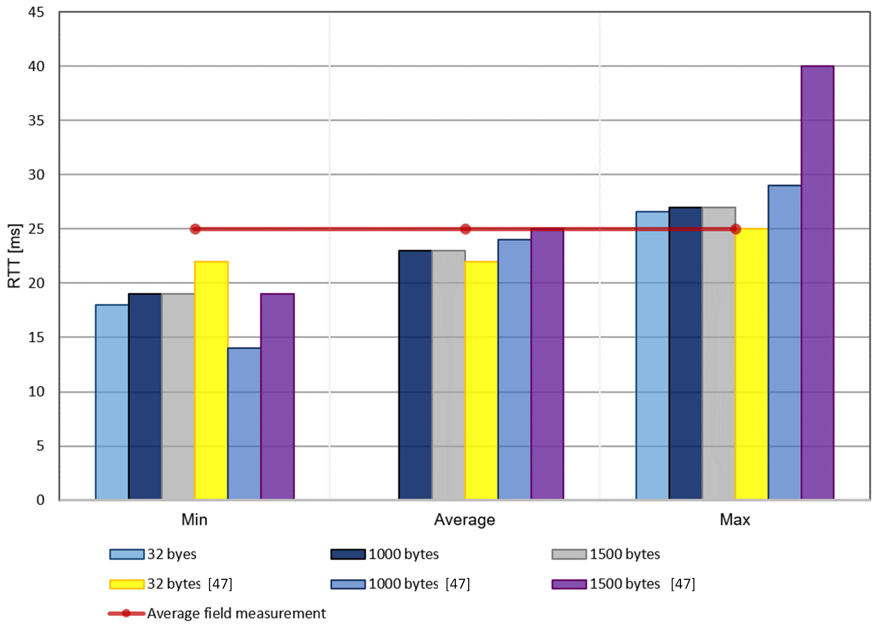


Figure 4.12: Latency comparison according to [47].

The emulator' RTT values represented in the graph of Figure 4.12 were extracted from the fitting curves previously obtained for the SNR range corresponding to good RF conditions. As one can see, the emulator results are quite similar to those obtained in the above mentioned reference. Regarding the minimum RTT values obtained for the SNR range considered, the emulator presented lower latency for all packet sizes, excluding the 1000 bytes packets, which presents 5 ms less than the other packet sizes, without apparent explanation on the reference. For the 32 bytes packets, the emulator reaches 4 ms less latency, which means a 18 % latency decrease. For the 1500 bytes, both emulator and reference tests present the same RTT, 22 ms. The emulator' average values, for 1000 and 1500 bytes, present less 4 % and 8 % latency respectively, and for 32 bytes both present 22 ms of latency. For the maximum values, the emulator present 7 % and 32.5 % less latency than [47], for 1000 and 1500 bytes respectively. For the 32 bytes, the emulator registers 6 % more latency. Also, the average field latency was measured by a *ping* application of the UEs operating system, and it was recorded the time difference between sending an ICMP *echo request* to an IP host and the reception of the corresponding ICMP *echo reply* message. This value is higher than the average, and slightly lower, 2-3 ms, than the maximum measurements performed in the emulator.

4.3 Use case - *Drive-tests*

In order to compare the latency results with real network measurements, a set of *drive-tests* were used, along with the network topology of Lisbon provided by a portuguese operator. Two types of *drive-tests* were used: one containing information regarding signal conditions, and other with RTT measurements. All *drive-tests* were used for the propagation model calibration and the second, containing the RTT information, were also used to compare the results measured with the ones obtained in the emulator.

Of all the data available in the *drive-tests*, the parameters used were:

- Physical Cell Identity (PCI);
- Transmission antenna height - h_{tx} ;
- Transmission antenna frequency - f ;
- LTE E-UTRAN band;
- Maximum transmission power - P_t ;
- Latitude and Longitude of the mobile user;
- Latitude and Longitude of the site.

4.3.1 Propagation Model calibration

In order to reproduce the *drive-tests*'s scenario, it was necessary to obtain an equation of losses which would allow to calculate the UE received power. To do so, two different propagation loss models were tested:

1. Standard Propagation Model (SPM)

   The SPM is a model used for network planning when calibration is required. It was developed based on the empirical formulas of Hata propagation and it is appropriate for mobile channel characterization. Ignoring the effects of diffraction, clutter and terrain, the resulting path loss equation was used [49]:

$$L_{p[dB]} = k_0 + k_1 \log_{10}(f_{[MHz]}) + k_2 \log_{10}(h_{tx[m]}) + [k_3 + k_4 \log_{10}(h_{tx[m]})] \log_{10}(d_{[km]}), \tag{4.5}$$

where h_{tx} represents the base station height, d is the distance between the UE used in the *drive-test* and the base station which is serving it, and f is the downlink channel frequency with a corresponding LTE E-UTRAN band. The constants k_0, k_1, k_2 and k_3, represent the parameters obtained by applying a fitting curve to the *drive-tests* data.

2. Ericsson model

   The second path loss model used, was created by Ericsson and is also used in network planning and in the deployment process of mobile communication networks. The Ericsson model stands on a modified Okumura-Hata model to allow room for changing in parameters based on signal level measurements, in 4G LTE networks [50]. This model has the following form:

$$L_{p[dB]} = k_0 + k_1 \log_{10}(d_{[km]}) + k_2 \log_{10}(h_{tx[m]}) + k_3 \log_{10}(h_{tx[m]}) \log_{10}(d_{[km]}) - 3.2(\log_{10}(11.75 h_m))^2 + g(f), \tag{4.6}$$

   where g(f) is

$$g(f) = 44.49 \log_{10}(f_{[MHz]}) - 4.78\,(\log_{10}(f_{[MHz]}))^2, \tag{4.7}$$

   and d is the distance between the *drive-test* and the base station which is serving it, $h_{tx[m]}$ the base station height, f is the downlink channel frequency and h_m is the mobile user height.

The *drive-tests* with the signal level were analyzed in two different areas, according to the cells corresponding to the RTT available *drive-tests*.

Scenario 1 - *Sapadores*

For this particular scenario, Table 4.11 values were used.

Table 4.11: Scenario 1 - *Drive-tests*' parameters.

Site name	PCI	Antenna height	Latitude ° N	Longitude ° W
RUA DOS SAPADORES - MACRO	447	23 m	38.72	-9.126
Frequency	**E-UTRAN band**	**Max tx power**		
1815 MHz	3	50 mWatt		

Others parameters were assumed, like both the transmission and reception antenna gain.

In order to get the path loss coefficients that better fit the environment in which the *drive-tests* were collected, a fit was performed, resulting in the curve of Figure 4.13.

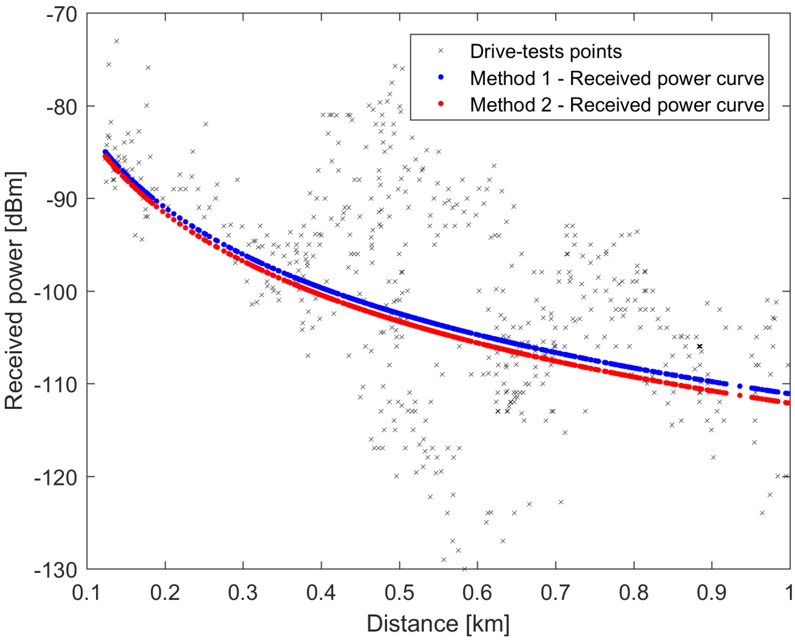


Figure 4.13: Received power curves - Scenario 1.

The resulting SPM curve is represented in blue and the Ericsson in red. Each one of these curves gave rise to the k_i coefficients presented in Table 4.12, as well as the respective RMSE.

Table 4.12: Scenario 1 - Propagation model parameters.

Model	k_0	k_1	k_2	k_3	k_4	RMSE
SPM	56.159	17.24	41.244	14.391	10.571	9.15 dB
Ericsson	40.861	14.391	30.009	10.571	-	9.15 dB

The area corresponding to scenario 1 is presented in Figure 4.14, along with the respective serving cell. The same figure represents the cell footprint according to the calibrated propagation loss model.

In order to obtain such footprint, an RF algorithm was used. In this algorithm, a map sample is cropped, corresponding to the surroundings of the target cell to which the footprint will be estimated. This map sample corresponds to 1 km ray from the cell centre. The map sample is divided as a grid, and in each square of the grid, the propagation model is applied to calculate the received power on that area. *Drive-tests* are used to calibrate the model, since they provide the real received power where they are measured. Thus, it is possible to calculate the real environment attenuation and then, calculate the coefficients that approximate the model to the drive-tests attenuation, using the Least Squares method. Basically, it is process similar to the Matlab® fitting function. The grid and cell schemes are presented in Figure 4.15.

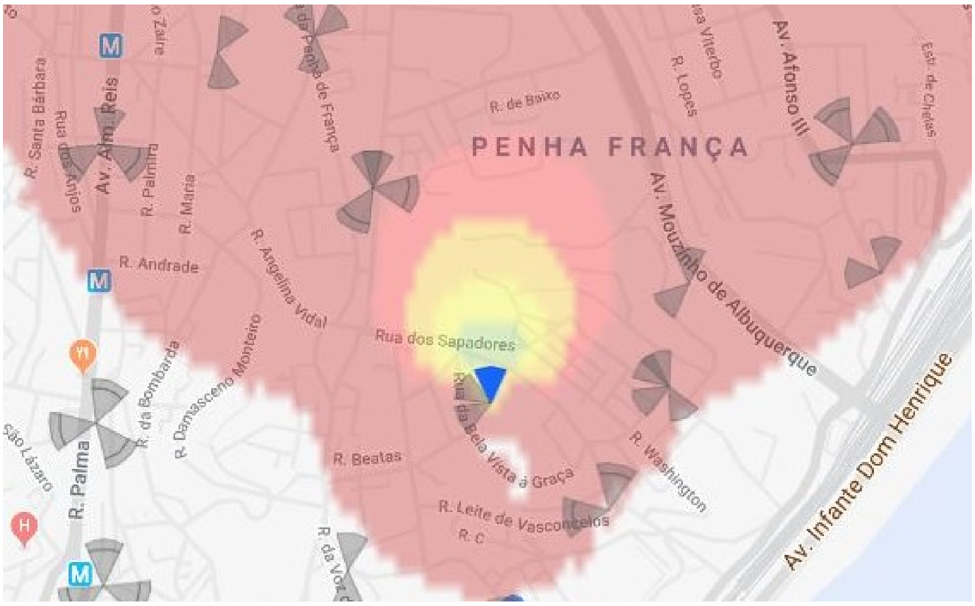


Figure 4.14: Scenario 1 - Footprint representation.

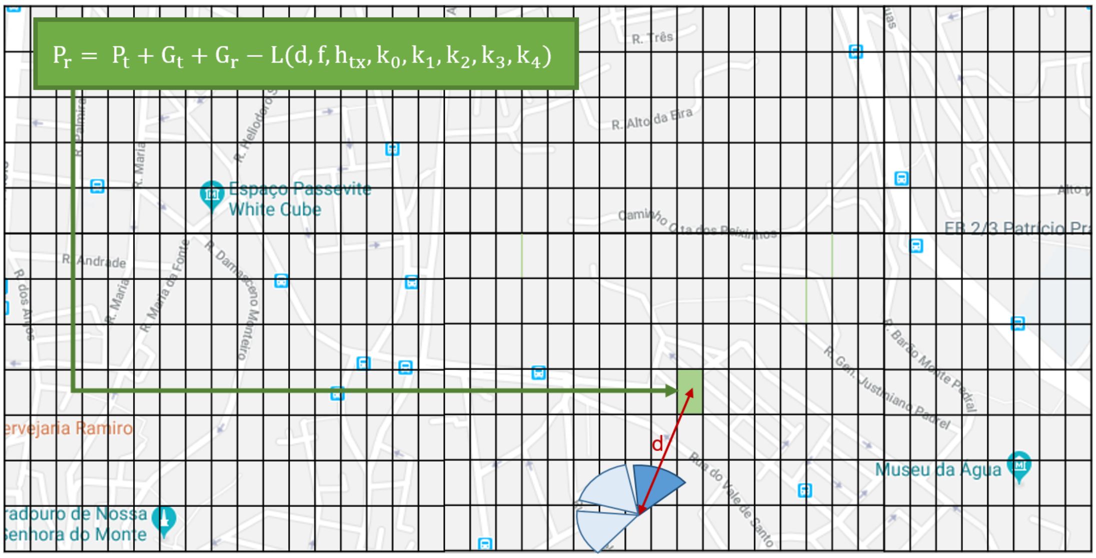

$P_r = P_t + G_t + G_r - L(d, f, h_{tx}, k_0, k_1, k_2, k_3, k_4)$
d

Figure 4.15: Algorithm explanation.

The algorithm described above, uses sector antennas, as represented in the previous figure, causing the transmission gain to vary with the horizontal and vertical angles between the antenna and the *drive-tests*.

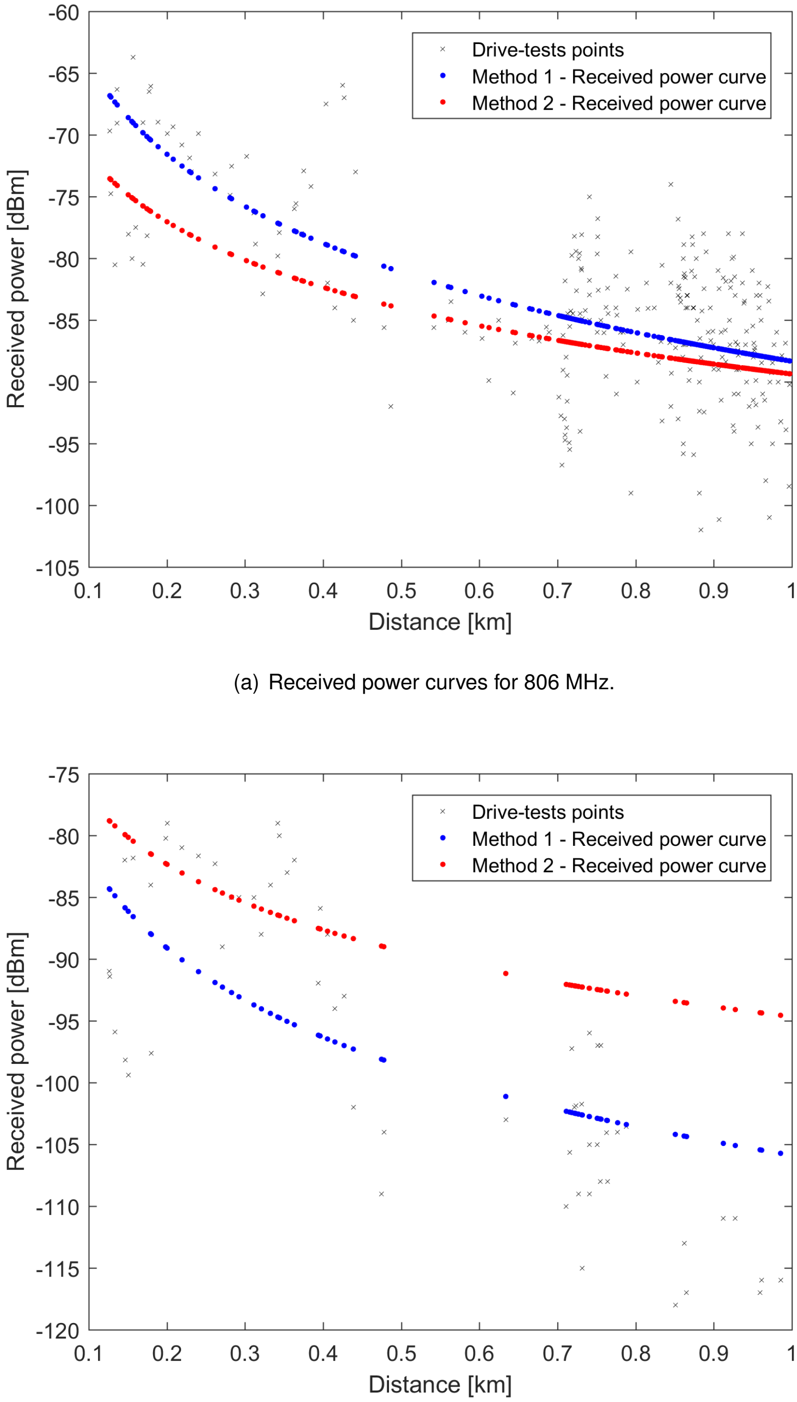
Scenario 2 - *Cidade Universitária*

The second scenario is part of the *Cidade Universitária* area, and for this scenario, the values from Table 4.13 were used together with the network topology and transmission antennas parameters.

Like the first scenario, Figure 4.16 shows the SPM curve in blue and the Ericsson in red. In Figure 4.16(a) the 806 MHz carrier *drive-tests* were plotted as well the 1815 MHz carrier *drive-tests* in Figure 4.16(b).

Table 4.13: Scenario 2 - *Drive-tests*' parameters.

Site name	**PCI**	**Antenna** height	**Latitude** ° N	**Longitude** ° W
CIDADE UNIVERSITARIA - MACRO	96	26 m	38.7518	-9.160
Frequency	**E-UTRAN band**	**Max tx power**		
1815 MHz	3	50 mWatt		
806 MHz	20			



(a) Received power curves for 806 MHz.

(b) Received power curves for 1815 MHz.

Figure 4.16: Received power curve - Scenario 2.

For both the propagation models, the resulting k_i coefficients from the fitting curves are presented in Table 4.14, along with the respective RMSE.

Table 4.14: Scenario 2 - Propagation model parameters.

Model	k_0	k_1	k_2	k_3	k_4	RMSE
SPM	1.8786	49.746	1.3306	12.001	8.4818	6.04 dB
Ericsson	32.926	8.8238	23.273	6.239	-	7.56 dB

The same footprint procedure was done for this area.

4.3.2 Latency analysis

With the propagation models calibrated according to the *drive-tests*' environment, the SPM was the model implemented in the emulator, since it presents better RMSE. Using the parameters previously described for each area, the results of latency were measured through *ping* tests for each SNR value, corresponding to a certain received power. Then, latency curves were traced according to the SNR.

The latency curves obtained were compared with *drive-tests* whit RTT measurements for the same two areas of Lisbon.

Scenario 1 - *Sapadores*

From the *drive-tests* with RTT information, only few had SNR and received power measurements. With those values, it was possible to predict the noise N, in order to calculate the SNR according to the received power, for the latency curve projection. These curve is presented in Figure 4.17, along with the *drive-tests*' points.

The latency curve has the form of the fitting curve presented in Section 4.2, and its coefficients are $b_1 = 17.044$ and $b_2 = -0.073963$.

Scenario 2 - *Cidade Universitária*

Similar to what was done in the first scenario, the SNR was obtained from the average of the noise N registered in the *drive-tests*. Thus, the latency curve was designed according to each SNR. These curve is presented in Figure 4.18, along with the points with the values from the *drive-tests*.

The fitting curve, as mentioned in the first scenario, coincides with the curve obtained in Section 4.2, and its coefficients are $b_1 = 21.475$ and $b_2 = -0.089154$.

Latency overview

Making an overall analysis of both scenarios, it can be extrapolated a general curve which includes all the RTT *drive-tests* available, positioned according to Figure 4.19. The final curve is the result of the average latency of both scenarios at each SNR point, and it is represented in Figure 4.20.

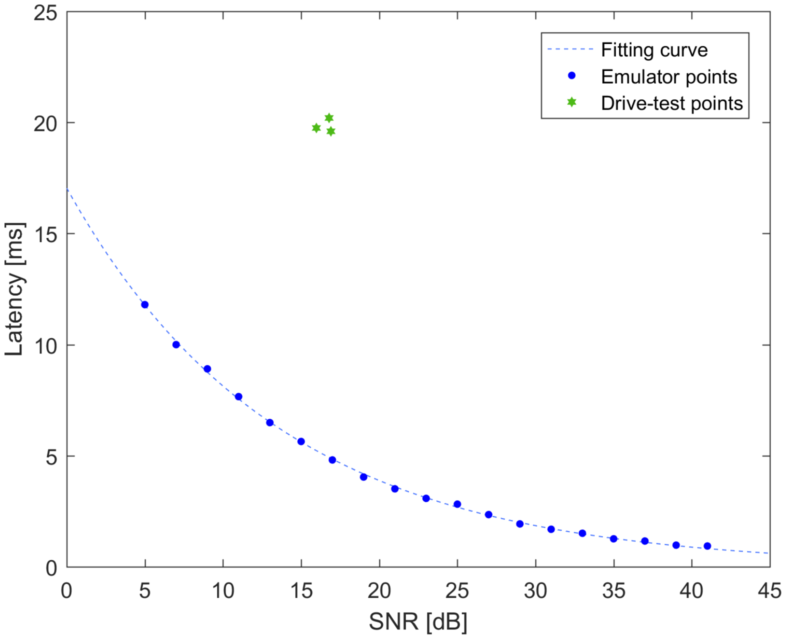


Figure 4.17: Scenario 1 - Latency fitting curve and *drive-tests*’ RTT.

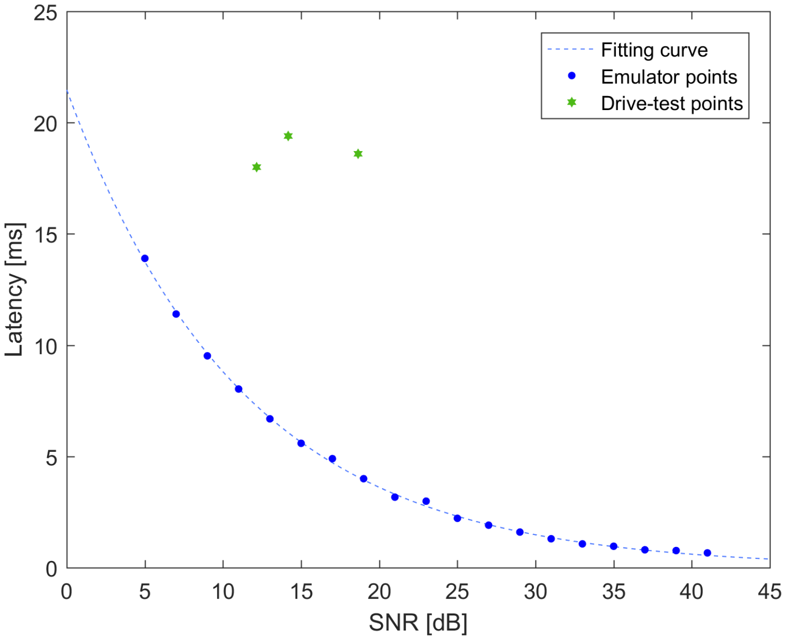


Figure 4.18: Scenario 2 - Latency fitting curve and *drive-tests*’ RTT.

Despite the large number of *drive-tests* available, many were duplicated. Then, after the removal of the duplicated data, most had only RTT values, that is, they had no information regarding the signal power or the signal-to-noise ratio. So in order to use those values, the following strategy was adopted:

1. Latitude and Longitude were extracted from both the *drive-test* and the cell site, which has the same PCI, and calculate the distance between them;

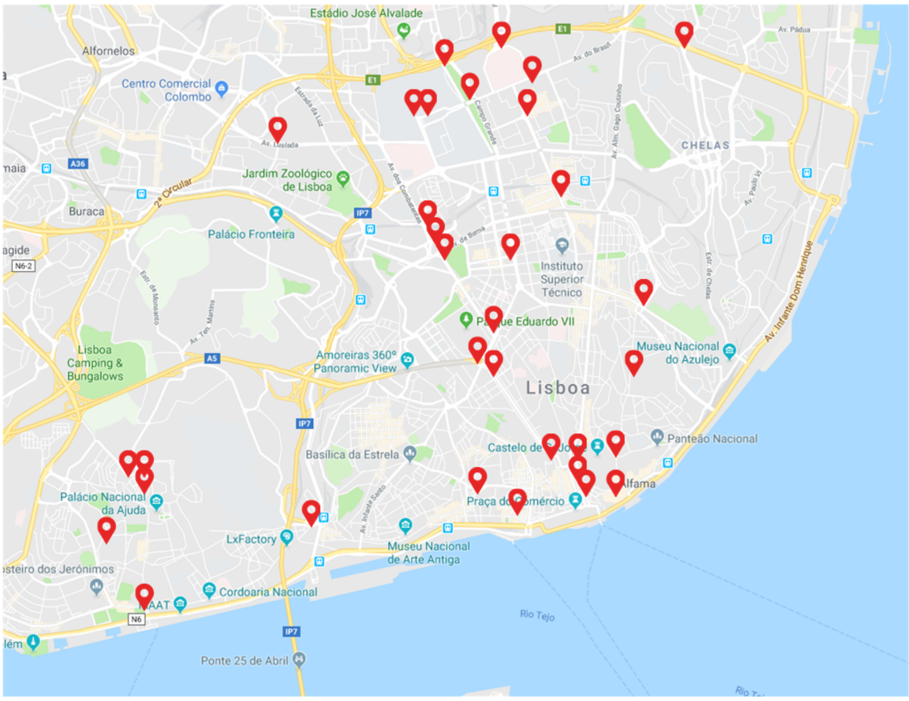


Figure 4.19: RTT *drive-tests*.

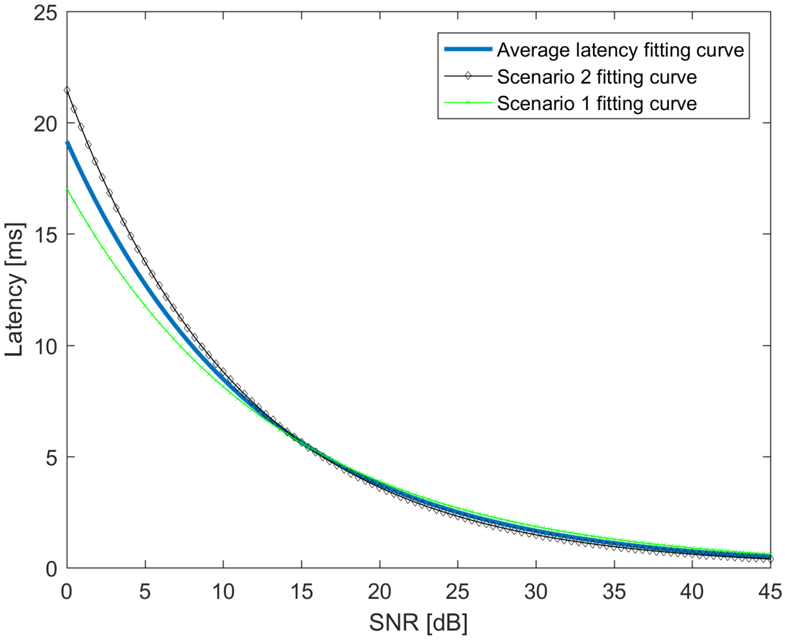


Figure 4.20: Latency curve obtained from scenarios 1 and 2.

2. Apply the SPM to obtain the path loss, followed by received power calculation;

3. Obtain the SNR with the received power, the transmission antenna parameters and the average of all *drive-tests*' noise;

4. Associate each RTT measurement to the corresponding SNR, and plot the *drive-tests*.

After applying these steps and having the SNR values for each *drive-test*, it is possible to compare the RTT from the *drive-tests* with the curve obtained in the emulator. This comparison can be observed in the graph of Figure 4.21.

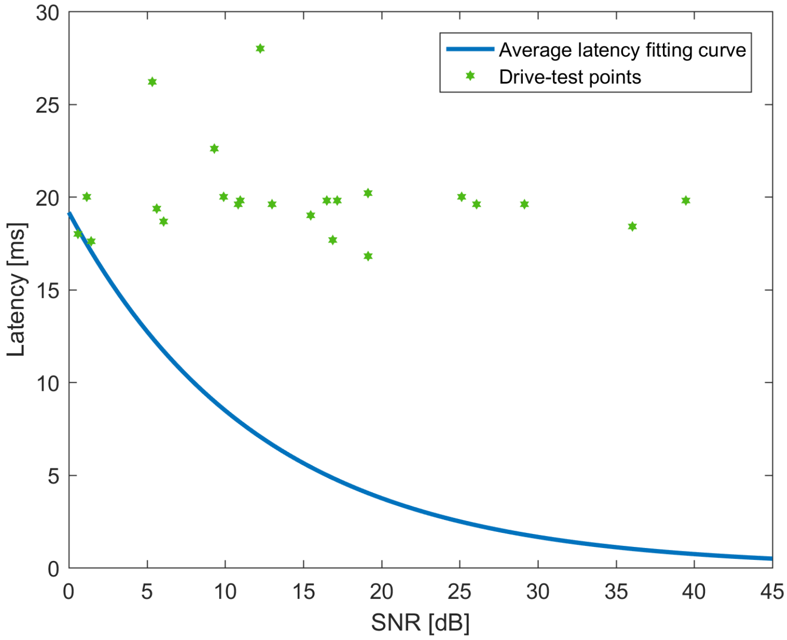


Figure 4.21: Latency curve obtained and RTT points from *drive-tests*.

The different behaviour of the two curves is notorious. The *drive-tests* have constant latency values for any SNR, varying between 18 and 30 ms, and presenting an average of 20.24 ms. On the other hand, it can be seen an exponential dependence between the latency and the SNR in the emulator. This variation of latency, in the emulator, happens in the wireless part of the network, and it is directly related to how it is implemented. Nevertheless, it presents very low values.

An analysis will be made, in the emulator, for the wired part of the network, in order to see how the SDN impacts the packet flow and the latency times.

4.3.3 End-to-End Latency

In order to analyze the influence of the SDN technology in packet forwarding, regarding the user plane latency, the wired part of the network was isolated, corresponding to the backhaul and core network. In this way it is possible to remove, in the emulator, the wireless contribution in the latency calculation. For this to happen, in the emulator an host was connected to the base station node in order to create a point to which traffic is transferred. The measurements were then performed, connecting the network elements in order to have a path between the base station antenna and an external network previously used and represented by the host *h*. Figure 4.22 presents the described architecture.

The *ping* test was performed, with 32 bytes packets being sent between hosts h1 and h2. The forwarding process continues to be managed by the SDN controller, which apply forwarding rules to the switches. The latency test was done by varying the number of switches on the connection, according to the topology from Figure 4.22, and the values of latency obtained are presented in Figure 4.23. It

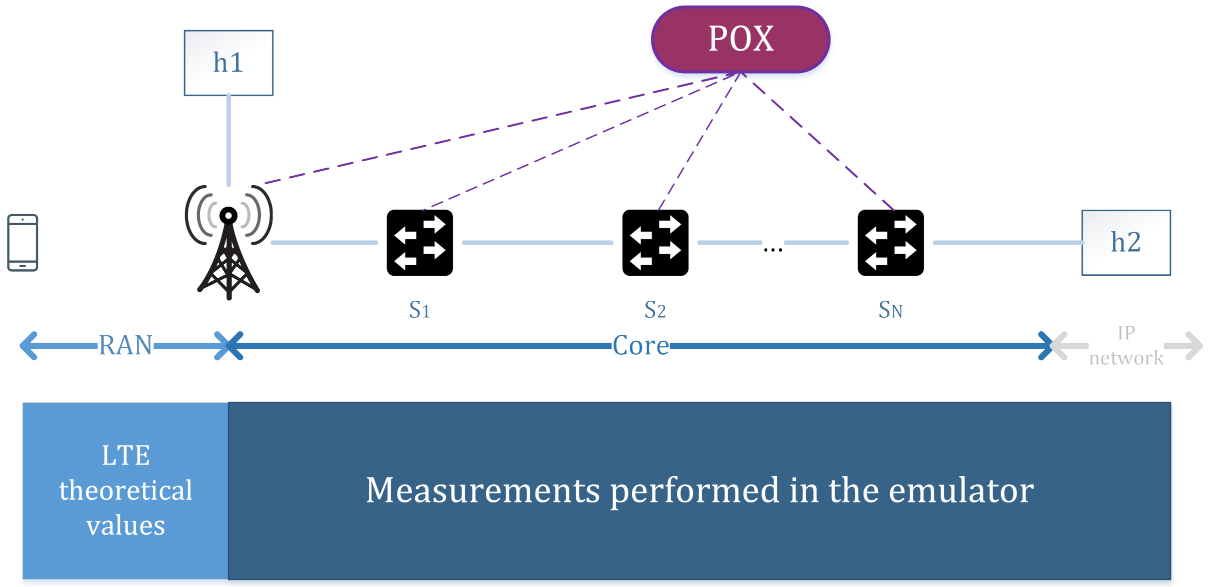


Figure 4.22: Wired network test in Mininet-WiFi.

can be seen that the average latency increases with the higher number of switches, even if non linearly. Moreover, for a small number of switches, very low values of latency were reached, corresponding to the minimum values. The latency values obtained in the *ping* tests, from h1 to h2, appear to be contained within a certain range of latency values, higher than 0.04 ms and smaller than 0.37 ms.

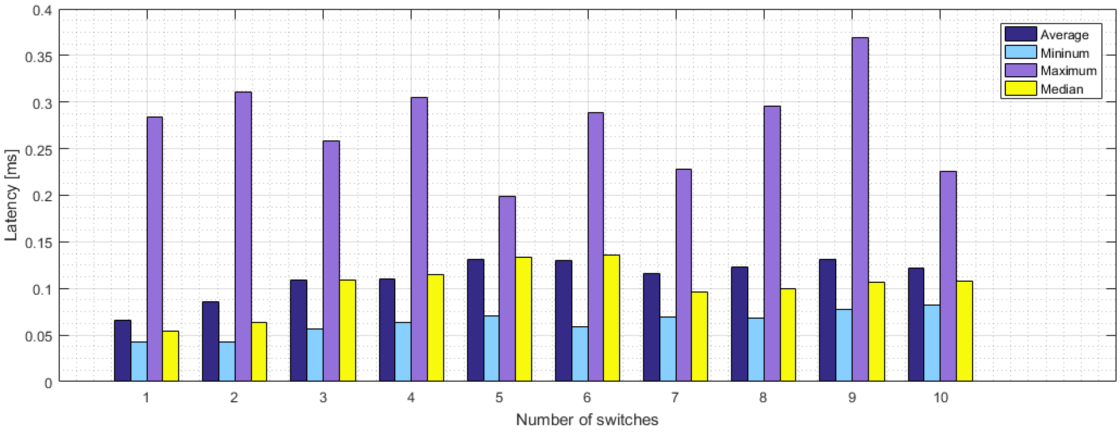


Figure 4.23: Latency according to the number of switches in the connection.

Assuming two switches, according to the topology described in Chapter 3, the average latency used for the wired part of the network obtained in the emulator was $T_{\text{emulator}} = 0.0853$ ms, taking into account that each packet goes through two switches in each direction. According to the latencies considered in Table 3.4, from Section 3.1, based on a LTE standard network, the following equation was assumed:

$$T_{\text{emulator}} = 2 \times T_{\text{S1-U}} \tag{4.8}$$

where the S1-U transfer delay, $T_{\text{S1-U}}$, is indicated by the yellow boxes in Figure 3.12, also from Section 3.1, and may vary between 1 and 15 ms.

Representing both emulator and 3GPP S1-U latencies in a bar plot the graph of Figure 4.24 was obtained.

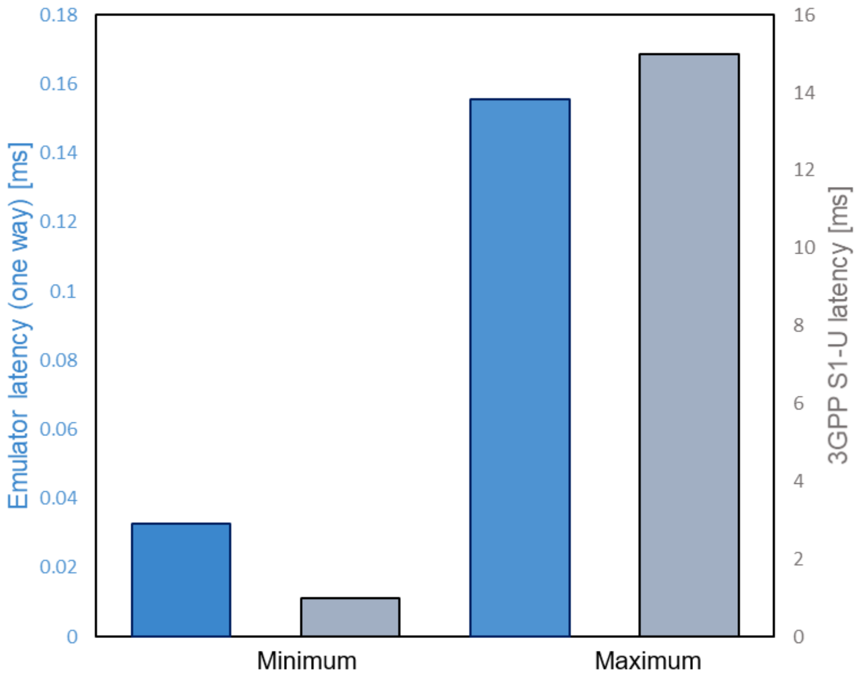


Figure 4.24: Emulator latency compared with the 3GPP S1-U latency.

Two comparisons can be made:

1. According to 3GPP [11] and ITU case study [51] for LTE:

   Assuming the latency values for the air interface and the processing delay of aGW presented in Table 3.4, along with the S1-U transfer delay ($T_{S1\text{-}U}$) presented in the graph of Figure 4.24, it is possible to compare the emulator latency with the values established in 3GPP [11]. The minimum and maximum values of $T_{S1\text{-}U}$ and $T_{emulator}$ (divided by two - representing one way latency) are listed in Table 4.15.

Table 4.15: RTT comparison between the emulator values and the values from 3GPP [11].

	HARQ	Air interface	S1-U	aGW	OWD	RTT
LTE (TR 125 192)	0 %	3.5 ms	1 ms 15 ms	0.5 ms	5 ms 19 ms	**10 ms** **38 ms**
	30 %	5 ms	1 ms 15 ms		6.5 ms 20.5 ms	**13 ms** **41 ms**
LTE (TR 125 192) + Emulator	0 %	3.5 ms	0.0327 ms 0.1555 ms	0.5 ms	4.033 ms 4.156 ms	**8.065 ms** **8.311 ms**
	30 %	5 ms	0.0327 ms 0.1555 ms		5.533 ms 5.656 ms	**11.066 ms** **11.312 ms**

Analyzing the values of Table 4.15, the following results can be extracted: for 0% HARQ the emulator presents 19.35 % and 78.13 % less latency than the 3GPP values, corresponding to the minimum and maximum Ts1u , respectively; for 30 % HARQ, the emulator presents a 14.88 % and 72.41 % latency decrease regarding 3GPP, once again for the Ts1u minimum and maximum respectively.

The average latency is not represented in the Table 4.15 but using the average values for both air interface and $T_{S1\text{-}U}$, 4 ms and 7.5 ms respectively, and adding the aGW latency, it results in a 3GPP RTT of 24 ms. Using the same 4 ms for the air interface, for each way, and the average emulator RTT latency (0.0853 ms), plus the aGW delay, the emulator registers an average RTT of 9.085 ms, a value 62.15 % lower than the average 3GPP value.

In a case study conducted by ITU [51], the U-Plane latency, end-to-end, is defined as 12.2 ms in downlink and 20.6 ms in uplink. The steps that contribute to these latencies results are presented in Appendix B. According to these values, a latency of 32.8 ms can be assumed, considering:

$$\mathrm{RTT} = \mathrm{T}_{\mathrm{downlink}} + \mathrm{T}_{\mathrm{uplink}}, \tag{4.9}$$

where $\mathrm{T}_{\mathrm{downlink}}$ is the U-Plane latency in the downlink and $\mathrm{T}_{\mathrm{uplink}}$ is the U-Plane latency in uplink. The values used by ITU are presented in Table 4.16.

Table 4.16: RTT comparison between the emulator values and the values from ITU [51].

	Direction	Air interface	S1-U	aGW	T_{OWD}	RTT
ITU [51]	Downlink	5.197 ms	7 ms		12.197 ms	**32.795 ms**
	Uplink	5.598 ms	15 ms		20.598 ms	
Emulator	Downlink	5.197 ms	0.04265 ms	0.5 ms	5.740 ms	**11.881 ms**
	Uplink	5.598 ms	0.04265 ms	0.5 ms	6.141 ms	

Following the methodology used previously for the 3GPP reference comparison, it were analyzed the emulator values, adopting the same S1-U latency for both downlink and uplink, and the air interface values obtained from ITU. The emulator registered a 63.77 % latency decrease.

2. According to *drive-tests*:

   Using the latency average value obtained in the emulator, corresponding to the RTT ($2 \times T_{S1\text{-}U}$), and the ITU air latency, presented in Table 4.16, a comparison can be made with the *drive-tests* for each SNR value. Plotting the RTT values from the *drive-tests* along with the emulator RTT, which do not depend on the SNR, the results in Figure 4.25 were obtained.

   As one can see, the average latency obtained in the emulator is much lower than the values of the *drive-tests*; more specifically, 40.18 % lower. On the other hand, it can be verified that *drive-tests* latency is, mostly, lower than the value obtained by ITU.

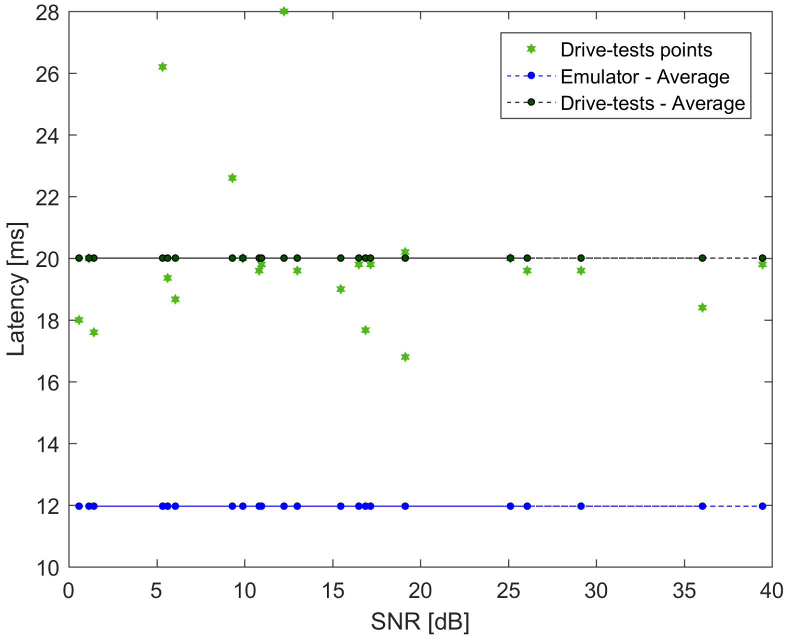


Figure 4.25: RTT comparison between *drive-tests* and Emulator/ITU.

4.4 Use Case - Energy saving

In LTE, load balancing and energy saving are two tightly coupled features. They aim at accomplishing a better distribution of users through the available eNodeBs and, at the same time, reduce energy consumption by avoiding unnecessary resources usage. An embryonic use case related with these two concepts started being developed in this master thesis, with the creation of a Python framework. It relies on having a generic topology, with mobile terminals (UEs) randomly moving. Depending on the number of users associated to each access point from the topology, operations of load balance or energy saving, are performed. Figure 4.26 provides an illustrative diagram of the framework operation. The framework has three processes running simultaneously per RRH, two concerning the turning on/off the RRH cells and a third controlling the connected stations. The controller is responsible for managing all these processes.

Turning off conditions:

– No users in a specific cell for more than a defined time interval (depending on the traffic on the neighbouring cells).

Turning on conditions:

– Excessive traffic in the neighbouring cells;

– To face a possible coverage hole.

Additional features:

- Every time there are users served by different cells that can be served by only one, without overwhelming that cell, users are transferred to that same cell.

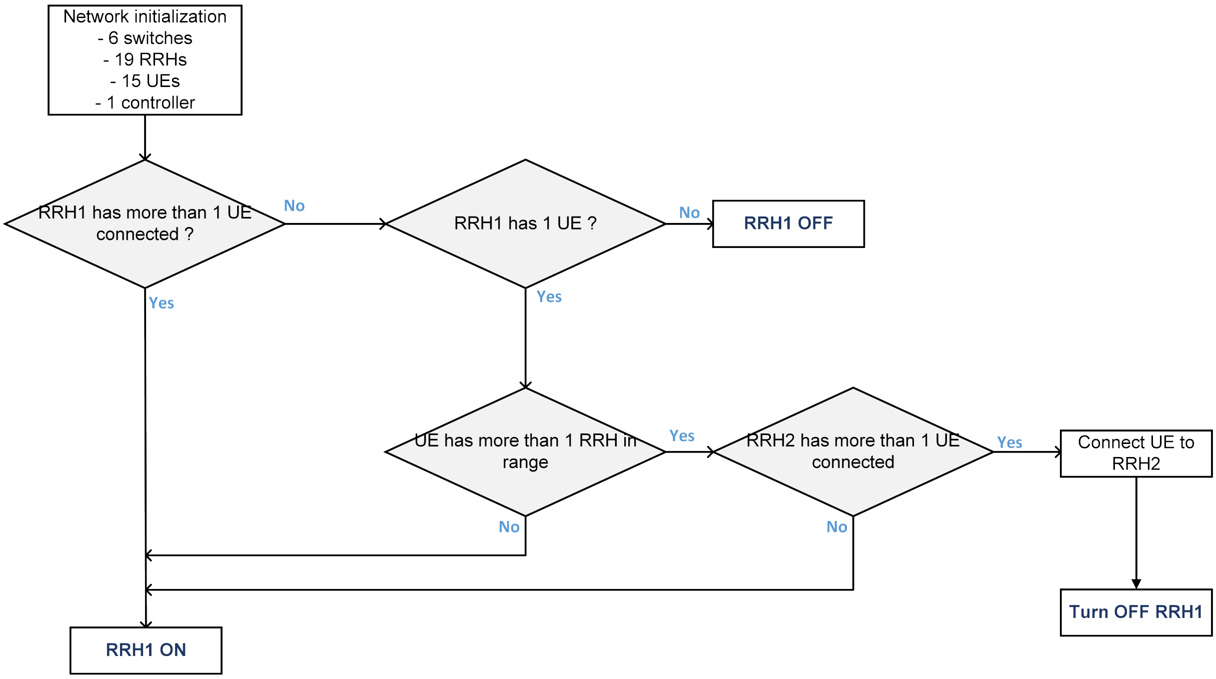


Figure 4.26: Energy saving diagram.

The processes previously described are showed in Figure 4.27, during the program execution. It can be seen that the RRHs1,2,8,9 and 6, were turned off after the users (here represented by 'sta') were connected to another RRH.

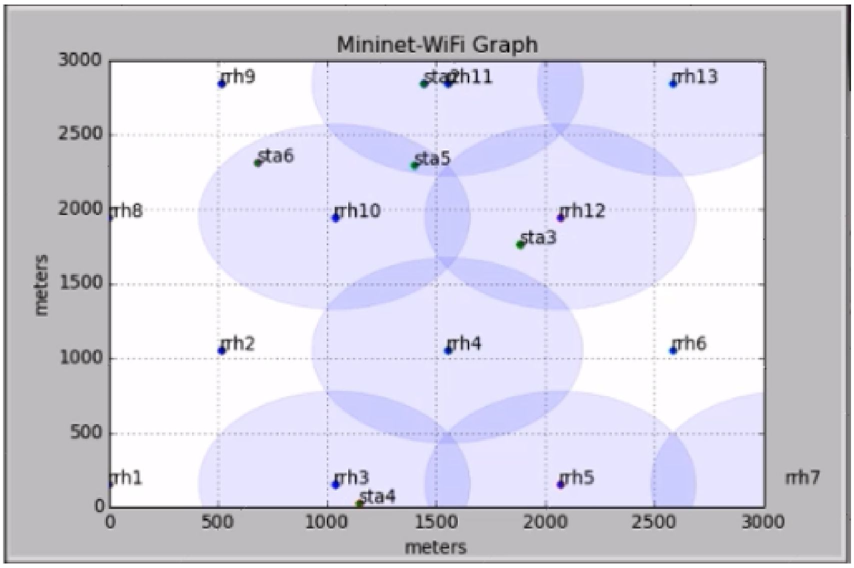


Figure 4.27: Energy saving execution.

All the mentioned features, when combined, allow going towards the load balancing and energy savings solutions. The implemented framework is intended to be fully operational with SDN technology. For now, since it only relies on basic forwarding rules of a SDN controller, few results can be taken from it. However, it allows to percept the ease on managing these type of features on an SDN based network since, by changing the algorithm (framework) in the controller part, it is possible to spread all the functionalities to the entire network, contrarily to what happens in LTE networks, where the algorithm needs to be loaded in each base station. In the future, integration of specific OpenFlow messages is needed, as well as the fusion with an LTE simulator, in order to study the results in terms of processing time or traffic overhead in a scenario as closer from reality as possible. Despite that, the implemented framework is operational to work as a basis for those future enhancements.

Chapter 5

Conclusions

This chapter finalizes this work, and it will summarize conclusions and enumerate aspects to be developed in future work.

5.1 Summary

This work started by introducing LTE concepts and the new technologies on which this thesis is based, namely SDN, NFV and C-RAN, and how they can be used in mobile networks.

The main goal was to evaluate the performance of a 4G network with SDN capabilities, and to do that, the Mininet-WiFi, an SDN emulator, was used to merge the mentioned technologies, towards a LTE RAN SDN based solution, which represents one of the strongest architectures concerning the 5G world.

The proposed approach crossed one major issue, which was the fact that the wireless part of the emulator, was implemented for WiFi networks. In order to adapt the emulator to this project needs, LTE modules were developed in the radio access network, namely, the RF scenario and the propagation loss model, the modulation and coding scheme and the RRHs' grid. The core network of both technologies is IP based and present similar topology, so the SDN approach for both was assumed to be similar. Also, in the backhaul/core network, since the exchanged messages are inherent in both LTE and WiFi technologies, and the channel was characterized according to LTE, through the LTE developed modules, it was possible to test a LTE connection over the mac80211 virtual driver from the WiFi emulation component.

After these proceedings, it was possible to start doing tests comparing the LTE SDN based approach with a traditional LTE network. The latency and throughput QoS metrics were the ones focused in this project.

Starting with the throughput analysis, it was obtained a spectral efficiency curve, for each environment, which presented very good results regarding the three metrics used, the RMSE with a value of 0.0783 dB for the urban environment and 0.0682 dB for rural, and the R^2 and correlation very close to 1. Analyzing the throughput regarding the network performance, the proposed solution for the urban environment, presented a 54 % spectral efficiency increase regarding the ITU requirement. For rural areas, it was obtained a 158 % spectral efficiency increase, also using ITU as reference.

Concerning the latency tests, an exponential dependence with SNR was obtained, for three packet sizes and the two environments, urban and rural. These tests also presented good RMSE, R^2 and correlation between the simulated points and the fitting curve obtained. For the three packet sizes in urban environments, the obtained average latency was around 21-22 ms, and for the rural environment the latency rises to 120 ms. Using [47] as reference to compare the RTT measurements, the emulator results were quite similar to the ones obtained in the reference, for good RF conditions. The minimum RTT values obtained in the emulator represent a 18 % latency decrease for 32 bytes packets; and for 1000 and 1500 bytes the results obtained are the same or worst. The average values are 4 % and 8 % lower in emulator for 1000 and 1500 bytes. The maximum latency in reference reaches 40 ms, 67.5 % more latency than the one obtained in the emulator. Summing up, regarding this particular reference, although not revolutionary, the emulator results are encouraging.

As an use case, *drive-tests* were used to compare the latency with real network measurements. They were initially applied to the calibration process of two propagation models, the SPM and Ericsson models, which better reflect the experimental environment (urban). The SPM was then implemented into the emulator, since it presented the best RMSE result. The latency measurements were then extracted from the emulator and compared with the *drive-tests* latency. The *drive-tests* latency is mostly constant, around 20 ms, with some punctual fluctuations within 18 ms and 30 ms, but SNR independent, while in the emulator, the latency results were considerably smaller but depended on the SNR.

The second part of the use case addressed the backhaul/core latency individually, and three comparisons were made. The first one refers to 3GPP [11], and the obtained results are presented in Table 5.1.

Table 5.1: RTT reduction regarding the reference [11] from 3GPP.

HARQ	Air interface	S1-U	RTT reduction
0 %	3.5 ms	Minimum Maximum	**19.35 %** **78.13 %**
30 %	5 ms	Minimum Maximum	**14.88 %** **72.41 %**
Avg	4 ms	7.5 ms	**62.15 %**

The second comparison is related with the ITU [51] case study, and the emulator reaches a 63.77 % latency reduction. The last comparison, relates the emulator backhaul/core latency with the *drive-tests* values; where the emulator presents 40.18 % less latency.

After all, it can be concluded, that for the 3GPP RF scenario dimensioned, the latency results for the urban environment are acceptable, but the rural scenario latencies are very high. On the other hand, the values obtained for the use case presents very good results when compared with the references.

The results obtained in this project show that an SDN approach applied to mobile networks has potential for future standardization, since the results obtained in the simulations show a better performance even with an adapted emulator. In this sense, the results are expected to improve with future technology enhancements.

5.2 Future Work

In the future it is intended to develop other LTE modules to include in the emulator, namely the sector antennas implementation and representation, and the handover process. Several use cases can also be developed over the knowledge acquired during this project, which contributes as a base for SDN architecture and functionalities. The prior use cases to be implemented are the energy saving and load balance ones, where some work is already done, as shown in Section 4.4. Some of the main advantages in a SDN based network is the flexibility in performing some network features as well as its development. Regarding this, it is expected this project to work as a basis for future tests with the explored technologies and where enhancements can easily be implemented.

References

[1] Ericsson. Future mobile data usage and traffic growth, 2016. https://www.ericsson.com/en/mobility-report/future-mobile-data-usage-and-traffic-growth [Online; accessed 18-May-2017].

[2] Mads Lauridsen, Lucas Chavarra Gimnez, Ignacio Rodriguez, Troels B. Srensen and Preben Mogensen. From LTE to 5G for Connected Mobility. *IEEE Communications Magazine (Vol: 55, Issue: 3)*, 2017.

[3] Ramon dos Reis Fontes, Christian Rodolfo Esteve Rothenberg. Mininet-WiFi software, 2018. https://github.com/intrig-unicamp/mininet-wifi [Online; accessed 28-April-2018].

[4] Li Erran Li, Z. Morley Mao, Jennifer Rexford. Toward Software-Defined Cellular Networks. *Software Defined Networking (EWSDN), 2012 European Workshop*, 2012. [Online; accessed 13-May-2018].

[5] Sudeep Palat, Philippe Godin. The LTE Network Architecture. *Published at Alcatel/Lucent White Paper*, 2009. [Online; accessed 13-May-2018].

[6] ANACOM. Information on multi-band spectrum auction (3). Public Consultation, Lisbon, Portugal, 12 2012. http://www.anacom.pt/render.jsp?contentId=1106646&languageId.

[7] Stefania Sesia, Issam Toufik, Matthew Baker. *LTE – The UMTS Long Term Evolution: From Theory to Practice*. John Wiley & Sons, Ltd, 2 edition, 2011. ISBN: 978-0-470-66025-6.

[8] Anritsu. LTE Resource Guide. Document Number: Anritsu LTE Resource Guide v1, 02 2015. http://www.cs.columbia.edu/6181/hw/anritsu.pdf.

[9] M. Sauter. *From GSM to LTE: an introduction to mobile networks and mobile broadband*. John Wiley & Sons, Ltd, 2010.

[10] H. Holma and A. Toskala. *LTE for UMTS: Evolution to LTE Advanced*. John Wiley & Sons, Chichester, UK, 2 edition, 2011.

[11] *"Universal Mobile Telecommunications System (UMTS); Feasibility study for evolved Universal Terrestrial Radio Access (UTRA) and Universal Terrestrial Radio Access Network (UTRAN) (3GPP TR 25.912 version 7.1.0 Release 7)"*. ETSI, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE, 2006.

[12] 4G Americas. LTE Aggregation & Unlicensed Spectrum. White Paper, 11 2015.

[13] Intel Corporation Sasha Sirotkin. LTE-WLAN aggregation (LWA): Benefits and Deployment Considerations. White Paper, 2016.

[14] G. Vijayalakshmy, G. Sivaradje. Interworking of WLAN-LTE for next generation wireless networks. *Information Communication and Embedded Systems (ICICES), 2014 International Conference*, 2014. [Online; accessed 26-April-2018].

[15] Stephany Merritt. An Introduction to Wireless LANs, 2016. https://http://slideplayer.com/slide/8983489/[Online; accessed 28-April-2018].

[16] ns 3 project. ns-3, 2018. https://www.nsnam.org/docs/models/html/lte-design.html [Online; accessed 28-April-2018].

[17] Michael Jarschel, Thomas Zinner, Tobias Hoßfeld, Phuoc Tran-Gia, Wolfgang Kellerer. Interfaces, Attributes, and Use Cases: A Compass for SDN. *IEEE Communications Magazine*, 2014. [Online; accessed 19-May-2017].

[18] Manar Jammal, Taranpreet Singh, Abdallah Shami, RasoolAsal, and Yiming Li. Software-defined networking: State of the art and research challenges. *Submitted for review and possible publication in Elsevier's Journal of Computer Networks*, 2016. [Online; accessed 18-May-2017].

[19] Open Networking Foundation. Openflow switch specification. *ONF TS-006*, 2012. [Online; accessed 18-May-2017].

[20] Bruno Astuto A. Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. *Published at https://hal.inria.fr*, 2014. [Online; accessed 18-May-2017].

[21] Open Networking Foundation. Software-defined networking: The new norm for networks. *ONF White Paper*, 2012. [Online; accessed 18-May-2017].

[22] Jose Costa-Requena, Raimo Kantola, Jesús Llorente Santos, Vicent Ferrer Guasch, Maël Kimmerlin, Antti Mikola, and Jukka Manner. *LTE Architecture Integration with SDN*. John Wiley & Sons, Ltd., 2015.

[23] Jérémy Pagé, Jean-Michel Dricot. Software-Defined Networking for Low-Latency 5G Core Network. *Military Communications and Information Systems (ICMCIS), 2016 International Conference*, 2016. [Online; accessed 20-April-2018].

[24] Arsany Basta, Wolfgang Kellerer, Marco Hoffmann, Hans Jochen Morper, Klaus Hoffmann. Applying NFV and SDN to LTE mobile core gateways, the functions placement problem. *AllThingsCellular '14 Proceedings of the 4th workshop on All things cellular: operations, applications, & challenges (SIGCOMM)*, 2014. [Online; accessed 13-May-2018].

[25] ETSI Industry Specification Group (ISG). Network Functions Virtualisation (NFV); Architectural Framework. *Published at http://www.etsi.org/*, 2013. [Online; accessed 13-May-2018].

[26] Jose Costa-Requena, Jesús Llorente Santos, Vicent Ferrer Guasch, Kimmo Ahokas, Gopika Premsankar, Sakari Luukkainen, Oscar López Pérez, Mikel Uriarte Itzazelaia, Ijaz Ahmad, Madhusanka Liyanage, Mika Ylianttila, Edgardo Montes de Oca. SDN and NFV integration in generalized mobile network architecture. *Networks and Communications (EuCNC), 2015 European Conference*, 2015. [Online; accessed 28-April-2018].

[27] Van-Giang Nguyen, Truong-Xuan Do, and YoungHan Kim. SDN and Virtualization-Based LTE Mobile Network Architectures: A Comprehensive Survey. *Published online at http://Springerlink.com*, 2015. [Online; accessed 09-May-2017].

[28] Intel technologies. Towards 5G – RAN Virtualization Enabled by Intel and ASTRI. White Paper, 2017. [Online; Accessed in 29-April-2018].

[29] Imtiaz Parvez, Ali Rahmati, Ismail Guvenc, Arif I. Sarwat and Huaiyu Dai. A Survey on Low Latency Towards 5G: RAN, Core network and caching solutions. *IEEE Communications Surveys and Tutorials*, 2017. [Online; accessed 26-April-2018].

[30] Chen-Nien Mao, Mu-Han Huang, Satyajit Padhy, Shu-Ting Wang, Wu-Chun Chung, Yeh-Ching Chung and Cheng-Hsin Hsu. Minimizing latency of real-time container cloud for software radio access networks. *Cloud Computing Technology and Science (CloudCom), 2015 IEEE 7th International Conference*, 12 2015. [Online; accessed 22-April-2018].

[31] Henrik Lehrmann; Yan Ying; Scolari Lara; Kardaras Georgios; Berger-Michael Stübert; Dittmann-Lars Checko, Aleksandra; Christiansen. Cloud RAN for Mobile Networks— A Technology Overview. *IEEE Communications Surveys & Tutorials (Volume: 17, Issue: 1)*, 2015. [Online; accessed 29-April-2018].

[32] Chen Kuilin and Duan Ran. C-RAN The Road Towards Green RAN. *China Mobile Research Institute*, 2011. [Online; accessed 18-May-2017].

[33] Mustafa Y. Arslan, Karthikeyan Sundaresan, and Sampath Rangarajan. Software-defined networking in cellular radio access networks: Potential and challenges. *IEEE Communications Magazine • January 2015*, 2015. [Online; accessed 20-April-2017].

[34] Mohammed Yazid Lyazidi, Nadjib Aitsaadi, and Rami Langar. Dynamic Resource Allocation for Cloud-RAN in LTE with Real-Time BBU/RRH Assignment. *IEEE ICC 2016 - Next-Generation Networking and Internet Symposium*, 2016. [Online; accessed 07-June-2017].

[35] Mininet Team. Mininet software, 2017. http://mininet.org/[Online; accessed 09-May-2017].

[36] mac80211_hwsim. https://wireless.wiki.kernel.org/en/users/drivers/mac80211_hwsim, 01 2015. [Online; Accessed in 14-February-2018].

[37] Christian Esteve Rothenberg Ramon dos Reis Fontes. Mininet-wifi - the user manual. Published at github.com/intrig-unicamp/mininet-wifi, 03 2018.

[38] Pox, 2015. https://noxrepo.github.io/pox-doc/html/[Online; accessed 30-April-2018].

[39] ETSI 3rd Generation Partnership Project (3GPP). Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency (RF) system scenarios (3GPP TR 36.942 version 10.2.0 Release 10). *ETSI TR 136 942 V10.2.0 (2011-05)*, 2011. [Online; accessed 05-June-2017].

[40] *"LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (3GPP TS 36.213 version 8.8.0 Release 8)"*. ETSI, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE, 2009.

[41] Pedro Manuel de Almeida Carvalho Vieira. *Spatial Multiplexing MIMO Capacity Enhancement for 4G Macro-cell Networks*. PhD thesis, Instituto Superior Técnico, 2008.

[42] RSRP and RSRQ measurement in LTE. https://www.laroccasolutions.com/78-rsrp-and-rsrq-measurement-in-lte/, 04 2016. [Online; Accessed in 26-April-2018].

[43] ESnet / Lawrence Berkeley National Laboratory. Iperf, 2003. https://iperf.fr/[Online; accessed 17-January-2018].

[44] *"Universal Mobile Telecommunications System (UMTS); LTE; Feasibility study for evolved Universal Terrestrial Radio Access (UTRA) and Universal Terrestrial Radio Access Network (UTRAN) (3GPP TR 25.912 version 9.0.0 Release 9)"*. ETSI, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE, 2009.

[45] Preben Mogensen, Wei Na, István Z. Kovács, Frank Frederiksen, Akhilesh Pokhariyal, Klaus I. Pedersen, Troels Kolding, Klaus Hugl and Markku Kuusela. Lte capacity compared to the shannon bound. *Vehicular Technology Conference, Ed: 65*, 2017. [Online; accessed 05-March-2018].

[46] ITU-R M.2134. Requirements related to technical performance for imt-advanced radio interface(s), 11 2008.

[47] Xincheng Zhang. *LTE Optimization Engineering Handbook*. Wiley-IEEE Press, 2018. ISBN: 978-1-119-15897-4.

[48] RSRP and RSRQ measurement in LTE. https://www.laroccasolutions.com/164-rsrq-to-sinr/, 04 2016. [Online; Accessed in 26-April-2018].

[49] Segun I. Popoola, Aderemi A. Atayero, Nasir Faruk, Carlos T. Calafate, Emmanuel Adetiba and Víctor O. Matthews. Calibrating the standard path loss model for urban environments using field measurements and geospatial data. *The 2017 International Conference of Wireless Networks*, 2017. [Online; accessed 26-April-2018].

[50] Ph. Atanasov, Zh. Kiss'ovski. Optimization of Path Loss Models Based on Signal Level Measurements in 4G LTE Network in Sofia. Heron Press Ltd., 06 2017.

[51] Martin Brand, Joachim Pomy. One-way Delays in Operating LTE Networks. ITU Workshop on " Monitoring and Benchmarking of QoS and QoE of Multimedia Services in Mobile Networks ", 07 2014.

Appendix A

LTE Modulation and Coding Scheme

Table A.1: 4-bit CQI Table [40]

CQI	modulation	code rate x 1024	efficiency
	out of range		
1	QPSK	78	0.1523
2	QPSK	120	0.2344
3	QPSK	193	0.3770
4	QPSK	308	0.6016
5	QPSK	449	0.8770
6	QPSK	602	1.1758
7	16QAM	378	1.4766
8	16QAM	490	1.9141
9	16QAM	616	2.4063
10	64QAM	466	2.7305
11	64QAM	567	3.3223
12	64QAM	666	3.9023
13	64QAM	772	4.5234
14	64QAM	873	5.1152
15	64QAM	948	5.5547

Appendix B

3GPP U-Plane latency

According with [44] for FDD the LTE U-plane one way latency for a scheduled UE consists of the fixed node processing delays (which includes radio frame alignment) and 1 ms Transmission Time Interval (TTI) duration. Considering that the number of HARQ[1] processes is fixed to 8 for FDD, the one-way latency can calculated as:

$$\mathrm{DUP[ms]} = 1 + 1.5 + 1 + \mathrm{n} \times 5, \tag{B.1}$$

where n is the number of HARQ retransmissions. Considering a typical case where there would be 0 or 1 retransmission.

B.1 ITU U-Plane latency analysis

ITU analysis the U-Plane latency for the downlink and the uplink. Each step and and its execution time is presented in Tables B.1 and B.2.

Table B.1: U-Plane latency analysis (estimated average in downlink) according TR 125 912 and S1-U Transfer Delay [51]

Step	Description	Value (30 % HARQ)
1	eNB Processing delay (S1-U -> Uu)	1 ms
2	Frame Allignment	1.022 ms
3	TTI for DL Data packet	0.675 ms
4	HARQ retransmission	0.3 * 5 ms
5	UE processing delay	1 ms
6	S1-U transfer delay and SGW/PGW	7 ms
	Total one way delay	12.2 ms

[1] Mechanism implemented to connect the error packets in the PHY layer.

Table B.2: U-Plane latency analysis (estimated average in uplink) according TR 125 912 and S1-U Transfer Delay [51]

Step	Description	Value (30 % HARQ)
0	UE wakeup time	Implementation dependent - Not included
1	UE processing delay	1 ms
2	Frame Alignment	1.423 ms
3	TTI for UL data packet	0.675
4	HARQ retransmission	0.3 * 5 ms
5	eNB processing delay (Uu -> S1-U)	1 ms
6	S1-U Transfer Delay and SGW/PGW	15 ms
	Total one way delay	20.6 ms