UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

LTE/WIFI AGGREGATION IMPLEMENTATION AND EVALUATION

A Master's Thesis
Submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by
Diego Patricio Ibarra Barreno

In partial fulfilment
of the requirements for the degree of
MASTER IN TELECOMMUNICATIONS ENGINEERING

Advisor: Dr. Ilker Demirkol

Barcelona, July 2017

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

telecom BCN

Title of the thesis : LTE/WiFi Aggregation Implementation and Evaluation

Author : Diego Patricio Ibarra Barreno

Advisor : Dr. Ilker Demirkol

Abstract

Due to fast growing of data consumed in mobile devices through cellular networks, it is necessary to think in solutions to face this challenge. One solution for this is the aggregation of mobile technologies (most commonly LTE) with wireless LAN solutions (most commonly) Wi-Fi.

In this thesis, a prototype is implemented to obtain a very tight coupling between LTE and Wi-Fi technologies, based on open source and commodity hardware. Three policies are developed for this prototype: 1) No Offload policy: data traffic is sent only over LTE, 2) Full Offload policy: data plane is sent only over Wi-Fi, and 3) Aggregation policy: data packets are sent with a transmission ratio of 50% over Wi-Fi, and 50% over LTE.

Aggregation and offloading process are managed by the eNB, therefore, the core network remains intact without any modification. The switch or split of the traffic according to the chosen policy is done in PDCP layer, which is the common layer between LTE and Wi-Fi. We evaluate in detail the network performances for all these three policies for TCP and UDP traffic and both for uplink and downlink connections.

In TCP transmissions with aggregation policy, the different delays between Wi-Fi and LTE links causes the performance degradation by the arrival of packets in a non-sequential way. For this, we evaluate a solution where an artificial delay is added to reduce the number of out-of-order packets.

Dedicated to God, and my family.

Acknowledgements

My acknowledgments to Ecuadorian people and Government of Ecuador represented by “Secretaría Nacional de Educación Superior, Ciencia y Tecnología” that through the scholarship program “Convocatoria Abierta” have trusted in capacity and talent of thousands of young Ecuadorians.

My recognition to my tutor Ilker Demirkol for his patience, guide and valuable contributions in this research process.

I would like to thank Younes Khadraoui and Xavier Lagrange for their cooperation and good predisposition to all the concerns raised.

Revision history and approval record

Revision	Date	Purpose
0	1/07/2017	Document creation
1	17/07/2017	Document revision

Written by:		Reviewed and approved by:	
Date	17/07/2017	Date	18/07/2017
Name	Diego Patricio Ibarra Barreno	Name	Ilker Demirkol
Position	Project Author	Position	Project Supervisor

Table of Contents

LIST OF FIGURES

1 Introduction

1.1 Motivation

As is shown in figure 1.1, the predicted global IP traffic, which includes fixed internet, managed IP[1] and mobile data, is going to increase exponentially in terms of exabytes consumed per month. There are several studies that provide similar estimates of annual growing IP traffic, absolute numbers could slightly differ but trends are similar to the study from Cisco described in [1].

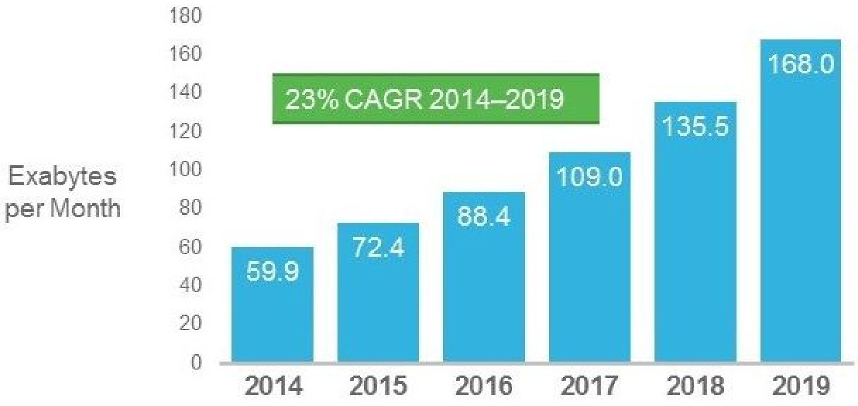


Figure 1.1 Global IP traffic growth [1]

These numbers reflect a 20% increase of traffic demand each year, this means that the Compound Annual Growth Rate (CAGR) exhibit a continuous incremental trend. In terms of absolute values, for instance, in the last year (2016) the total amount of consumed data per month was 88.4 EB. As a reference, 1EB is the unit that is equivalent to 1 billion of GB (10^9 GB) and the total world population[2] is 7 billion people, so it is possible to notice that this amount of data means that on average each person consumes 12.5 GB.

Figure 1.2 shows the fraction of this global IP traffic that is sent through mobile communication networks. In 2014, it is 2.5 EB, i.e. 4% of total IP traffic but prediction for 2019 indicates that this percentage will increase to 14%. The demand keeps increasing annually but at a faster speed, compared to global growth. In this case, the annual increase is around 50%, and doing a similar exercise (as in Global IP traffic), considering the world population, mobile traffic data in 2016 was close to 7 EB, meaning that 1GB was consumed per person on mobile communication network.

---

[1] Managed IP includes corporate IP WAN traffic and IP transport of TV and VoD.
[2] (2015, julio 29). The World Population Prospects: 2015 Revision | Latest Major
http://www.un.org/en/development/desa/publications/world-population-prospects-2015-revision.html

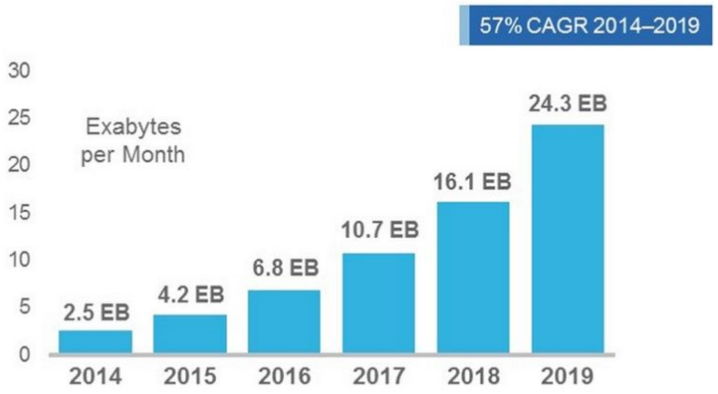


Figure 1.2 Mobile data traffic [1]

The prediction of mobile data traffic for 2019 will be 3.5 times higher than the value obtained in 2016. In this way, the expectations for 2019 are that in average it will be consumed 4GB of data with smartphones or other devices connected to mobile networks. In a study produced by Ericsson [2], the applications behind this traffic growth are shown. Figure 1.3 describes the mobile data traffic by application type, specified in Ericsson's study.

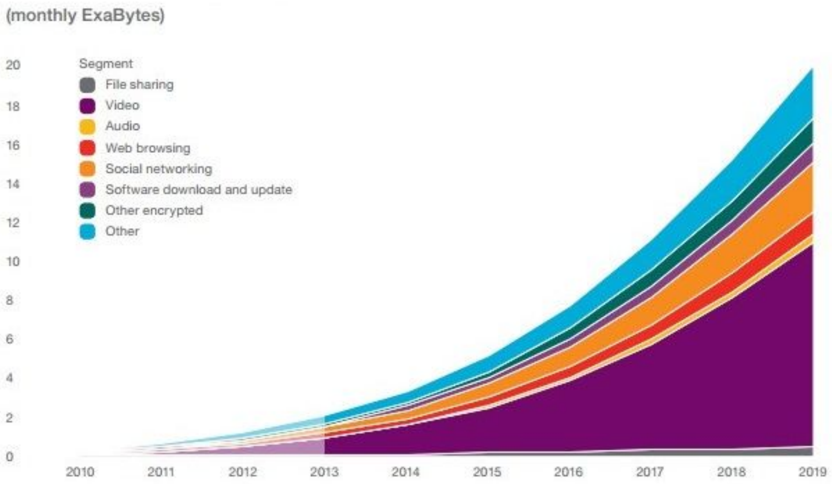


Figure 1.3 Mobile data traffic by application type [2]

Video consumption on devices connected to mobile networks represents more than half of the overall traffic. In 2019, it is estimated that 60% of all mobile data traffic will be video. Besides, another study[3] indicates that in mobile networks between 40% - 70% of video traffic is from Youtube.

The high traffic demand is challenging the evolution of 4G technology. Therefore, to satisfy all this increasing demand, it should be analyzed what must be done to increase mobile networks capacity. There are three basic enablers that can be considered: 1) installation of multi antenna system (MIMO system); 2) more spectrum, operators are continually lobbying

---

[3] "Ericsson Mobility Report November 2016." 3 nov.. 2016, https://www.ericsson.com/assets/local/mobility-report/documents/2016/ericsson-mobility-report-november-2016.pdf.

to get more spectrum from regulators, which implies significant costs; and 3) exploit the heterogeneous networks, which is achieved by different access technologies such as Wi-Fi.

An important fact to analyze is that nowadays and for the next years, most of the traffic in cellular devices, e.g. smartphones, is created when these devices are not connected to the cellular network. This is mainly because devices are also equipped with Wi-Fi interfaces and it is common that users do massive downloading when the device is connected to Wi-Fi and not to waste monthly allowed contracted data.

Figure 1.4 shows the fraction of traffic that is consumed when the devices (mobile or tablet devices that have cellular connection) are connected to the cellular network (in blue) and the traffic consumed when the device is using the Wi-Fi interface (in green). An important issue is that nowadays, approximately half of the traffic is still offloaded to Wi-Fi networks. In terminology of papers, the traffic through Wi-Fi is known as Wi-Fi offloading, and it has been a good support for cellular networks, because it has helped alleviate the congestion in these networks.

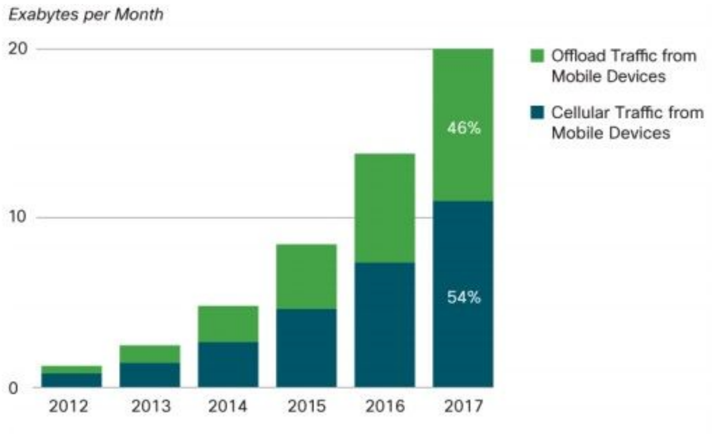


Figure 1.4 Mobile Data Traffic will be Offloaded in 2017[3]

Operators have not avoided the Wi-Fi offloading, but on the contrary, they have favored it. The interworking between cellular technologies and Wi-Fi technologies is an important point because for smartphones, the typical policy is that if the Wi-Fi network is among the preferences recorded in the device, then automatically the terminal connects to the Wi-Fi as soon as it has detected this network.

Commonly, Wi-Fi AP delivers an internet connection across a fixed network to UE and EPC can deliver an IP address because it is connected to eNB. EPC has an internet connection across another transport network (figure 1.5 a). However, LTE and Wi-Fi should work together instead of continuing a basic complementary relationship.

1.2 Research objectives and methodology

Towards the target of more efficient LTE and Wi-Fi coupling, in this thesis three principal objectives are proposed:

- Implementation of offloading traffic with Packet Data Convergence Protocol (PDCP) layer as common layer between LTE and Wi-Fi.
- Implementation of aggregation traffic with PDCP layer as common layer between LTE and Wi-Fi.
- Evaluation of performance with TCP and UDP traffic.

In this document, it is proposed an aggregation solution between LTE and Wi-Fi with PDCP layer as common layer between them. As a previous stage, Wi-Fi offloading is implemented. Both processes, aggregation and offloading, are managed by the eNB, therefore, the core network remains intact without any modification.

Wi-Fi AP should have a direct connection with eNB (figure 1.5 b), hence, in a general perspective, a Wi-Fi AP has been integrated to evolved UMTS Terrestrial Radio Access Network (E-UTRAN). The research situation implies a ENB connected to a Wi-Fi AP, and a UE able to receive traffic separately or at the same time from eNB and Wi-Fi.

Moreover, the Wi-Fi network is not using any secret key and uses an open authentication system, which allows a fast attachment.

In this work, the implementation is built on open source solutions:

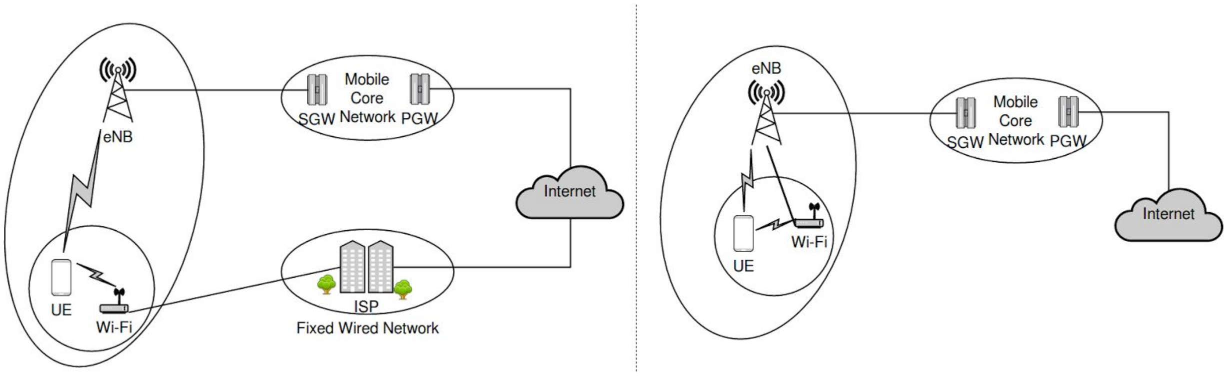
1) For LTE, the development was made on Open Air Interface code[4], together with rebuilt code of the software solution developed by Institut Mines Telecom, to provide up to date version of LTE/Wi-Fi aggregation solution. This project provides Software Defined Radio (SDR)-based software implementation for both UE and eNB, along with softwarized EPC.

2) For Wi-Fi it is used hostapd, which is a software that allows to convert a Linux device into a fully configurable Wi-Fi access point.

These two implementations, are used over commodity hardware.

---

[4] https://gitlab.eurecom.fr/oai/openairinterface5g.git



(a) (b)

Figure 1.5 a) Complementary relationship Wi-Fi/LTE b) Aggregation solution between Wi-Fi/LTE

In a more detailed vision three functionalities are implemented, for each one, a policy is implied:

1. No offload policy:

   This scenery implies a normal LTE transmission without intervention of Wi-Fi technology, the traffic follow path one of protocol stack (figure 1.6).

2. Offload policy:

   Radio bearer is switched to Wi-Fi, it means that data traffic obtained in PDCP layer is send only through Wi-Fi interface, the data traffic follow path two of protocol stack (figure 1.6). LTE still sends/receives control plane.

3. Aggregation policy:

   Radio bearer is split between lower layers of LTE and Wi-Fi , the policy defines when each interface is used. Within this thesis, a simple policy is implemented and evaluated, which sends even numbered frames through LTE and odd numbered frames through Wi-Fi. Data traffic follows Path-1 and Path-2 of protocol stack shown in Figure 1.6, alternatingly.

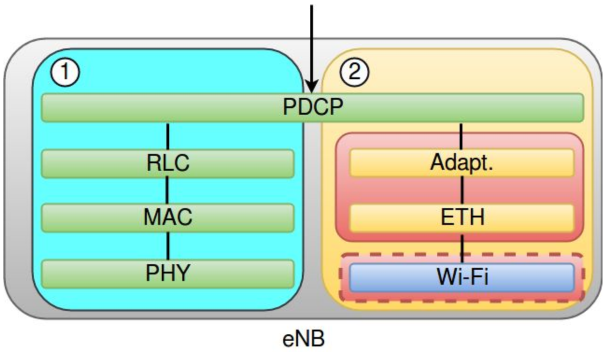


Figure 1.6 Protocol stack proposed

The protocol stack includes lower layers of both technologies LTE and Wi-Fi with PDCP layer as common layer. When IP data packet arrives to PDCP layer, a PDCP header is added obtaining a PDCP PDU. If that PDU will be sent through WiFi, an adaptation header is added which is required to recognize unequivocally each PDCP PDU, then the data packet continues to the next layer where an Ethernet header is added and finally the information is sent through physical Wi-Fi interface. This process is known as Very Tight Coupling between LTE and Wi-Fi [4].

In order to know the evaluate the developed solution, TCP and UDP traffic are generated and analyzed for uplink and downlink in three scenarios: only LTE (No offload policy), Wi-Fi offloading (offload policy), and traffic aggregation (aggregation policy).

1.3 Thesis organisation

The chapters in this thesis follow this structure:

Chapter 1 The first part of this document will concentrate on the introduction.

Chapter 2 contains principal concepts used in the research. It starts with the drivers to increase network's capacity, and continue describing important elements such as: LTE, open source LTE, wireless area network, main features of IEEE 802.11 standard.

Chapter 3 includes state of art related to LTE Wi-Fi aggregation. The proposal LWA and LWIP in release 13 of 3GPP are described.

Chapter 4 is a complete description regarding the implementation of the entire system, both hardware- and software-wise. Hardware implementation describes step by step how LTE and Wi-Fi links are defined and integrated in each machine (UE/eNB). Software implementation describes how the process is developed to obtain aggregation and offloading of traffic.

Chapter 5 involves the performance evaluations and results for each policy, measuring uplink and downlink traffic rates with TCP and UDP traffic.

Chapter 6 contains the conclusions of the research.

2 Background

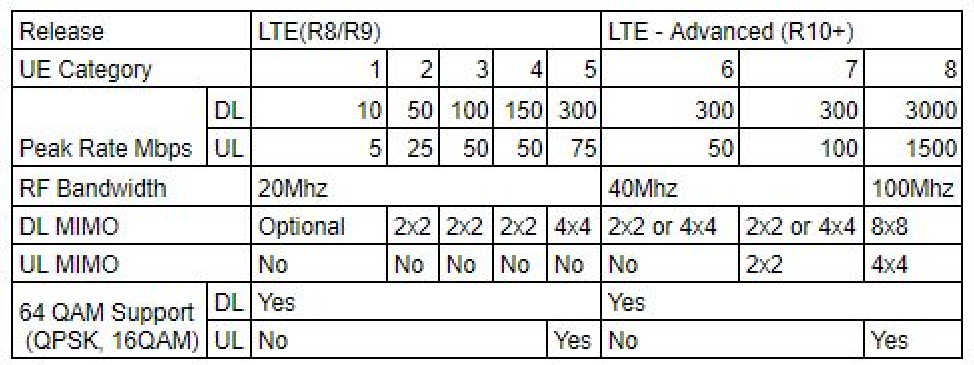
2.1 Drivers to increase available network's capacity

There are three main components to increase the mobile network capacity.

- First element implies more spectral efficiency technologies to deliver more bits per second per hertz, e.g., using MIMO and higher order modulation.
- Second element is the spectrum, with more spectrum per site, it is possible to increase capacity and,
- Third element is to have heterogeneous networks and/or network densification (more sites).

2.1.1 Improving spectral efficiency

Table 2.1 shows the majority of devices' categories specified for LTE technology. In fact, this classifies devices in terms of their capabilities. For instance, typical devices are category six, which are able to deliver 300 Mbps DL and 50 Mbps UL using 40 Mhz bandwidth and then, in terms of MIMO configuration this category admits two configurations: 2x2 MIMO or 4x4 MIMO, for smartphones the most common configuration nowadays is 2x2.



Release		LTE(R8/R9)					LTE - Advanced (R10+)		
UE Category		1	2	3	4	5	6	7	8
Peak Rate Mbps	DL	10	50	100	150	300	300	300	3000
	UL	5	25	50	50	75	50	100	1500
RF Bandwidth		20Mhz					40Mhz		100Mhz
DL MIMO		Optional	2x2	2x2	2x2	4x4	2x2 or 4x4	2x2 or 4x4	8x8
UL MIMO		No	No	No	No	No	No	2x2	4x4
64 QAM Support (QPSK, 16QAM)	DL	Yes					Yes		
	UL	No				Yes	No		Yes

Table 2.1 LTE User equipment categories

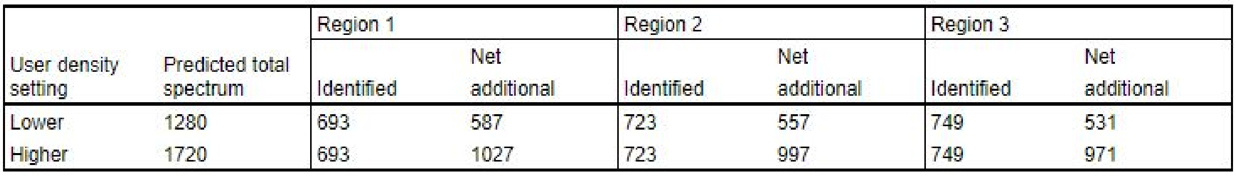
Category six devices in the downlink are able to support 64 QAM modulation, in the uplink 16QAM is the higher that can be used.

The category five was included in the first release of LTE and this category according to the configuration is able to achieve a peak spectral efficiency of 15 b/s/Hz, this spectral efficiency is obtained dividing the 300 Mbps over 20 MHz channel. Release 10 introduce category 8, which theoretically can achieve 3Gbps over 100 MHz channel with a peak spectral efficiency of 30 b/s/hz, in this case, 8x8 MIMO is needed.

The next technology advances can still improve spectral efficiency but they are very close to the Shannon limit and if traffic demand keeps exponential increasing, this might not be the path that will solve the issue of capacity.

2.1.2 ITU-R predicted spectrum requirements by 2020

According to ITU-R M.2078, the spectrum demand for the upcoming development of IMT-2000 and IMT-Advanced will be close to 1280 MHz in low market setting and 1720 MHz in a higher market settings, for 2020. This means new spectrum requirement of 500 MHz and 1 GHz, respectively, in all ITU regions as shown in Table 2.2:



		Region 1		Region 2		Region 3	
User density setting	Predicted total spectrum	Identified	Net additional	Identified	Net additional	Identified	Net additional
Lower	1280	693	587	723	557	749	531
Higher	1720	693	1027	723	997	749	971

Table 2.2 Predicted spectrum requirements by 2020 for future development of IMT-2000 and for IMT-Advanced [5]

2.1.3 Network densification: Heterogeneous Networks and small cells

To get densified networks it is necessary to split the coverage areas. Figure 2.1 shows the cell sizes gets smaller. This kind of implementations is made with the same kind of technology, i.e., in a single layer.

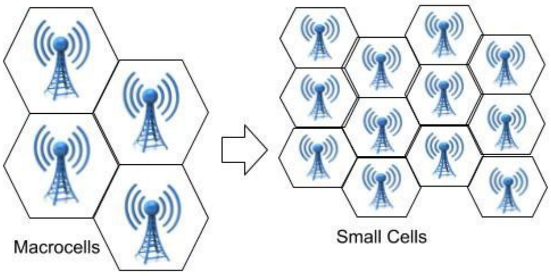


Figure 2.1. Macro cell splitting

Nowadays, the trend to provide or to have more cells that do not follow this path, consists of adding different layers of cells, this concept is called heterogeneous networks.

In heterogeneous networks, cells that provide coverage in limited places coexist inside of cells that provide coverage in large areas. This can be seen as a combination of areas covering: cell covering large areas, cell covering small areas and even the cells using different access technologies such as Wi-Fi. Hence, cells with different sizes and technologies are referred to as heterogeneous networks.

2.2 LTE

2.2.1 Evolved Packet System (EPS) architecture

The set of radio access and core network of LTE receive the name in the specification of Evolved Packet System (EPS). EPS has an access network called Evolved Universal

Terrestrial Radio Access Network (E-UTRAN) and a core network called Evolved Packet Core (EPC), these two elements form the EPS.

Long Term Evolution (LTE) was an acronym introduced at the beginning to describe the new radio interface of the 4G system developed by 3GPP. This term has remained but in specifications it is found as EPS network and this latter denomination is known as LTE network, which includes the radio part and the core part. In most of the books this terminology is used and in this document it is also adopted. The mobile terminal is named in the specifications as User Equipment (UE).

LTE network allows a user equipment to connect it to an external network through the access network and the core network. Basically, a LTE network provides a connection to an external network. It also assigns to the user equipment an IP address that is valid in the external network, so that traffic can be exchanged between the external network and the user equipment. In this context, LTE network provides the IP connectivity service.

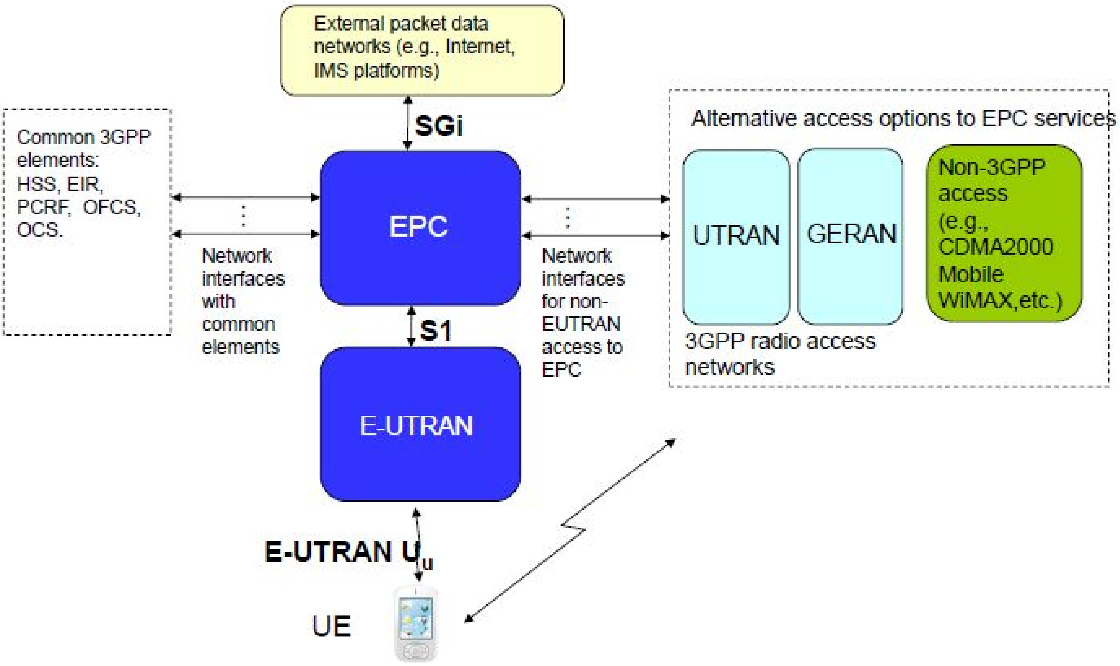


Figure 2.2 LTE Architecture [6]

E-UTRAN block and EPC block are based on blocks that are common to other access technology: Home Subscriber Server (HSS), Equipment Identity Register (EIR), Policy and Charging Rules Function (PCRF), Online Charging System (OCS), Offline Charging System (OFCS). To deploy a LTE network it is necessary to have all these elements contemplated. Figure 2.2 shows the architecture of LTE system.

The deployment of a LTE network is not done in a decoupled way from the previous technologies and have interfaces for integration with others access networks like UTRAN, GERAN. The user equipment can move between networks without any additional action done by the user, which means that there exists continuity of service. In this way, when the user starts a connection to an external network in LTE and if he moves to a HSPA network, it does not lose this connectivity. The integration with others networks is done through the core network.

These interfaces were designed to connect networks of the same family 3GPP, as well as, non-3GPP networks, allowing other technological families to converge to the use of LTE as the main mobile communication technology.

2.2.2 Access network E-UTRAN

The access network in LTE only contains a single element called evolved NodeB (eNB). The access network can be formed by multiple eNBs, where each eNB is connected to the core network through the S1 interface. This interface connected to the core network is divided into two: one for the transport of the user plane and one for the transport of the control plane. S1-U is the user plane and S1-MME is the control plane interface. Figure 2.3 shows the Access network E-UTRAN

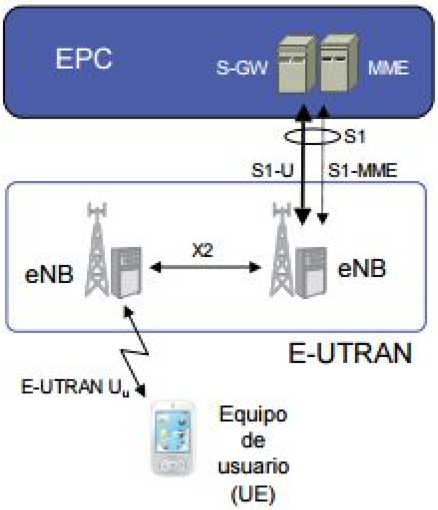


Figure 2.3 Access Network E-UTRAN [6]

LTE is a flat network, which means that there is no hierarchical structure with controller and base stations. All the functionalities of the radio interface are contained in a single element the eNB. In this way, each of these elements is autonomous and can provide access to the core network, which means that the radio functionality is completely distributed.

Another feature of LTE access network is that flat architecture allows communication between eNB via X2 interface, where the eNBs can interchange information between them. X2 interface is optional and it is not required for system operation, unlike the S1 interface, without what the system does not work.

eNB hosts the following functions[5]:

---

[5] Defined in ETSI TS 136 300

- Radio Resource Management functions: Radio bearer control, radio admission control, connection mobility control, dynamic allocation of resources to UEs in both uplink and downlink (scheduling).
- Measurement and measurement reporting configuration for mobility and scheduling.
- Access Stratum (AS), i.e. RAN communication security.
- IP header compression and encryption of user data stream.
- Selection of an MME at UE attachment when no routing to an MME can be determined from the information provided by the UE.
- Routing of User Plane data towards serving gateway.
- Scheduling and transmission of paging messages (originated from the MME).
- Scheduling and transmission of broadcast information (originated from the MME or O&M).

2.2.3 LTE radio interface: protocol stack

The functionalities of LTE access network are supported in the protocol stack represented in Figure 2.4:

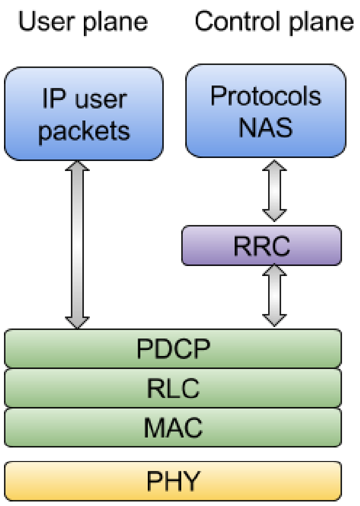


Figure 2.4 Access network, Protocol stack

The protocol stack is executed in the eNB and the UE. A user plane and a control plane are distinguished.

The user plane refers to send IP packets, there are no voice frames. The circuit switched service cannot be provided through the LTE access network, in this case access network only carries IP packets.

The control plane refers to radio resource control RRC (radio signaling) and non-access stratum (NAS) protocols. NAS protocols appear here but are not executed in the eNB. They are sent through the eNB between UE and the core network, encapsulated through the RRC protocol and finally in lower layers of the protocol structure. There is a unique physical layer and a link layer with three sublayers: a PDCP layer, an RLC layer, and a MAC layer.

2.2.3.1 Radio Resource Control protocol (RRC)

RRC is the central protocol for managing the use of the radio interface, which performs:

- Broadcast of system information related to non-access stratum and access stratum: The identification messages of the operator of the access network, and the parameters that UE uses to get into the access network is called "system information". These messages are sent by the broadcast channels.

- Establishment, maintenance and release of RRC connection: the first thing a UE does before performing any management with the network, is to establish the RRC connection, which involves establishing a communication channel between the eNB and the UE. Through this communication channel it sends messages from higher layers as NAS messages and all RRC signaling required for the establishment of radio bearers.

- Establishment, configuration, maintenance and release of signaling and data radio bearers (SRBs and DRBs).

- Security function including key management: This layer is responsible for performing security functions such as key exchange sending the encrypted information in the radio protocols, so another terminal will not be able to decode the RRC messages, the encryption function is performed by the PDCP layer but the activation control of this encryption is performed from RRC.

- Mobility functions including: control of UE cell selection/reselection, paging, UE measurement configuration and reporting, handover.

- QoS management functions.

- UE measurement reporting and control of the reporting.

2.2.3.2 Link Layer

This section will describe the details of the link layer in each of the three sublayers: PDCP, RLC, and MAC. Among the main functions, the most relevant are provided in the following.

2.2.3.2.1 Packet Data Convergence Protocol (PDCP)[6]

- Header compression and decompression using a standardized protocol named Robust Header Compression (RoHC).
- In-sequence delivery and retransmission of PDCP SDUs for Acknowledge Mode (AM) radio bearers at handover.
- Duplicate detection.
- Ciphering and deciphering.
- Integrity protection and integrity verification.

---

[6] 3GPP TS 36.323

2.2.3.2.2 Radio Link Control (RLC)[7]

- RLC handles the transfer of PDUs of upper layers, the RLC layer supports data transfer in acknowledge mode (AM), un-acknowledge mode (UM) and in transparent mode (TM). TM is equivalent to not using RLC layer. In UM a header is added with a sequence number but the retransmission mechanism is not active, being very useful to know the order of delivery of the information. In AM the sequence number is used for retransmission request in the event that an RLC message is missing.
- In acknowledge mode, error correction is applied using ARQ.
- The RLC layer is in charge of the fragmentation of the size of packages that come from the upper layers.
- Concatenation, segmentation and reassembly of RLC SDUs (only for UM and AM data transfer).
- Re-segmentation of RLC data PDUs(only for AM data transfer).
- Protocol error detection (only for AM data transfer).
- In-sequence delivery.

2.2.3.2.3 Media Access Control (MAC)[8]

- Multiplexing and demultiplexing of RLC PDUs.
- Scheduling information reporting.
- Error correction through HARQ.
- Logical Channel prioritization.
- Padding.

2.2.3.3 Physical Layer

The physical layer of LTE only defines shared channels and there are no dedicated channels. LTE defines the concept of transport channel as the service that offers the physical layer to the upper layers, so that through the same transport channel an eNB distributes information to multiple users.

2.3 Open-source LTE

There are some efforts to implement software-based on 3GPP LTE specifications, with open source: gr-LTE, srsLTE, Open Source Long-Term Evolution Deployment, Open Air Interface, and with software-license: Amarisoft LTE.

- Gr-LTE[9] : GNU Radio LTE Receiver was developed in Communication Engineering Lab at Karlsruhe Institute of Technology. The aim of this is to receive, synchronize and decode LTE signals, then Gr-LTE supplies all necessary elements for an LTE downlink receiver.

---

[7] 3GPP TS 36.322

[8] 3GPP TS 36.321

[9] https://github.com/kit-cel/gr-lte/blob/master/README.md

- srsLTE[10] : This project was developed by Software Radio System (SRS). Its environment is composed of srsUE and srsENB with whole layers from physical to IP, so that represents a step further than gr-LTE.

- Open Source Long-Term Evolution Deployment (OSLD)[11]: OSLD was developed by FlexNets group, and offers a LTE library for building base stations and mobile terminals on general purpose processors.

- Open Air Interface[12]: The nonprofit consortium OSA (OpenAirInterface Software Alliance) is responsible for the project. OAI has a complete LTE environment, it includes E-UTRAN and EPC that have interoperability with LTE commercial devices.

- Amarisoft LTE[13]: It has an important LTE ecosystem being release 13 compliant. To be used, it requires the purchase of the license.

Of the above mentioned descriptions, OAI is the most complete open source project of LTE system which can be found for the development of prototypes and academic projects.

2.3.1 Open Air Interface overview

The main goal of OAI is "to bring academia closer to complex real world systems with open source tools to ensure a common R&D and prototyping framework for rapid proof of concept designs"[14]. Open Air Interface (OAI) is an open source software, which puts into operation LTE release 10, deploying the complete protocol stack of 3GPP standards.

In E-UTRAN side, eNB has been developed entirely, and a UE also. On the other hand, EPC side is composed of: mobility management entity (MME), serving gateway (SGW) and packet data network gateway (PGW), and home subscriber server (HSS). SGW and PGW are working together in a block called S+PGW.

Open source software works over Linux computing equipment in x86 platforms with different software defined radio (SDR) front ends like: ExpressMIMO2, USRP, BladeRF, LimeSDR.

Hardware and software let to get real time radio frequency experience and an emulation environment for practical proof of concept implementations.

2.3.2 Open Air Interface usage

Open air interface is capable of being used with commercial off-the-shelf hardware, for instance: UE such as smartphones and LTE dongles, namely Huawei E392, E398u-1, Bandrich 500; eNB such as Ericsson com4Innov and commercial EPC. Figure 2.5 shows a traditional 3GPP network that is possible implement with OAI.

---

[10] https://github.com/srsLTE/srsLTE
[11] https://sites.google.com/site/osldproject/
[12] http://www.openairinterface.org/
[13] https://www.amarisoft.com/software-enb-epc-ue-simulator/
[14] http://openairinterface.eurecom.fr/

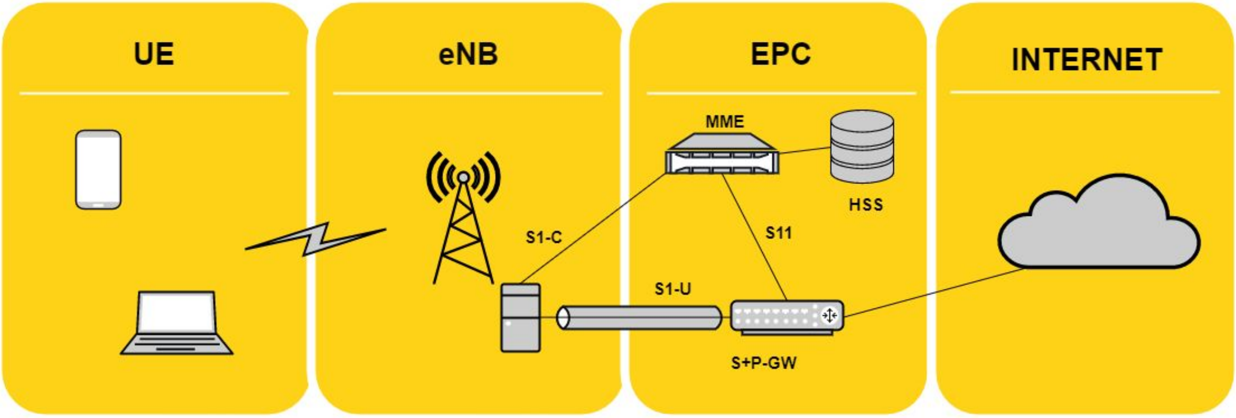


Figure 2.5 Traditional 3GPP network

Traditional 3GPP:

Open Air Interfaces has developed deployments to interact with OAI elements and commercial devices, for instance:

- Radio access network can be deployed with Open Air Interface eNB and a commercial UE, in core side Open Air Interface Evolved Packed Core is used.
- Other possible configuration in Radio access network is an Open Air Interface eNB and a commercial UE with a Commercial Evolved Packed Core.
- A commercial eNB can be configured in Radio access network with a commercial UE and an Open Air Interface Evolved Packed Core.

Simulation:

- Open Air Interface Evolved Packed Core with Open Air Interface eNB connecting an Open Air Interface UE.

Unitary simulators:

These type of implementations are useful for experimenting new code/innovations in physical functionalities.

- Downlink simulations: Downlink Shared Channel (DLSCH), Physical Downlink Shared Channel (PDSCH), Physical Downlink Control Channel (PDCCH).

- Uplink simulations: Uplink Shared Channel (ULSCH), Physical Uplink Shared Channel (PUSCH), Physical Uplink Control Channel (PUCCH).

- Other simulation: Physical Multicast Channel (PMCH), Physical Random Access Channel (PRACH), Physical Broadcast Channel (PBCH)

Other uses:

- Open Air interface can interact with Gnu/OCTAVE Interface.
- There is a simplified configuration for the case of not requiring the EPC, which does not belong to any standard 3GPP. It implies an OAI UE with an OAI eNB.

2.3.3 Open Air Interface source code organization

Source code of OAI is developed with regard to 3GPP LTE standard and it is systematized in separate folders depending of the layer implemented and its functionality. All source code is distributed through git repository. The software package contains some README documentation and scripts with helper information. It is systematized according to the following directory structure:

Openair5G:
It includes the software package for deployment of Open Air Interface radio access network.

- Openair1
  Layer 1 code that consists of all signal processing related to physical layer procedures, physical radio frequency simulation testbenches, schedules several physical functions according to the use as well as UE and eNB.

- Openair2
  Layer 2 code that contains: radio link control (RLC), medium access control (MAC), packet data convergence protocol (PDCP), radio resource control (RRC) and X2AP service.

- Openair3
  Middleware code that includes S1AP, NAS GTPV1-U for both eNB and UE.

- Common
  Common folder includes utilities that are general for all layers.

- Cmake_targets
  This folder is to build system, it means specific code for executables with everything related to configurations and compilations.

OpenairCN:
It includes the software package for implementation of Open Air Interface Evolved Packet Core.

- Script: It contains the implementation of procedures as MME, HSS, S+P-GW.
- Src: It is a folder that consists of code for GTP, NAS, and interfaces.
- Docs: This folder includes documents and user guides.
- Etc: Some configuration files.
- Test: There are scripts for the testing and performance of the system.

2.3.4 High level software architecture

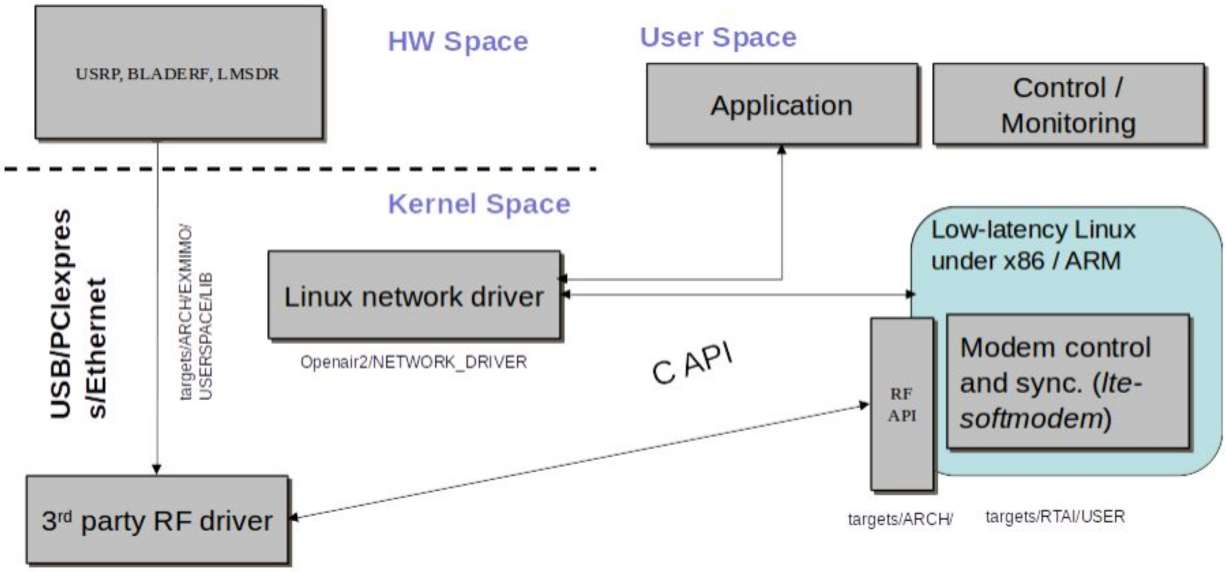


Figure 2.6 High level software architecture

Figure 2.6 shows the architecture corresponding to the open air interface system. The different interacting components are organized in three spaces, which are hardware space, kernel space, and user space.

Hardware space is the physical part responsible for transmitting/receiving radio frequency. These elements require a USB interface and can be as follows: USRP, BLADERF, and LMSDR.

In kernel space there are the RF drivers, as well as, the Linux network drivers. The RF driver has an Application Programming Interface (API) written in C language with which the driver is accessible to third-party programs. Linux network drivers are accessible to user space elements.

User space includes to application also control and monitoring elements as well as control modem and synchronization called lte-softmodem whose operation depends on a low latency linux under x86 processors, lte-softmodem through RF API interacts with RF driver and through low latency linux interacts with linux driver network.

2.3.5 Supported RF platforms

Open Air Interface can interact with some RF devices, nowadays it is possible to develop applications with the following platforms: EURECOM EXPRESSMIMO2 RF, NI/Ettus USRP B200/B210, BladeRF, and LimeSDR.

Each RF platform is connected to PC with different slots, table 2.3 shows the PC hardware requirements:

RF Device	Detail of hardware requirement
EURECOM EXPRESSMIMO2 RF	8/16-way PCIe slot
NI/Ettus USRP B200/B210	USB3 port
BladeRF	USB3 port
LimeSDR	USB3 port

Table 2.3 RF device compatible with OAI

In this research, we used the RF platform Ettus USRP B200.

2.3.5.1 USRP B200

USRP is a family of Software Defined Radio (SDR) products designed and manufactured by Ettus Research. It is a relatively inexpensive radio software platform for research laboratories, universities.

The USRP B200 is composed of blocks, which facilitate the transmission, reception and processing of the signal, as shown in Figure 2.7 of which the following blocks can be highlighted:

The USB 3.0 connector allows communication between the computer and the USRP, through this port the transfer of information at high speed is performed.

In the FPGA block, control, transport and synchronization are performed using the UHD driver that is required to work with USRP devices. It is a library written in C++ designed to work on Linux, Windows and Mac OS platforms. This driver is in charge of providing control over the products of Ettus Research.

In addition, within the FPGA block, the configured port is identified, i.e. if a transmission or reception is performed.

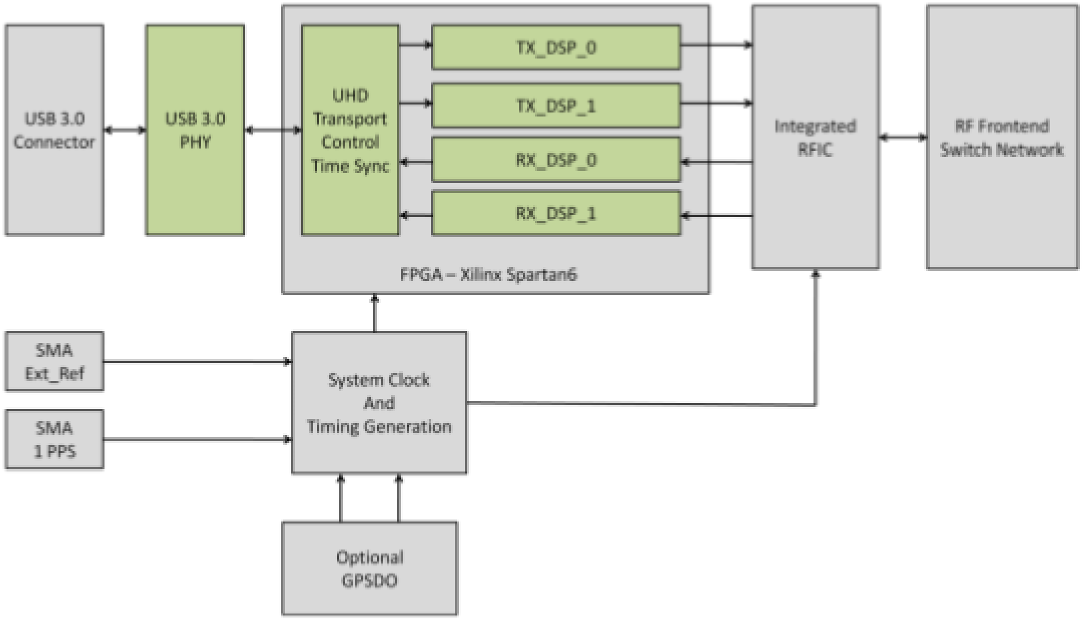


Figure 2.7 Block diagram of USRP B200[15].

2.4 Wireless local area networks

This thesis proposes the implementation of a system that coordinates the necessary actions between the cellular network and the wireless local area network for the process of aggregation and offloading of traffic, thus requiring an overview of wireless local area networks, as well as the detail of some concepts that will be used in the implementation of the system as detailed in the following.

2.4.1 Network topology

2.4.1.1 Ad hoc

Ad hoc network is a particular design because it does not have infrastructure, such as routers or access points. The communication is established through forwarding data in some nodes. The definition of which nodes forward the data is made according to the structure of the network and the routing algorithm adopted.

Ad hoc networks are typically used in short time period, where some nodes should share files and a WLAN with infrastructure is not deployed.

2.4.1.2 With infrastructure

A wireless area network with infrastructure has one or some wireless access points and nodes can transmit data through them. Usually, the wireless access point has a wired connection to establish communication to other WLAN computers.

---

[15] https://www.ettus.com/content/files/b200-b210_spec_sheet.pdf

Wireless access points are fixed, and provide wireless communication to nodes within range. Infrastructure network is scalable and allows a centralized security management.

2.4.2 Nomenclature

2.4.2.1 Access Point (AP)

A wireless access point, in a computer network is a network device that interconnects wireless communication equipment, to form a wireless network that interconnects mobile devices or wireless network cards.

2.4.2.2 Basic Service Set (BSS)

A BSS consist of all wireless nodes with a unique access point in the same range.

2.4.2.3 Distribution System (DS)

It is a network that connects the APs.

2.4.2.4 Extended Service Set (ESS):

An Extended service set is two or more interconnected basic service sets (BSSs), that appear as one BSS. It extends the range of mobility.

2.4.3 Main features of IEEE 802.11

The process of transmission developed in this project involves the Wi-Fi interface to switch or split bearers. Below we detail the main features of IEEE 802.11, IEEE 802.11b, IEEE 802.11a, IEEE 802.11g, and IEEE 802.11n.

2.4.3.1 IEEE 802.11

It was the first wireless standard published by IEEE in 1997. The transmissions are made in infrared signals with theoretical rates of 1 Mbps or 2 Mbps, over 2,4 GHz band.

The spectrum modulation technique was Frequency-Hopping spread spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS). This version is currently obsolete.

2.4.3.2 IEEE 802.11b

IEEE ratified this standard in 1999. It works with a modulation known as High Rate direct sequence spread spectrum (HR/DSSS) over 2,4 Ghz band. 802.11b has a maximum transmission speed of 11 Mbps.

To listen transmissions in the channel and to avoid collisions it uses CSMA/CA protocol, which has an important overheading. In fact, the maximum rate with this standard is near to 5.9 Mbps over TCP and 7.1 Mbps over UDP.

In this standard long and short preamble are defined.

2.4.3.3 IEEE 802.11a

Also approved by IEEE in 1999, in this standard, 52 carriers of orthogonal frequency division multiple access (orthogonal frequency division multiplexing (OFDM)) is used, with BPSK, QPSK, 16-QAM or 64 QAM modulation.

It has a maximum data rate of 54Mbps, and according to channel conditions the speeds can be adjusted to: 6, 9, 12, 18, 24, 36, 48, and 54Mbps.

The transmission band used is of 5 GHz, this makes it incompatible with 802.11b or 802.11g, and the higher frequency means shorter reach compared with counterparts that use 2.4Ghz band at the same power.

FEC coding is used with a coding rate of 1/2, 2/3, or 3/4.

2.4.3.4 IEEE 802.11g

IEEE finalized this standard in 2003. 802.11g is the “de facto” standard wireless networking protocol [7]. It is working in the same band with 802.11b over 2,4Ghz, with a different transmission technique. In this case, it uses Orthogonal Frequency Division Multiplexing (OFDM), using BPSK, QPSK, 16-QAM or 64 QAM.

For compatibility with 802.11b, there are transmissions with DSSS and maximum data rate of 11 Mbps.

FEC coding is used with a coding rate of 1/2, 2/3, or 3/4.

2.4.3.5 IEEE 802.11n

It is High Throughput (HT) standard and can achieve until 600 Mbps in both bands, 2,4 GHz or 5Ghz. The rates are obtained with more channel bandwidth, 20Mhz or 40Mhz.

Transmission techniques used is OFDM with Multiple Input Multiple Output (MIMO) system.

2.4.4 Authentication

The initial stage before getting a connection in wireless LAN is to execute 802.11 station authentication, this procedure is required every time a station attaches to a network.

The IEEE 802.11 standard describes two classes of authentication: Open system and shared key.

2.4.5 Open system authentication

Open System authentication is a null authentication algorithm [8] defined by standard.

It is a process where the access point always allows all mobile stations which are trying to get a connection in wireless LAN a successful authentication.

In the process, two frames are used, which are shown in the figure 2.8:

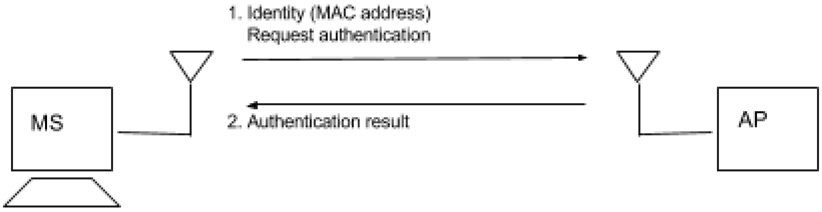


Figure 2.8 Open System Authentication

The mobile station starts the process sending a frame with identity information and requests authentication.

Identity in this case, corresponds to mobile station MAC address and must be unique in the network. Authentication requests have two elements: authentication algorithm identification whose value is 0 to denote that open system authentication is used, and authentication transaction sequence number, with 1 showing that it is the first frame.

The access point receives an authentication frame requesting open system authentication and returns the authentications result. Authentication result contains three elements: authentication algorithm identification, sequence number, and status code.

If status code is successful then mobile station will be declared mutually authenticated.

2.4.6 Shared Key authentication

Shared key authentication is a process in which mobile station and access point are manually assigned a shared key, or passphrase. The first shared key authentication was Wired Equivalent Privacy (WEP). Due to security issues Wi-Fi Protected Access (WPA) was implemented, and finally, Wi-Fi Protected Access 2 (WPA2) appeared as an improvement of WPA.

3 The State of the Art

In release 13, 3GPP has described two methods for interworking between LTE and WLAN[16]:

- LTE-WLAN Aggregation (LWA).
- LTE WLAN Radio Level Integration with IPsec Tunnel (LWIP).

In this chapter, we will detail these methods and provide other related work from the literature.

3.1 LTE/WLAN Aggregation (LWA)

LWA aggregation is specified between LTE and WLAN at Radio Access Network Level (RAN). Wi-Fi AP interacts directly with LTE eNB and EPC does not have any degree of communication with Wi-Fi. Figure 3.1 shows LWA architecture.

eNB decides about the interface to transmit data, which could be only LTE or both LTE and Wi-Fi interface,and this is determined based on the measurement reported by UE.

An important fact is that Wi-Fi was designed over unlicensed bands and LTE was designed over licensed bands. With this type of integration, the problems of fairness and regulations are avoided.

Principal elements of LWA consist of eNB, Wi-Fi AP, and UE that should be LWA-enabled. Therefore, LWA requires software updates for devices and networks, which means a relative low cost of implementation.

Depending on the scenario to be implemented, the eNB and Wi-Fi AP may be collocated or non-collocated. In the case that they are not integrated (non-collocated), the data is delivered through WLAN Termination (WT) using Xw interface.

UE receives packets over LTE and Wi-Fi interface, both are aggregated in PDCP layer. On the other hand, eNB transmits split/switch bearer. In case that data is transmitted over Wi-Fi AP, the PDCP packets are encapsulated in Wi-Fi frames. LWA works in downlink only, the communication in uplink is over LTE.

---

[16] ETSI TS 136 300 V13.2.0 (2016-01)

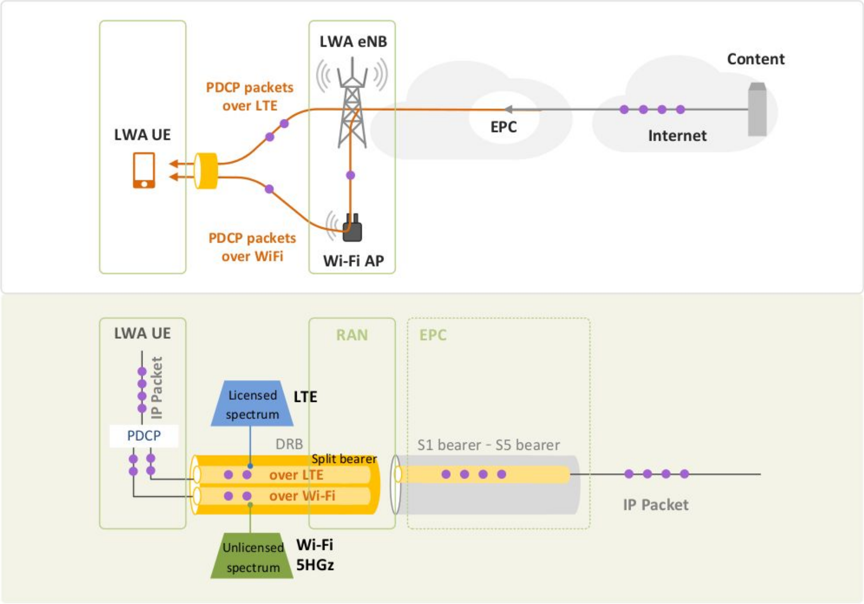


Figure 3.1 LWA Architecture [9]

3.2 LTE/WLAN Radio Level Integration with IPsec Tunnel (LWIP)

In LWIP, the principal idea is that eNB configures the UE in RRC_CONNECTED message to use (or not) WLAN through IPsec tunneling, and the eNB and WLAN communication is over IP. LWIP supports uplink and downlink data over wireless area network.

eNB in downlink and UE en uplink do not send packets at the same time through LTE and WLAN, there reordering function is not supported.

The process configuration of LWIP includes three elements: UE, eNB, and LWIP-SeGW through RRCConnectionReconfiguration message sent between UE, eNB.

LWIP overall architecture is shown in the figure 3.2.

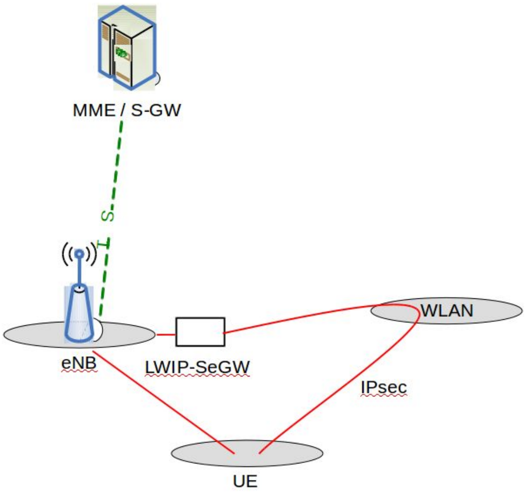


Figure 3.2 LWIP overall architecture [10]

Report made with WLAN measurements in UE is generated with WLAN identifier, backhaul rate, admission capacity, channel utilization, RSSI.

Report is used to allow three procedures: LWIP activation, inter WLAN mobility, LWIP deactivation.

In LWIP, a WLAN mobility set corresponds to all WLAN Access Points (one or more WLAN APs) distinguished by BSSID/HESSID/SSISDs, inside which a UE can carry out mobility without having to inform the eNB, always that UE is working with LWIP bearers. WLAN mobility set is configured at UE by eNB, then UE will try to perform a connection to a WLAN AP that is registered in mobility set. eNB manages UE mobility in case that WLAN APs is not registered in UE mobility set.

A UE reports measurement regarding the WLAN mobility set to eNB and eNB decides whether to update or not the WLAN mobility set. UE only can be connected to one mobility set at a time.

In figure 3.3, the LWIP protocol architecture is described.

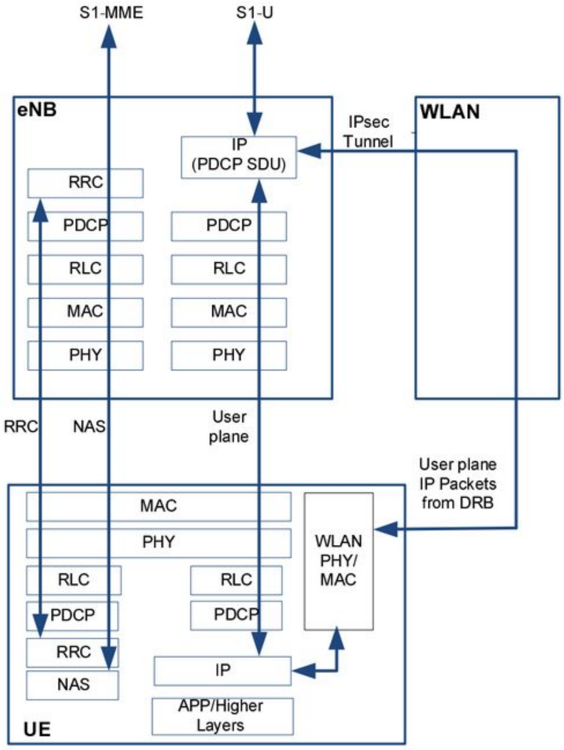


Figure 3.3 LWIP Protocol architecture [10]

RRC and NAS layer have not changed their functions. With LWIP in user plane appears a new scheme to transmit PDCP SDUs using WLAN through IPsec tunnel established between UE and eNB.

UE obtains information on whether to start the connection with the IPsec tunnel from RRCConnectionReconfiguration message to transfer DRB.

From the beginning to the end, LWIP procedure requires three stages:

- Defining how a bearer is transferred over the IPsec tunnel.
- Eliminating WLAN radio resources from DRB.
- IPsec tunnel liberation.

3.3 Related works

In [11], a study focusing on the downlink aggregation between LTE and Wi-Fi at Radio Link Control (RLC) level of eNB is proposed. Figure 3.4 shows LWIR architecture.

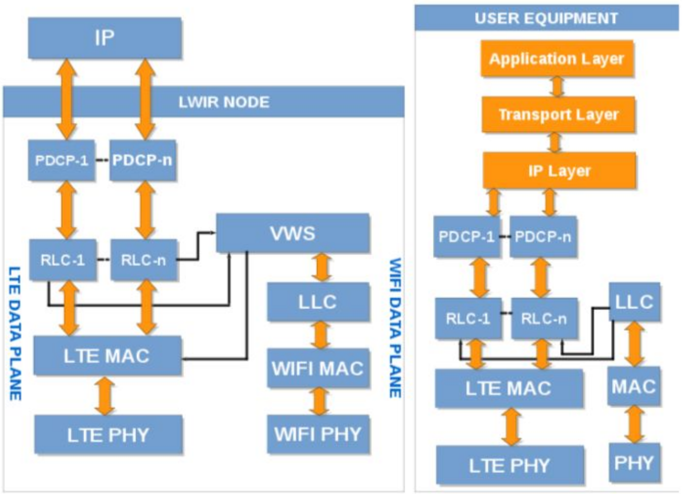


Figure 3.4 LWIR Architecture [11]

This architecture includes an additional entity named Virtual WLAN Scheduler (VWS). It manages packets in RLC queue and Wi-Fi MAC queue obtaining similar transmission times in both interfaces (Wi-Fi and LTE).

Fairness in the transmission is also taken into account, some packets will be sent over Wi-Fi and schedulers in LTE do not know this information, and the research proposes a feedback mechanism that notify the total of packets sent by Wi-Fi to LTE schedulers to guarantee fairness among users. Finally, a study of performance of the new architecture proposal with NS-3 simulator is presented.

The research [11] differs from this thesis work because it performs the aggregation of traffic in RLC layer for downlink while in our case the aggregation of traffic is done at PDCP level for downlink and uplink. In [11] a new VWS layer is defined to minimize delay between LTE and Wi-Fi interfaces, in our case the adaptation layer is defined, that has as purpose to identify the frames to be sent by the Wi-Fi interface. As for the implementation, [11] uses the NS3 simulator and in this research, it is done in a prototype based on free code and commodity hardware.

In [12], a practical implementation of integration between LTE and Wi-Fi modules at IP level is presented.

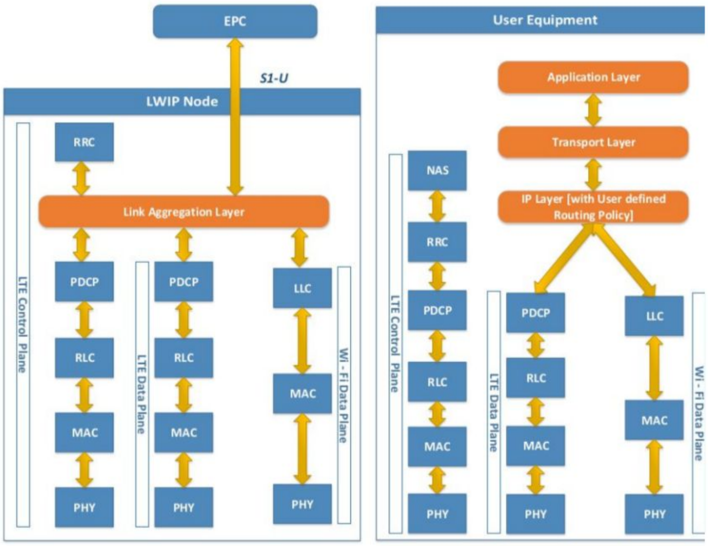


Figure 3.5 Protocol stack proposed by IITH [12]

Figure 3.5 shows the protocol stack in LWIP node and UE. LWIP node contains a new layer named Link Aggregation Layer (LAL) which is responsible of managing traffic between Wi-Fi and LTE interfaces in downlink, and it can drive the traffic at bearer/flow/packet level. LAL does not include headers to IP data packets.

Implementation was done over Open Air Interface for LTE system, hostapd is employed for Wi-Fi system, and UE receives the traffic with an Android application capable of receiving traffic on both interfaces.

The research [12] differs from this thesis work because it performs the aggregation of traffic in IP layer for downlink while in our case the aggregation of traffic is done at PDCP level for downlink and uplink. Moreover, in [12], only a single throughput evaluation is done, whereas in this thesis, we study in detail both TCP and UDP traffic and both for uplink and downlink connections.

In [13], a practical implementation of Very Tight Coupling VTC based in [4] is proposed for offload and aggregation processes. The prototype is developed over Open Air Interface framework. The proposed VTC architecture is shown in figure 3.6, where Wi-Fi AP is within the coverage area of the eNB and it is connected directly to eNB.

Offloading procedure is done for data plane, while control plane continues in LTE network, this procedure implies that Wi-Fi and LTE packets converge below IP layer.

Figure 3.6 Very Tight Coupling architecture [13]

The implementation of the research includes three components: 1) Wi-Fi integration: this component enables the interworking between LTE and Wi-Fi, 2) Selection procedure: to define in which interface PDUs are sent, and 3) PDCP reordering: used in case of splitting bearers to order the packets received.

We build on the source code developed for the solution of [13], and migrate it to the latest version of OAI, with which we can achieve much higher data rates compared to [13]. Moreover, we evaluate the performance of higher layers (TCP, UDP), analysing them in detail using different protocol parameter settings.

4 Implementation

The starting point for the development of this research is the use of two source code: the first corresponding to the repository openairinterface5g, from which the configuration of the link between OAI UE and OAI eNB without S1 interface is constructed, updated to April 2017. The second one is the source code of the development carried out in the Institut Mines Telecom [13], which includes additional functionalities not found in openairinterface5g, among the most outstanding VTC code over an OAI version of February 2016.

VTC comes with several parameter options corresponding to different functionalities, e.g. including a “network coding” functionality. Before migrating the VTC code to an up-to-date OAI version, we tried it in a setup similar to the one explained in Section 4.1, however, it was not functional after many trials. Then, we decided to migrate the VTC of the Institut Mines Telecom source code directly to the latest version of OAI.

Some functionalities that were not required for the purpose of this research of the VTC source code, e.g. “network coding”, were disabled accordingly.

4.1 Scenario description

Hardware architecture implemented is shown in figure 4.1. Two functional links must be implemented independently, the first one is an LTE link between eNB and UE, and the second is the Wi-Fi link between UE and eNB.

The eNB implementation option of “without S1 interface “ is used, which implies that processes that are performed by EPC are emulated at eNB, without an actual EPC software.

eNB is connected to a Wi-Fi AP through Ethernet network, and UE is able to receive traffic separately or at the same time from LTE and Wi-Fi interfaces.

Wi-Fi AP should have a direct connection with eNB, hence, in a general perspective, a Wi-Fi AP has been integrated to evolved UMTS Terrestrial Radio Access Network.

In order to avoid/minimize modifications on the current infrastructure, and to have something as transparent as possible, Wi-Fi AP is used in bridge mode. Thus, in AP device the Wi-Fi interface connected to the UE is bridged with the Ethernet interface connected to the eNB. With this, the Wi-Fi packets are manipulated at eNB through Ethernet captures.

Moreover, the Wi-Fi network is not using any secret key and uses an open authentication system, which allows a fast attachment.

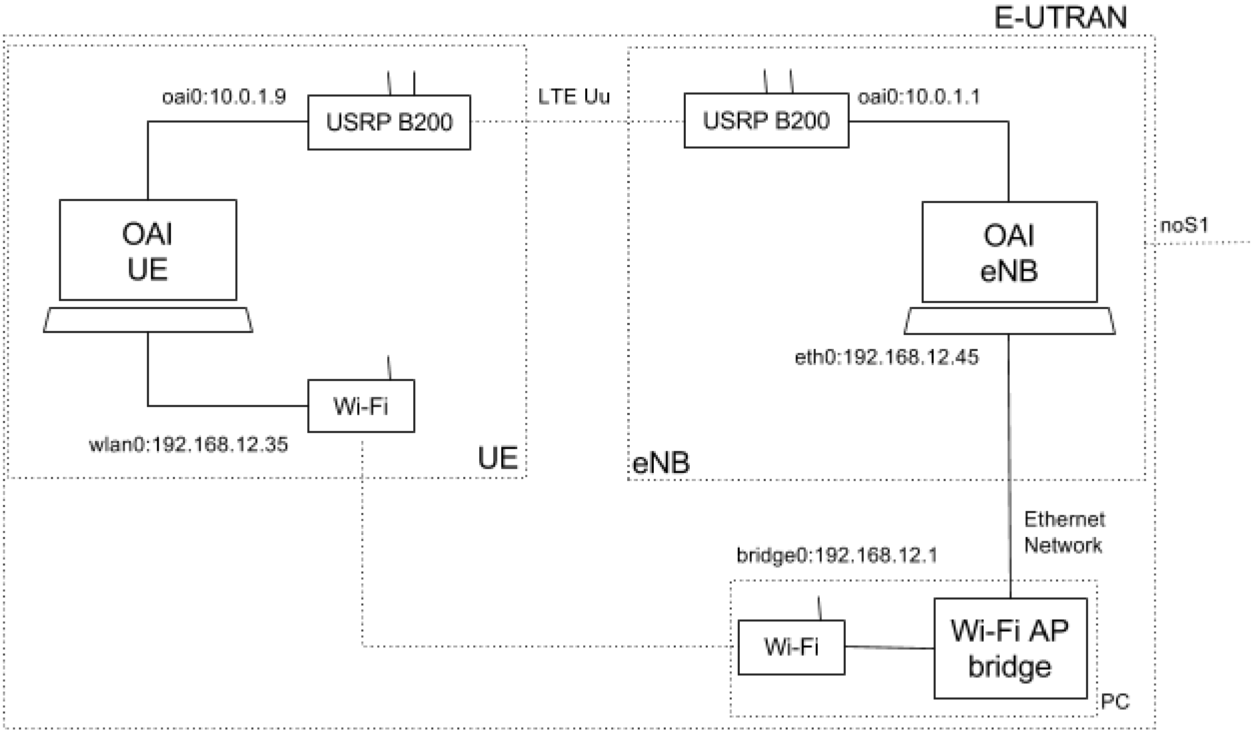


Figure 4.1 Hardware architecture implemented

4.2 Functionalities

Three functionalities are implemented, for each one, a policy is implied:

1. No Offload traffic.

> This policy implies a standard LTE transmission without intervention of Wi-Fi technology, and is named No Offload. The protocol stack is a regular on: IP data packet is sent through PDCP layer and continue to RLC, MAC, and PHY, i.e., the traffic follows Path-1 of the protocol stack shown in figure 4.2.

2. Offload traffic.

> Radio bearer is switched to Wi-Fi, which means that data traffic obtained in PDCP layer is sent only through Wi-Fi interface. The data traffic changes the regular path followed and now an adaptation layer is required to identify each PDCP PDU, as detailed later. After that the data is sent through lower layers of Wi-Fi link. The policy is named as Full Offload, following Path-2 of protocol stack shown in figure 4.2. LTE still sends/receives control plane traffic.

3. Aggregation traffic.

> Radio bearer is split between lower layers of LTE and Wi-Fi technologies. The policy implemented in this thesis is a simple one, which defines that even numbered frames are sent through LTE and odd numbered frames are sent through Wi-Fi. Data traffic follows Path-1 and Path-2 of protocol stack alternatingly (figure 4.2).

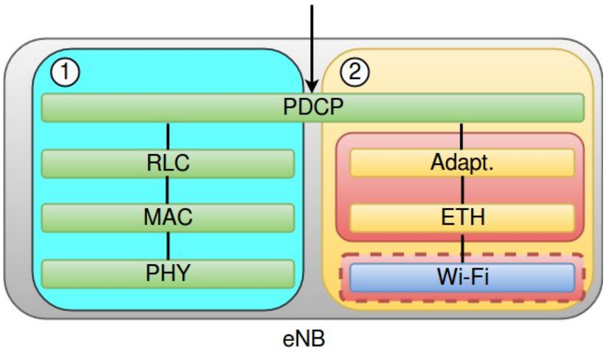


Figure 4.2 Protocol stack proposed

The protocol stack includes lower layers of both technologies LTE and Wi-Fi with PDCP layer as common layer. When IP data packet arrives to PDCP layer, a PDCP header is added obtaining a PDCP PDU. If that PDU will be sent through Wi-Fi, an adaptation header is added which is required to recognize unequivocally each PDCP PDU.

Adaptation layer adds four fields to frame: RNTI, rb_id, module_id, eNB_index. Radio Network Temporary Identifier (RNTI) is assigned when a UE has one or more active connections, it requires 2 bytes. rb_id is the radio bearer identification, it also requires 2 bytes. module_id field identifies the UE, it has 1 byte. eNB_index indicates the index of connected eNB and it has 1 byte. Figure 4.3 shows how an IP packet is transported over Wi-Fi interface.

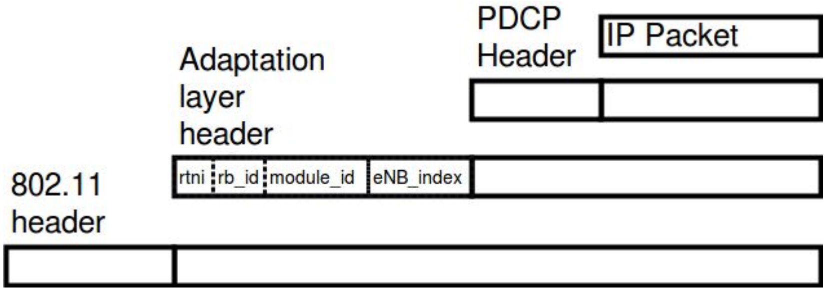


Figure 4.3 Headers of data packet in Wi-Fi transmission

After that, the data packet continues to the next layer where an Ethernet header is added with Ethernet type field “0x99ff”, then the information is sent through Ethernet network to Wi-Fi AP configured in bridge mode. With this configuration, Ethernet header is converted to Wi-Fi header and finally data is sent by physical Wi-Fi interface.

4.3 Configuration of OAI/VTC code

The operating system defines the interfaces depending on the type: Ethernet with eth and Wi-Fi with wlan. They are numbered sequentially by default, for instance: eth0, eth1, wlan0, wlan1. The interfaces to be used for the Ethernet frames must be included in the OAI code, which is done in wifi_includes.c located in openair2/WIFI in both machines UE/eNB.

Once the names of the interfaces used for sending Ethernet frames have been defined, the MAC address wlan0/eth0 of the interfaces must be included too in both machines UE/eNB. This configuration is done in openair2/WIFI/ue_wifi_threads.c in init_UE_wifi_threads() function.

The operating policy for sending the traffic on LTE or Wi-Fi to be executed by the UE/eNB is defined in openair2/WIFI/offload_config.c in the init_offload_config() function.

4.4 LTE Link

4.4.1 Overview Schematic

Figure 4.4 shows a schematic overview to link OAI UE with OAI eNB lacking of S1 interface.

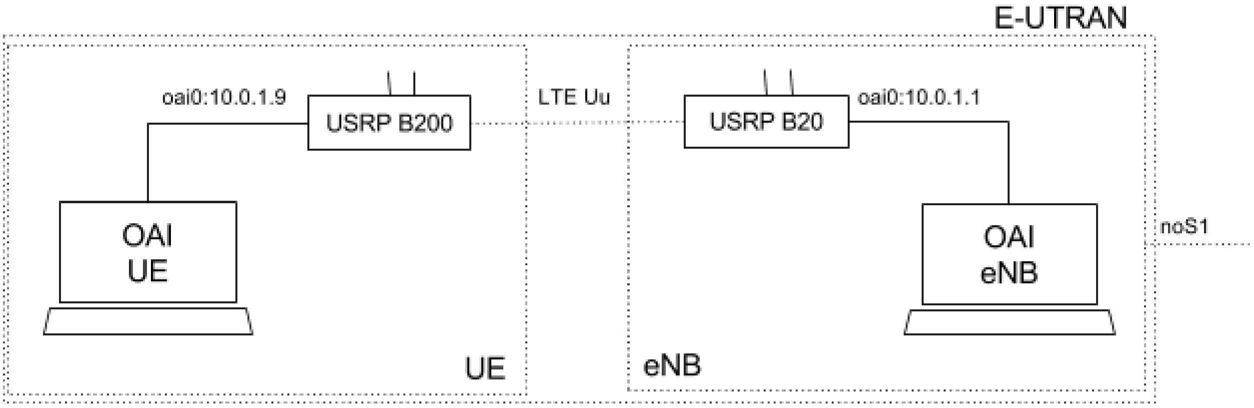


Figure 4.4 Overview schematic: LTE link

Uu interface is the goal and for its implementation the principal hardware components are:

- OAI eNB: one Ubuntu machine and one RF USRP B200.
- OAI UE: one Ubuntu machine and one RF USRP B200.

In total two Ubuntu machines and two RF USRP B200 devices are used.

For the use of Open Air Interface to LTE setup, it is required to check the hardware and software used, which is explained in the following.

4.4.2 Hardware verification

In order to implement a User Equipment or an eNB, it is required at least Intel Core i5-6600 CPU @ 3.30GHz × 4.

Intel architecture based Personal computers are used in all project OAI due to utilization of complete SIMD instructions (SSE, SSE2, SSE3, and SSE4).

With regard to support RF, this project uses USRP B200 and it demands a free USB3 port in PC.

4.4.3 Operating system verification

PCs should have installed Ubuntu LTS 14.04.3 (64bits) with low latency Kernel version 3.19, if kernel installed is of a different version it can be updated with:

```
$ sudo apt-get install linux-image-3.19.0-61-lowlatency linux-headers-3.19.0-61-lowlatency
```

In order to take effect the new kernel, PC should be rebooted, and to confirm it is typed:

```
$ uname -a
Linux "NAME" 3.19.0-61-lowlatency #69~14.04.1-Ubuntu SMP PREEMPT Thu Jun 9 10:15:00 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux.
```

Due to real time communications, it is necessary to disable power management features in the BIOS as p-states, c-states and CPU frequency control. If PC has available hyperthreading option, it must also be disabled.

In Linux, c-states are removed in /etc/default/grub, the parameter modified is GRUB_CMDLINE_LINUX_DEFAULT= , like this:

```
$ sudo vi /etc/default/grub
GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_pstate=disable processor.max_cstate=1 intel_idle.max_cstate=0 idle=poll"
```

Finally, changes should be applied:

```
$ update-grub
```

There exists a module called intel_powerclamp, which is a power management module and should be disabled. Then, the file /etc/modprobe.d/blacklist.conf will be added:

```
$ sudo vi /etc/modprobe.d/blacklist.conf
blacklist intel_powerclamp
```

CPU frequency should be constant, and for that, cpufrequtils is required. In file /etc/default/cpufrequtils a new parameter will be added:

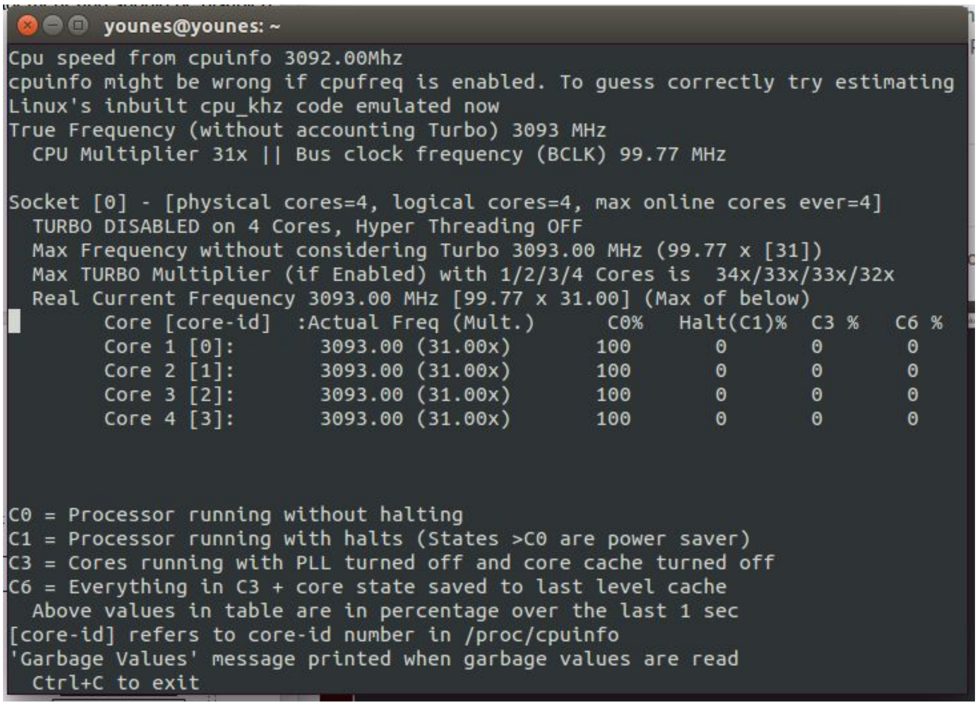
```
$ sudo apt-get install cpufrequtils
$ sudo vi /etc/default/cpufrequtils
GOVERNOR="performance"
```

Finally, on demand daemon should be disabled, otherwise, the next time that PC be turned on, CPU frequency scaling will be available.

```
$ sudo update-rc.d ondemand disable
```

Once power management features has been disabled, i7z tool is used to verify that CPU does not change its frequency, and just C0 state remains available. Figure 4.5 shows the result obtained with i7z tool.

```
$ sudo i7z
```



```
Cpu speed from cpuinfo 3092.00Mhz
cpuinfo might be wrong if cpufreq is enabled. To guess correctly try estimating
Linux's inbuilt cpu_khz code emulated now
True Frequency (without accounting Turbo) 3093 MHz
  CPU Multiplier 31x || Bus clock frequency (BCLK) 99.77 MHz

Socket [0] - [physical cores=4, logical cores=4, max online cores ever=4]
  TURBO DISABLED on 4 Cores, Hyper Threading OFF
  Max Frequency without considering Turbo 3093.00 MHz (99.77 x [31])
  Max TURBO Multiplier (if Enabled) with 1/2/3/4 Cores is  34x/33x/33x/32x
  Real Current Frequency 3093.00 MHz [99.77 x 31.00] (Max of below)
        Core [core-id]  :Actual Freq (Mult.)      C0%   Halt(C1)%  C3 %   C6 %
        Core 1 [0]:       3093.00 (31.00x)       100       0       0       0
        Core 2 [1]:       3093.00 (31.00x)       100       0       0       0
        Core 3 [2]:       3093.00 (31.00x)       100       0       0       0
        Core 4 [3]:       3093.00 (31.00x)       100       0       0       0



C0 = Processor running without halting
C1 = Processor running with halts (States >C0 are power saver)
C3 = Cores running with PLL turned off and core cache turned off
C6 = Everything in C3 + core state saved to last level cache
  Above values in table are in percentage over the last 1 sec
[core-id] refers to core-id number in /proc/cpuinfo
'Garbage Values' message printed when garbage values are read
  Ctrl+C to exit
```

Figure 4.5 Power management features disabled

4.4.4 Get repository

In order to obtain the repository for UE/eNB, a version control software called git should be installed.

```
$ sudo apt-get update
```

```
$sudo apt-get install subversion git
```

Before obtaining the repository, it is necessary to implement a generic SSL client which can establish a transparent connection to a gitlab.eurecom.fr server speaking SSL, this can be done using the next command:

```
$ echo -n | openssl s_client -showcerts -connect gitlab.eurecom.fr:443 2>/dev/null | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' | sudo tee -a /etc/ssl/certs/ca-certificates.crt
```

Some repositories are posted in EURECOM gitlab. The interesting one is openairinterface5g, because it contains source code for UE/eNB RAN.

```
$ cd ~/
$ git clone https://gitlab.eurecom.fr/oai/openairinterface5g.git
```

Finally, it is necessary to run just one time to install missing packages.

```
$ cd ~/openairinterface5g/
```

```
$ cd cmake_targets
$ ./build_oai -I -w USRP
```

4.4.5 Setup eNB and UE

Previous verifications, both of hardware and software such those described in section 4.1.2 and 4.1.3, are a prerequisite for continuing the installation process in UE/eNB Ubuntu machines.

Repository openairinterface5g includes scripts to build UE/eNB in various scenarios, scripts must be run without sudo.

4.4.5.1 OAI eNB without S1 interface

Inside folder openairinterface5g/cmake_targets, there is the script build_oai to build eNB without S1 interface.

In oaienv file, some variables regard to actual working directory are contained.

```
$ cd ~/openairinterface5g/
$ source oaienv
$ cd cmake_targets
$ ./build_oai -I -w USRP
$ ./build_oai -w USRP --eNB --noS1 -x
```

Parameters used in build_oai are:

-w USRP	parameter -w defines RF hardware to be utilized.
--eNB	parameter --eNB is used to make the LTE softmodem.
--noS1	parameter --noS1 is defined to compile eNB without S1 interface.
-x	value -x will generate the software oscilloscope features.

The result of the previous procedure is in ~/openairinterface5g/cmake_targets/lte_noS1_build-oai.

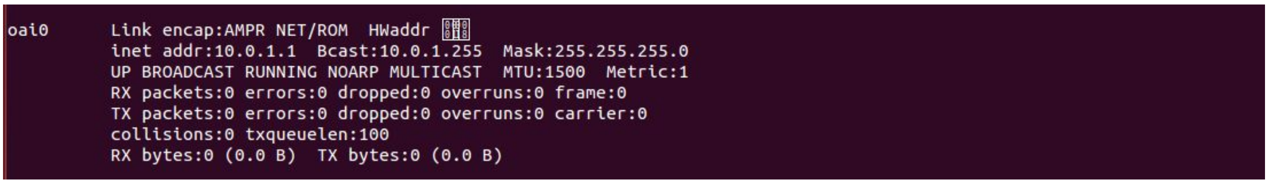
eNB machine requires an IP interface to support the radio bearers, for that, nasmesh kernel module should be loaded.

```
cd ~/openairinterface5g/
source oaienv
source ./cmake_targets/tools/init_nas_nos1 eNB
```

If nasmesh module is loaded due to a previous process, it will be removed and loaded again.

A new interface called “oai0” with IP 10.0.1.1 and netmask 255.255.255.0 is created, which can be shown with ifconfig command (see figure 4.6).

```
$ ifconfig
```



```
oai0      Link encap:AMPR NET/ROM  HWaddr
          inet addr:10.0.1.1  Bcast:10.0.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Figure 4.6 oai0 Interface in eNB

The carrier frequency needs to be configured before running eNB, it is a parameter in the eNB config file as shown below. Any LTE band can be used, however, we used 2.660 GHz as Downlink frequency:

```
$ cat $OPENAIR_TARGETS/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band7.tm1.usrpb210.conf
downlink _frequency                          = 2660000000L
```

eNB is ready to run, the command is described in the following lines:

```
cd cmake_targets
sudo -E ./lte_noS1_build_oai/build/lte-softmodem-nos1 -d -O $OPENAIR_TARGETS/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band7.tm1.usrpb210.conf 2>&1 | tee ENB.log
```

-O parameter defines the path to configuration file of eNB, and -d parameter enables soft scope and L1 and L2 stats (Xforms).

ENB.log is a file created in folder cmake_targets which saves information of link connection.

4.4.5.2 OAI UE without S1 interface

Scripts to build UE without S1 interface are inside folder openairterface5g/cmake_targegts.

In order to start the process, some variables are defined regard to the actual working directory in oaienv file.

```
cd ~/openairinterface5g
source oaienv
cd cmake_targets
./build_oai -w USRP --eNB --UE --noS1 -x
```

Parameters used in build_oai are :

-w USRP	parameter -w defines RF hardware to be utilized
--eNB	parameter --eNB is used to make the LTE softmodem
--UE	parameter --UE makes the UE specific parts
--noS1	parameter --noS1 is defined to compile eNB without S1 interface
-x	value -x will generate the software oscilloscope features

The result of the previous procedure is in ~/openairinterface5g/cmake_targets/lte_noS1_build-oai.

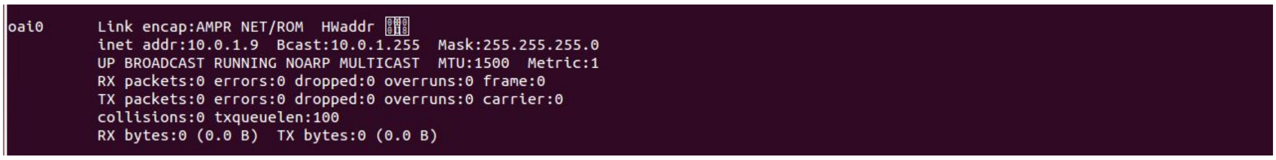
UE machine requires an IP interface to support the radio bearer, for that nasmesh kernel module should be loaded.

```
cd ~/openairinterface5g/
source ./targets/bin/init_nas_nos1 UE
```

If nasmesh module is loaded due to a previous process, it will be removed and loaded again.

A new interface, called “oai0”, is created with IP 10.0.1.9 and netmask 255.255.255.0 (see figure 4.7), this one is shown with ifconfig command.

```
$ ifconfig
```



```
oai0      Link encap:AMPR NET/ROM  HWaddr
          inet addr:10.0.1.9  Bcast:10.0.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Figure 4.7 oai0 Interface in UE

UE is ready to run, the command is described in following lines:

```
cd cmake_targets
sudo -E ./lte_noS1_build_oai/build/lte-softmodem-nos1 -U -C2660000000 -r25
--ue-scan-carrier --ue-txgain 90 --ue-rxgain 115 -d >&1 | tee UE.log
```

The parameters used are the following:

-U	Set the lte softmodem as a UE
-C	Set the downlink frequency for all component carriers
-r	Set the PRB, valid values: 6, 25, 50, 100
-d	Enable soft scope and L1 and L2 stats (Xforms)
--ue-scan_carrier	set UE to scan around carrier
--ue-rxgain	set UE RX gain
--ue-txgain	set UE TX gain

UE.log is a file created in folder cmake_targets which saves information of the link connection.

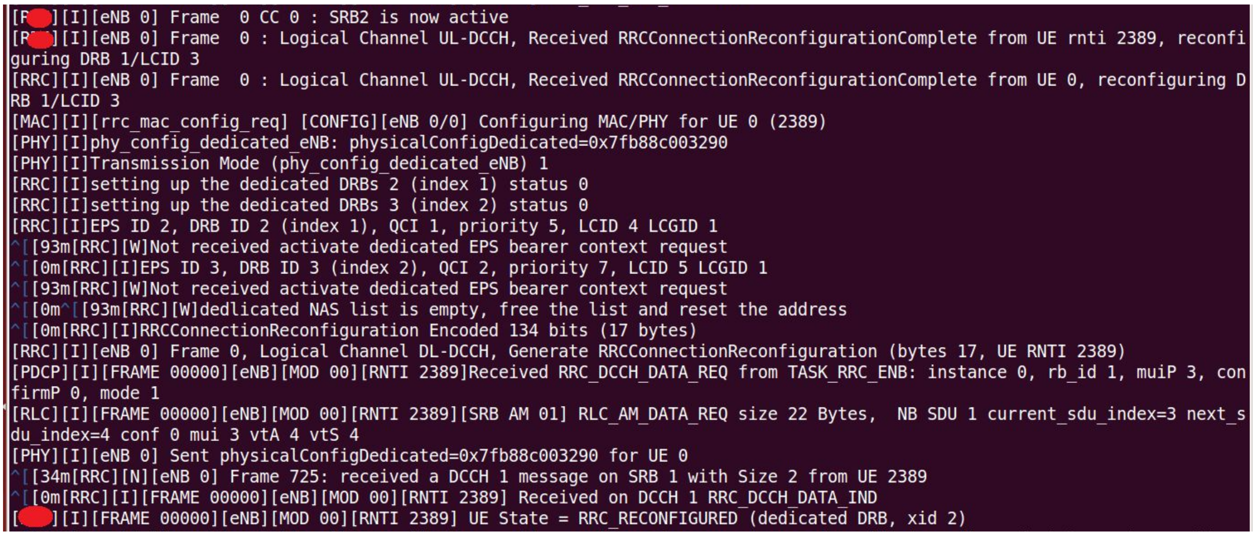
4.4.6 Link confirmation

UE.log file can be used to successfully recognize attachment between UE and eNB, UE sends RRCConnectionReconfigurationComplete message and UE state is RRC_RECONFIGURED, as shown in figure 4.8.

```
[RRC][I][rrc_ue_generate_RRCConnectionReconfigurationComplete] [FRAME 00724][ UE][MOD 00][RNTI 2389] Logical Channel UL-DCCH
 (SRB1), Generating RRCConnectionReconfigurationComplete (bytes 2, eNB_index 0)
[RRC][I][rrc_ue_decode_dcch] [UE 0] State = RRC_RECONFIGURED (eNB 0)
```

Figure 4.8 UE.log file with information that confirm a successful link

In ENB.log file also is present information about the RRCConnectionReconfigurationComplete message sent by UE, and UE state should be RRC_RECONFIGURED, as shown and highlighted in figure 4.9, which shows a successful link.



```
[R[redacted]][I][eNB 0] Frame  0 CC 0 : SRB2 is now active
[R[redacted]][I][eNB 0] Frame  0 : Logical Channel UL-DCCH, Received RRCConnectionReconfigurationComplete from UE rnti 2389, reconfi
guring DRB 1/LCID 3
[RRC][I][eNB 0] Frame  0 : Logical Channel UL-DCCH, Received RRCConnectionReconfigurationComplete from UE 0, reconfiguring D
RB 1/LCID 3
[MAC][I][rrc_mac_config_req] [CONFIG][eNB 0/0] Configuring MAC/PHY for UE 0 (2389)
[PHY][I]phy_config_dedicated_eNB: physicalConfigDedicated=0x7fb88c003290
[PHY][I]Transmission Mode (phy_config_dedicated_eNB) 1
[RRC][I]setting up the dedicated DRBs 2 (index 1) status 0
[RRC][I]setting up the dedicated DRBs 3 (index 2) status 0
[RRC][I]EPS ID 2, DRB ID 2 (index 1), QCI 1, priority 5, LCID 4 LCGID 1
^[[93m[RRC][W]Not received activate dedicated EPS bearer context request
^[[0m[RRC][I]EPS ID 3, DRB ID 3 (index 2), QCI 2, priority 7, LCID 5 LCGID 1
^[[93m[RRC][W]Not received activate dedicated EPS bearer context request
^[[0m^[[93m[RRC][W]dedlicated NAS list is empty, free the list and reset the address
^[[0m[RRC][I]RRCConnectionReconfiguration Encoded 134 bits (17 bytes)
[RRC][I][eNB 0] Frame 0, Logical Channel DL-DCCH, Generate RRCConnectionReconfiguration (bytes 17, UE RNTI 2389)
[PDCP][I][FRAME 00000][eNB][MOD 00][RNTI 2389]Received RRC_DCCH_DATA_REQ from TASK_RRC_ENB: instance 0, rb_id 1, muiP 3, con
firmP 0, mode 1
[RLC][I][FRAME 00000][eNB][MOD 00][RNTI 2389][SRB AM 01] RLC_AM_DATA_REQ size 22 Bytes,  NB SDU 1 current_sdu_index=3 next_s
du_index=4 conf 0 mui 3 vtA 4 vtS 4
[PHY][I][eNB 0] Sent physicalConfigDedicated=0x7fb88c003290 for UE 0
^[[34m[RRC][N][eNB 0] Frame 725: received a DCCH 1 message on SRB 1 with Size 2 from UE 2389
^[[0m[RRC][I][FRAME 00000][eNB][MOD 00][RNTI 2389] Received on DCCH 1 RRC_DCCH_DATA_IND
[[redacted]][I][FRAME 00000][eNB][MOD 00][RNTI 2389] UE State = RRC_RECONFIGURED (dedicated DRB, xid 2)
```

Figure 4.9 ENB.log file with information that confirm a successful link

eNB shows a software oscilloscope, some general statistics and Layer 2 statistics. Figure 4.10 shows UE and eNB having traffic between them.

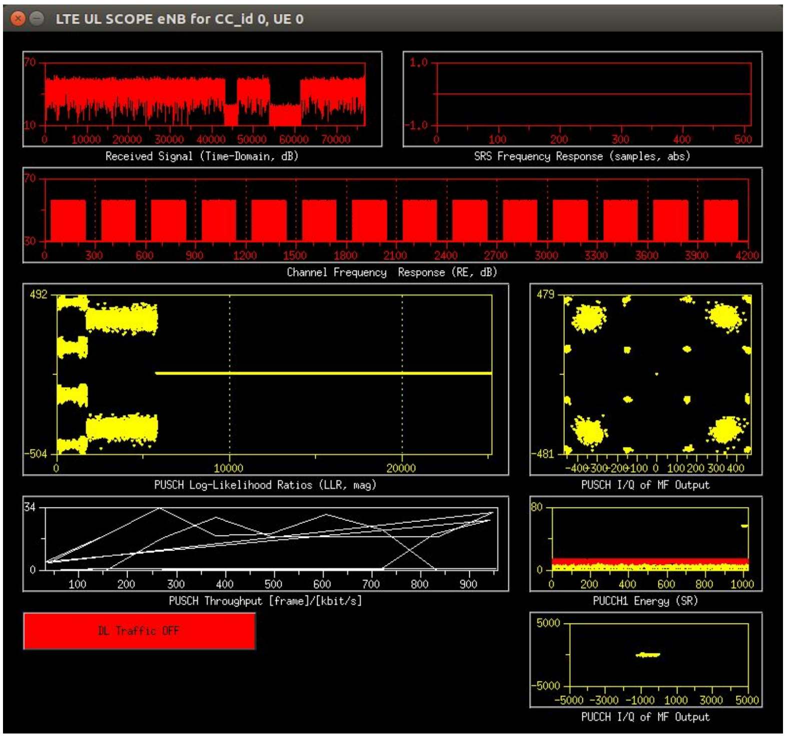
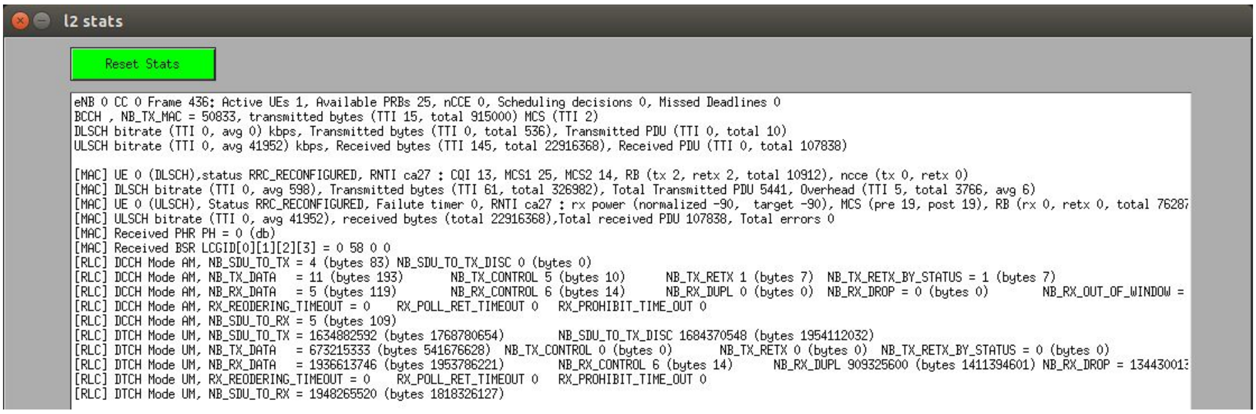


Figure 4.10 LTE UL scope eNB

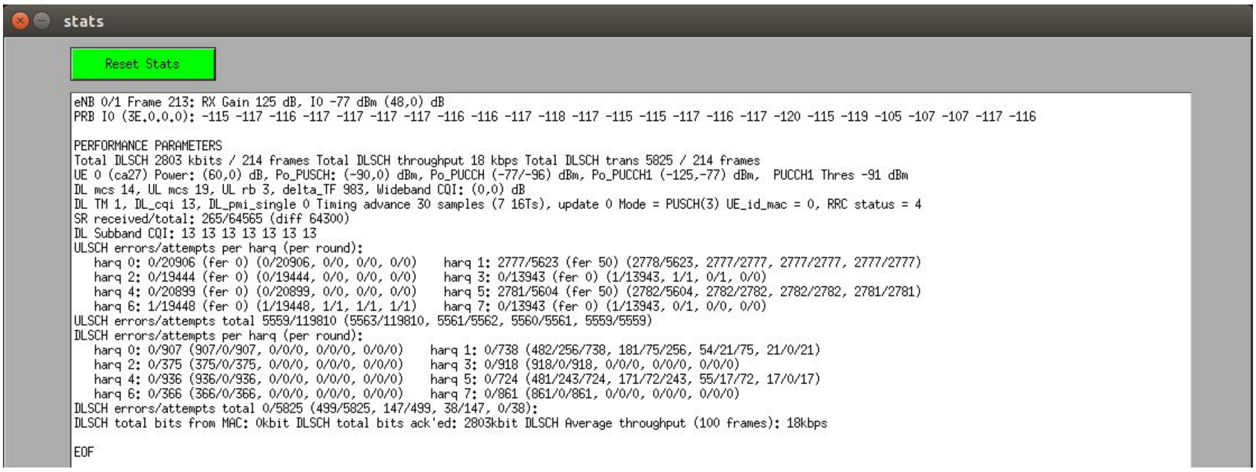
Figures 4.11 and 4.12 shows general statistics and Layer 2 statistics.



```
eNB 0 CC 0 Frame 436: Active UEs 1, Available PRBs 25, nCCE 0, Scheduling decisions 0, Missed Deadlines 0
BCCH , NB_TX_MAC = 50833, transmitted bytes (TTI 15, total 915000) MCS (TTI 2)
DLSCH bitrate (TTI 0, avg 0) kbps, Transmitted bytes (TTI 0, total 536), Transmitted PDU (TTI 0, total 10)
ULSCH bitrate (TTI 0, avg 41952) kbps, Received bytes (TTI 145, total 22916368), Received PDU (TTI 0, total 107838)

[MAC] UE 0 (DLSCH),status RRC_RECONFIGURED, RNTI ca27 : CQI 13, MCS1 25, MCS2 14, RB (tx 2, retx 2, total 10912), ncce (tx 0, retx 0)
[MAC] DLSCH bitrate (TTI 0, avg 598), Transmitted bytes (TTI 61, total 326982), Total Transmitted PDU 5441, Overhead (TTI 5, total 3766, avg 6)
[MAC] UE 0 (ULSCH), Status RRC_RECONFIGURED, Failute timer 0, RNTI ca27 : rx power (normalized -90,  target -90), MCS (pre 19, post 19), RB (rx 0, retx 0, total 76287
[MAC] ULSCH bitrate (TTI 0, avg 41952), received bytes (total 22916368),Total received PDU 107838, Total errors 0
[MAC] Received PHR PH = 0 (db)
[MAC] Received BSR LCGID[0][1][2][3] = 0 58 0 0
[RLC] DCCH Mode AM, NB_SDU_TO_TX = 4 (bytes 83) NB_SDU_TO_TX_DISC 0 (bytes 0)
[RLC] DCCH Mode AM, NB_TX_DATA   = 11 (bytes 193)       NB_TX_CONTROL 5 (bytes 10)      NB_TX_RETX 1 (bytes 7)  NB_TX_RETX_BY_STATUS = 1 (bytes 7)
[RLC] DCCH Mode AM, NB_RX_DATA   = 5 (bytes 119)        NB_RX_CONTROL 6 (bytes 14)      NB_RX_DUPL 0 (bytes 0)  NB_RX_DROP = 0 (bytes 0)        NB_RX_OUT_OF_WINDOW =
[RLC] DCCH Mode AM, RX_REODERING_TIMEOUT = 0    RX_POLL_RET_TIMEOUT 0   RX_PROHIBIT_TIME_OUT 0
[RLC] DCCH Mode AM, NB_SDU_TO_RX = 5 (bytes 109)
[RLC] DTCH Mode UM, NB_SDU_TO_TX = 1634882592 (bytes 1768780654)        NB_SDU_TO_TX_DISC 1684370548 (bytes 1954112032)
[RLC] DTCH Mode UM, NB_TX_DATA   = 673215333 (bytes 541676628)  NB_TX_CONTROL 0 (bytes 0)       NB_TX_RETX 0 (bytes 0)  NB_TX_RETX_BY_STATUS = 0 (bytes 0)
[RLC] DTCH Mode UM, NB_RX_DATA   = 1936613746 (bytes 1953786221)        NB_RX_CONTROL 6 (bytes 14)      NB_RX_DUPL 909325600 (bytes 1411394601) NB_RX_DROP = 134430013
[RLC] DTCH Mode UM, RX_REODERING_TIMEOUT = 0    RX_POLL_RET_TIMEOUT 0   RX_PROHIBIT_TIME_OUT 0
[RLC] DTCH Mode UM, NB_SDU_TO_RX = 1948265520 (bytes 1818326127)
```

Figure 4.11 L2 stats



```
eNB 0/1 Frame 213: RX Gain 125 dB, I0 -77 dBm (48,0) dB
PRB I0 (3E.0.0.0): -115 -117 -116 -117 -117 -117 -117 -116 -116 -117 -118 -117 -115 -115 -117 -116 -117 -120 -115 -119 -105 -107 -107 -117 -116

PERFORMANCE PARAMETERS
Total DLSCH 2803 kbits / 214 frames Total DLSCH throughput 18 kbps Total DLSCH trans 5825 / 214 frames
UE 0 (ca27) Power: (60,0) dB, Po_PUSCH: (-90,0) dBm, Po_PUCCH (-77/-96) dBm, Po_PUCCH1 (-125,-77) dBm,  PUCCH1 Thres -91 dBm
DL mcs 14, UL mcs 19, UL rb 3, delta_TF 983, Wideband CQI: (0,0) dB
DL TM 1, DL_cqi 13, DL_pmi_single 0 Timing advance 30 samples (7 16Ts), update 0 Mode = PUSCH(3) UE_id_mac = 0, RRC status = 4
SR received/total: 265/64565 (diff 64300)
DL Subband CQI: 13 13 13 13 13 13 13
ULSCH errors/attempts per harq (per round):
   harq 0: 0/20906 (fer 0) (0/20906, 0/0, 0/0, 0/0)    harq 1: 2777/5623 (fer 50) (2778/5623, 2777/2777, 2777/2777, 2777/2777)
   harq 2: 0/19444 (fer 0) (0/19444, 0/0, 0/0, 0/0)    harq 3: 0/13943 (fer 0) (1/13943, 1/1, 0/1, 0/0)
   harq 4: 0/20899 (fer 0) (0/20899, 0/0, 0/0, 0/0)    harq 5: 2781/5604 (fer 50) (2782/5604, 2782/2782, 2782/2782, 2781/2781)
   harq 6: 1/19448 (fer 0) (1/19448, 1/1, 1/1, 1/1)    harq 7: 0/13943 (fer 0) (1/13943, 0/1, 0/0, 0/0)
ULSCH errors/attempts total 5559/119810 (5563/119810, 5561/5562, 5560/5561, 5559/5559)
DLSCH errors/attempts per harq (per round):
   harq 0: 0/907 (907/0/907, 0/0/0, 0/0/0, 0/0/0)    harq 1: 0/738 (482/256/738, 181/75/256, 54/21/75, 21/0/21)
   harq 2: 0/375 (375/0/375, 0/0/0, 0/0/0, 0/0/0)    harq 3: 0/918 (918/0/918, 0/0/0, 0/0/0, 0/0/0)
   harq 4: 0/936 (936/0/936, 0/0/0, 0/0/0, 0/0/0)    harq 5: 0/724 (481/243/724, 171/72/243, 55/17/72, 17/0/17)
   harq 6: 0/366 (366/0/366, 0/0/0, 0/0/0, 0/0/0)    harq 7: 0/861 (861/0/861, 0/0/0, 0/0/0, 0/0/0)
DLSCH errors/attempts total 0/5825 (499/5825, 147/499, 38/147, 0/38):
DLSCH total bits from MAC: 0kbit DLSCH total bits ack'ed: 2803kbit DLSCH Average throughput (100 frames): 18kbps

EOF
```

Figure 4.12 General Stats

On the other side, UE shows LTE DL scope and statistics. Figure 4.13 shows how it looks like when there is traffic between UE and eNB, and figure 4.14 shows some statistics.

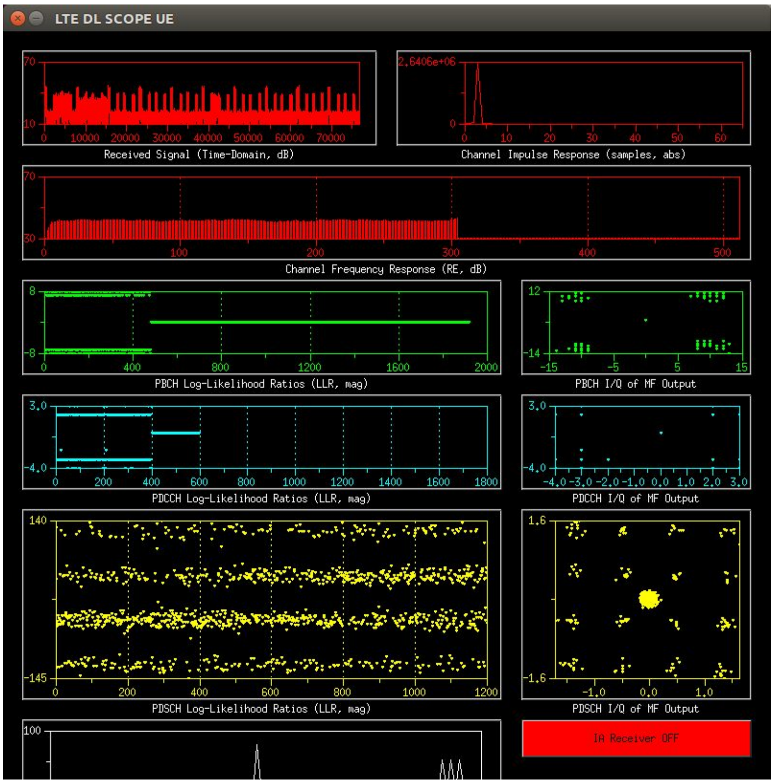
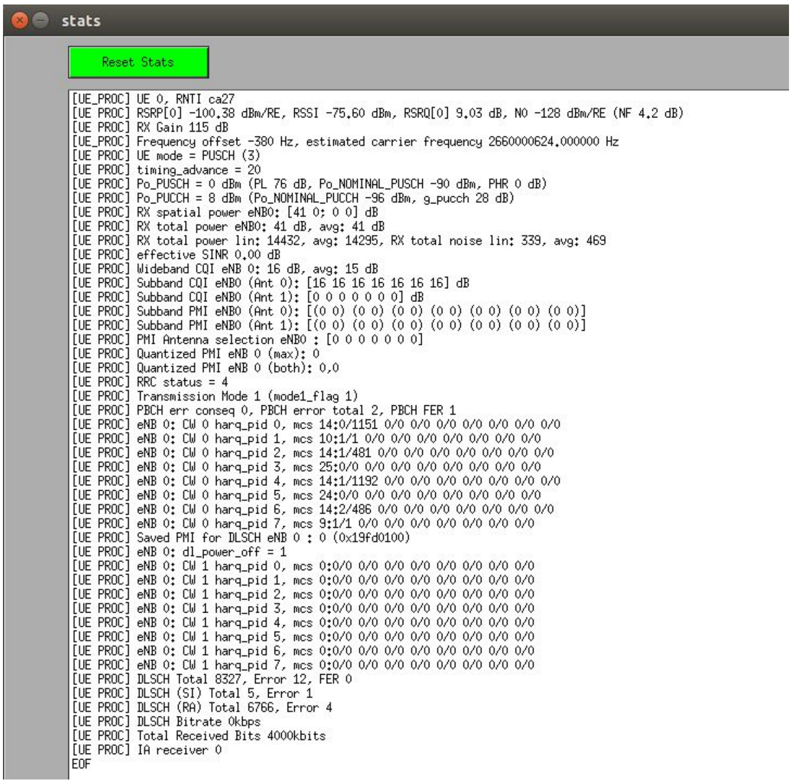


Figure 4.13 LTE DL scope UE

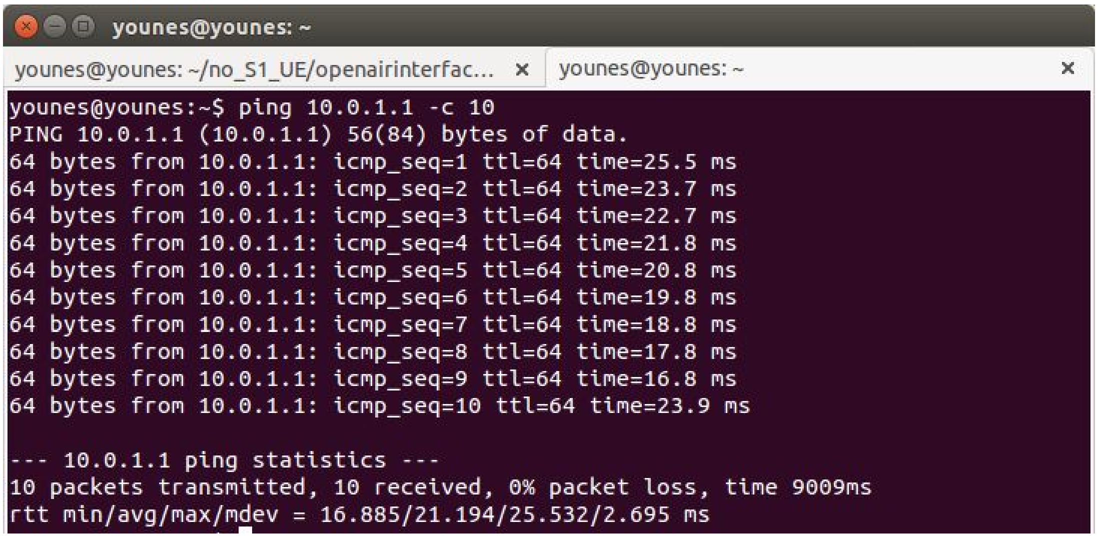


```
[UE_PROC] UE 0, RNTI ca27
[UE PROC] RSRP[0] -100.38 dBm/RE, RSSI -75.60 dBm, RSRQ[0] 9.03 dB, N0 -128 dBm/RE (NF 4.2 dB)
[UE PROC] RX Gain 115 dB
[UE_PROC] Frequency offset -380 Hz, estimated carrier frequency 2660000624.000000 Hz
[UE PROC] UE mode = PUSCH (3)
[UE PROC] timing_advance = 20
[UE PROC] Po_PUSCH = 0 dBm (PL 76 dB, Po_NOMINAL_PUSCH -90 dBm, PHR 0 dB)
[UE PROC] Po_PUCCH = 8 dBm (Po_NOMINAL_PUCCH -96 dBm, g_pucch 28 dB)
[UE PROC] RX spatial power eNB0: [41 0; 0 0] dB
[UE PROC] RX total power eNB0: 41 dB, avg: 41 dB
[UE PROC] RX total power lin: 14432, avg: 14295, RX total noise lin: 339, avg: 469
[UE PROC] effective SINR 0.00 dB
[UE PROC] Wideband CQI eNB 0: 16 dB, avg: 15 dB
[UE PROC] Subband CQI eNB0 (Ant 0): [16 16 16 16 16 16 16] dB
[UE PROC] Subband CQI eNB0 (Ant 1): [0 0 0 0 0 0 0] dB
[UE PROC] Subband PMI eNB0 (Ant 0): [(0 0) (0 0) (0 0) (0 0) (0 0) (0 0) (0 0)]
[UE PROC] Subband PMI eNB0 (Ant 1): [(0 0) (0 0) (0 0) (0 0) (0 0) (0 0) (0 0)]
[UE PROC] PMI Antenna selection eNB0 : [0 0 0 0 0 0 0]
[UE PROC] Quantized PMI eNB 0 (max): 0
[UE PROC] Quantized PMI eNB 0 (both): 0,0
[UE PROC] RRC status = 4
[UE PROC] Transmission Mode 1 (mode1_flag 1)
[UE PROC] PBCH err conseq 0, PBCH error total 2, PBCH FER 1
[UE PROC] eNB 0: CW 0 harq_pid 0, mcs 14:0/1151 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] eNB 0: CW 0 harq_pid 1, mcs 10:1/1 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] eNB 0: CW 0 harq_pid 2, mcs 14:1/481 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] eNB 0: CW 0 harq_pid 3, mcs 25:0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] eNB 0: CW 0 harq_pid 4, mcs 14:1/1192 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] eNB 0: CW 0 harq_pid 5, mcs 24:0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] eNB 0: CW 0 harq_pid 6, mcs 14:2/486 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] eNB 0: CW 0 harq_pid 7, mcs 9:1/1 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] Saved PMI for DLSCH eNB 0 : 0 (0x19fd0100)
[UE PROC] eNB 0: dl_power_off = 1
[UE PROC] eNB 0: CW 1 harq_pid 0, mcs 0:0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] eNB 0: CW 1 harq_pid 1, mcs 0:0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] eNB 0: CW 1 harq_pid 2, mcs 0:0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] eNB 0: CW 1 harq_pid 3, mcs 0:0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] eNB 0: CW 1 harq_pid 4, mcs 0:0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] eNB 0: CW 1 harq_pid 5, mcs 0:0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] eNB 0: CW 1 harq_pid 6, mcs 0:0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] eNB 0: CW 1 harq_pid 7, mcs 0:0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
[UE PROC] DLSCH Total 8327, Error 12, FER 0
[UE PROC] DLSCH (SI) Total 5, Error 1
[UE PROC] DLSCH (RA) Total 6766, Error 4
[UE PROC] DLSCH Bitrate 0kbps
[UE PROC] Total Received Bits 4000kbits
[UE PROC] IA receiver 0
EOF
```

Figure 4.14 Stats UE

4.4.7 Network Testing in Uu Interface.

In Figure 4.15, it is shown a successful connection between UE and eNB, ping command is used from UE.

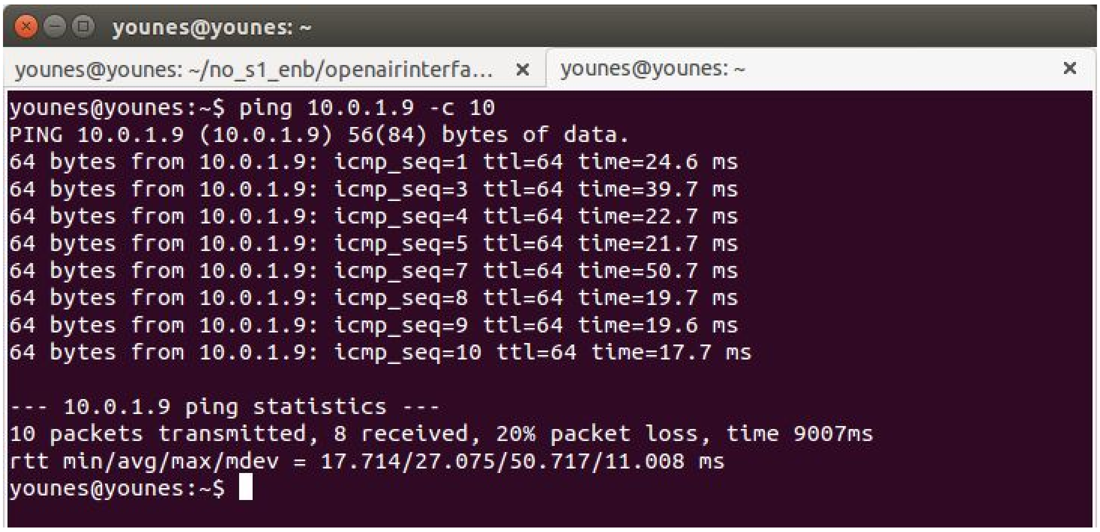


```
younes@younes:~$ ping 10.0.1.1 -c 10
PING 10.0.1.1 (10.0.1.1) 56(84) bytes of data.
64 bytes from 10.0.1.1: icmp_seq=1 ttl=64 time=25.5 ms
64 bytes from 10.0.1.1: icmp_seq=2 ttl=64 time=23.7 ms
64 bytes from 10.0.1.1: icmp_seq=3 ttl=64 time=22.7 ms
64 bytes from 10.0.1.1: icmp_seq=4 ttl=64 time=21.8 ms
64 bytes from 10.0.1.1: icmp_seq=5 ttl=64 time=20.8 ms
64 bytes from 10.0.1.1: icmp_seq=6 ttl=64 time=19.8 ms
64 bytes from 10.0.1.1: icmp_seq=7 ttl=64 time=18.8 ms
64 bytes from 10.0.1.1: icmp_seq=8 ttl=64 time=17.8 ms
64 bytes from 10.0.1.1: icmp_seq=9 ttl=64 time=16.8 ms
64 bytes from 10.0.1.1: icmp_seq=10 ttl=64 time=23.9 ms

--- 10.0.1.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9009ms
rtt min/avg/max/mdev = 16.885/21.194/25.532/2.695 ms
```

Figure 4.15 ICMP echo reply from eNB

In Figure 4.16 a successful connection between UE and eNB is shown, ping command is used from eNB.



```
younes@younes:~$ ping 10.0.1.9 -c 10
PING 10.0.1.9 (10.0.1.9) 56(84) bytes of data.
64 bytes from 10.0.1.9: icmp_seq=1 ttl=64 time=24.6 ms
64 bytes from 10.0.1.9: icmp_seq=3 ttl=64 time=39.7 ms
64 bytes from 10.0.1.9: icmp_seq=4 ttl=64 time=22.7 ms
64 bytes from 10.0.1.9: icmp_seq=5 ttl=64 time=21.7 ms
64 bytes from 10.0.1.9: icmp_seq=7 ttl=64 time=50.7 ms
64 bytes from 10.0.1.9: icmp_seq=8 ttl=64 time=19.7 ms
64 bytes from 10.0.1.9: icmp_seq=9 ttl=64 time=19.6 ms
64 bytes from 10.0.1.9: icmp_seq=10 ttl=64 time=17.7 ms

--- 10.0.1.9 ping statistics ---
10 packets transmitted, 8 received, 20% packet loss, time 9007ms
rtt min/avg/max/mdev = 17.714/27.075/50.717/11.008 ms
younes@younes:~$ 
```

Figure 4.16 ICMP echo reply from UE

4.5 Wi-Fi link

4.5.1 Overview schematic

Figure 4.17 shows a schematic overview to link UE machine with eNB machine through Wi-Fi interface.

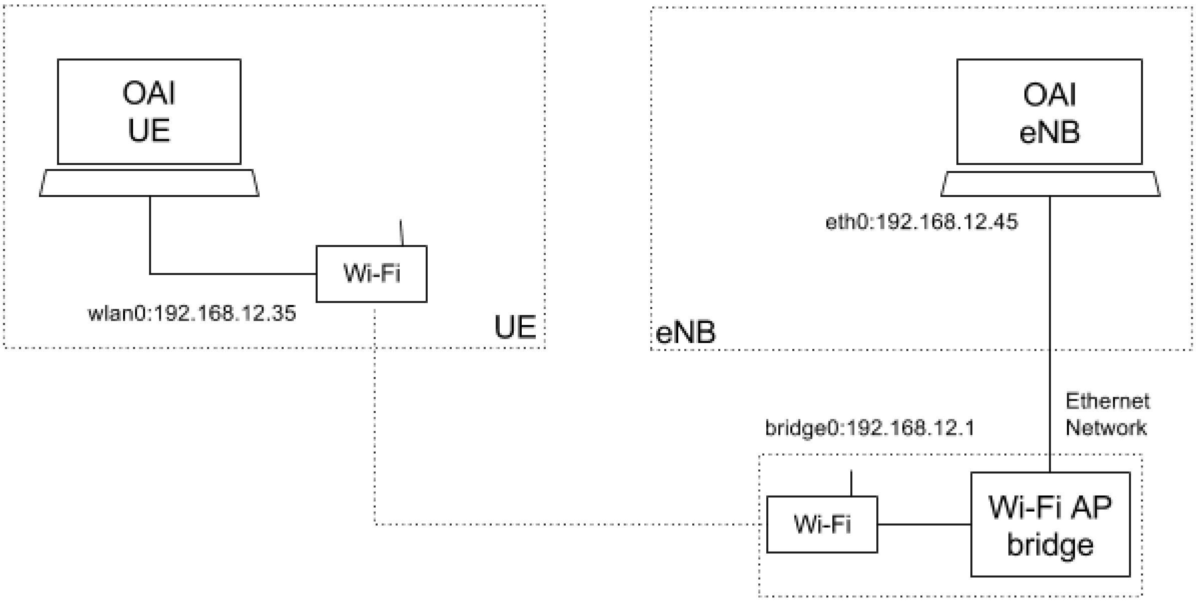


Figure 4.17 Overview schematic: Wi-Fi link

For the implementation of the Wi-Fi link, the principal hardware components are:

- eNB machine: one Ubuntu machine and one Ethernet interface.
- UE machine: one Ubuntu machine and one Wi-Fi adapter.
- Hostapd AP: one intermediate device to bridge Wi-Fi adapter and Ethernet interface.

In total, two Ubuntu machines, two Wi-Fi adapters, two Ethernet interfaces and one intermediate device is required. The computational requirements of the intermediate device is very low, hence any device that can run a Linux distribution is possible. In our evaluations, we successfully tested both Raspberry Pi and Odroid devices.

4.5.2 Hostapd

Hostapd [14] is an open-source software for Wi-Fi AP solution, which has two principal functions: Wi-Fi link layer and network configuration. The former is responsible of attaching wireless clients to access point and also that they can transmit/receive IP packets. The latter's goal is relaying IP packets between Ethernet/wireless interfaces of AP device.

In order to start, the hostapd software should be installed in the device. It will work as host access point.

```
$ sudo apt-get update
$ sudo apt-get install hostapd
$ apt-get install bridge-utils
```

Usually, the next step is to install a DHCP server, but because hostapd will be configured as a bridge, then DHCP is not required.

In /etc/hostapd/hostapd.conf file, Wi-Fi link layer settings are defined.

```
$ sudo vi /etc/hostapd/hostapd.conf
```

```
interface=wlan0
bridge=br0
driver=nl80211
ssid=RPi_AP
country_code=US
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
```

Later on, network configuration is applied over Ethernet/wireless interfaces. There are two ways to do this: bridge and NAT.

With NAT configuration, each interface (eth0,wlan0) has its corresponding subnet. NAT is a procedure used to interchange packets between two networks with incompatible addresses, in real time the addresses utilized are converted. In this implementation, we used bridge configuration because it supports the case when network is working in the same IP subnet.

To create a bridge, the network interfaces of the device should be binded. First, a bridge interface is created, then Ethernet interface is added and wireless interface will be added by hostapd, based on the configuration file settings shown above.

```
$ sudo ifdown eth0
$ sudo ifdown wlan0
$ sudo ifconfig eth0 0.0.0.0
$ sudo ifconfig wlan0 0.0.0.0
$ sudo brctl addbr br0
$ sudo brctl addif eth0
$ sudo ifconfig br0 192.168.12.1
```

The file /etc/sysctl.conf is updated to enable packet forwarding for IPv4.

```
$ sudo vi /etc/sysctl.conf
net.ipv4.ip_forward=1
```

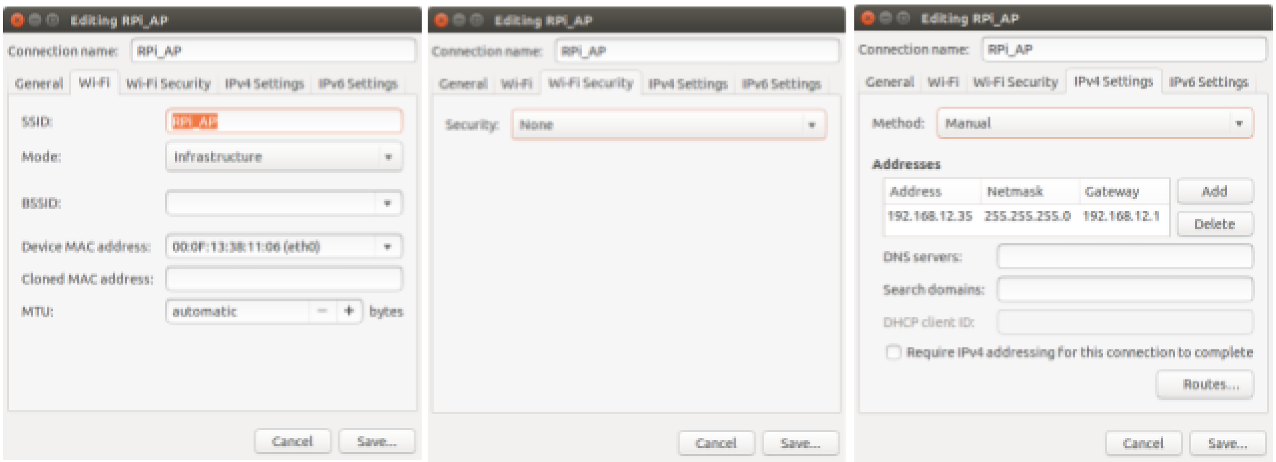
Finally, hostapd requires to be initialized:

```
$ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

4.5.3 Wireless interface configuration in UE machine

The hostapd is configured in bridge mode and it does not have a DHCP server, then wireless interface in UE machine should be configured manually.

Figure 4.18 shows the configuration of a new wireless network connection.



(a)
Editing RPi_AP
Connection name: RPi_AP
SSID: RPi_AP
Mode: Infrastructure
BSSID:
Device MAC address: 00:0F:13:38:11:06 (eth0)
Cloned MAC address:
MTU: automatic bytes

(b)
Editing RPi_AP
Connection name: RPi_AP
Security: None

(c)
Editing RPi_AP
Connection name: RPi_AP
Method: Manual

Addresses

Address	Netmask	Gateway
192.168.12.35	255.255.255.0	192.168.12.1

DNS servers:
Search domains:
DHCP client ID:
☐ Require IPv4 addressing for this connection to complete

(a) (b) (c)

Figure 4.18 Adding a new wireless network connection

In Wi-Fi tab (figure 4.18 a) SSID parameter is defined as “RPi_AP” and mode parameter as “infrastructure”. Wi-Fi security tab (figure 4.18 b) does not have any information because the communication uses an open authentication system. Finally in IPV4 Settings tab (figure 4.18 c) IP address and netmask of UE machine are defined.

Instead of a graphical configuration, this can also be configured through the following commands:

```
$ sudo ifconfig wlan0 192.168.12.35
$ sudo iwconfig wlan0 essid RPi_AP
```

4.5.4 Ethernet interface configuration in eNB machine

The communication is done at Layer 2, so that no routing is necessary, the important thing is to define the eNB Ethernet interface in the same network segment of the Wi-Fi link.

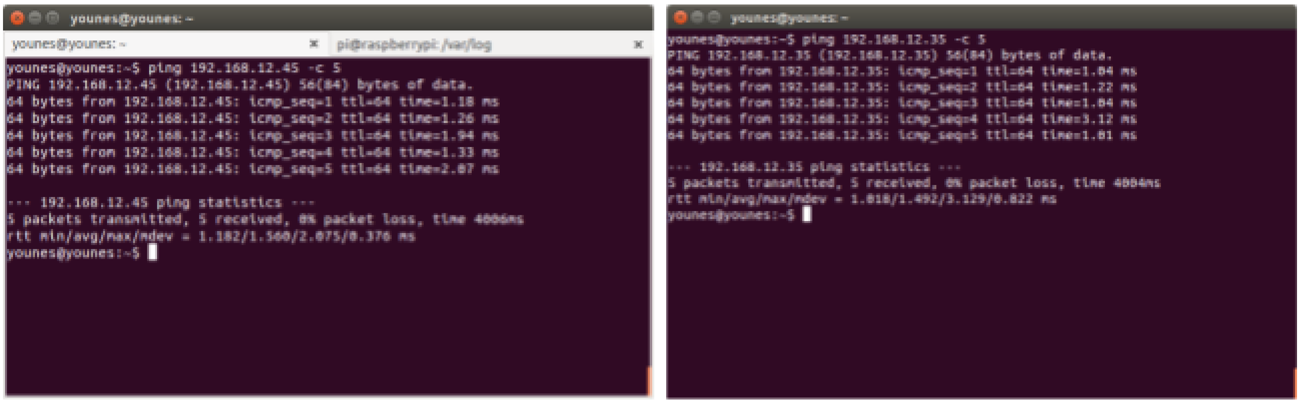
For a permanent configuration of the interface, the /etc/network/interface file is configured as follows:

```
$ sudo vi /etc/network/interface
```

```
auto eth0
iface eth0 inet static
address 192.168.12.45
netmask 255.255.255.0
```

4.5.5 Network Testing in Wi-Fi Interface.

In figure 4.19, a successful connection between UE and eNB is shown, in figure 4.19 (a) ping command is used from UE, and in figure 4.19 (b) ping command is used from eNB.



```
younes@younes:~$ ping 192.168.12.45 -c 5
PING 192.168.12.45 (192.168.12.45) 56(84) bytes of data.
64 bytes from 192.168.12.45: icmp_seq=1 ttl=64 time=1.18 ms
64 bytes from 192.168.12.45: icmp_seq=2 ttl=64 time=1.26 ms
64 bytes from 192.168.12.45: icmp_seq=3 ttl=64 time=1.94 ms
64 bytes from 192.168.12.45: icmp_seq=4 ttl=64 time=1.33 ms
64 bytes from 192.168.12.45: icmp_seq=5 ttl=64 time=2.07 ms

--- 192.168.12.45 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 1.182/1.560/2.075/0.376 ms
younes@younes:~$ 
```

```
younes@younes:~$ ping 192.168.12.35 -c 5
PING 192.168.12.35 (192.168.12.35) 56(84) bytes of data.
64 bytes from 192.168.12.35: icmp_seq=1 ttl=64 time=1.04 ms
64 bytes from 192.168.12.35: icmp_seq=2 ttl=64 time=1.22 ms
64 bytes from 192.168.12.35: icmp_seq=3 ttl=64 time=1.04 ms
64 bytes from 192.168.12.35: icmp_seq=4 ttl=64 time=3.12 ms
64 bytes from 192.168.12.35: icmp_seq=5 ttl=64 time=1.01 ms

--- 192.168.12.35 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 1.018/1.492/3.129/0.822 ms
younes@younes:~$ 
```

(a) (b)

Figure 4.19 Network testing in Wi-Fi interface

5 Evaluations

This chapter evaluates the performance of the system by applying each of the developed policies: No offload, offload, and aggregation. Before starting the transmission, it is assumed that UE/eNB machines have an established Wi-Fi connection and at zero time, the data transmission begins with the characteristics defined in the chosen policy, with a duration of 30 seconds.

The transmissions realized in LTE were configured with bandwidth of 5MHz, and in the oai0 interface with MTU of 1450.

The obtained results are presented with two perspectives: intra policy and inter policy. The former presents the data rates of downlink/uplink of the UDP/TCP traffic for each policy. In the latter, an overall perspective of the three used policies behavior is detailed by type of link and type of traffic.

5.1 Intra policy performance

5.1.1 No offload Policy

This policy corresponds to using only the transmission on LTE, which will serve as a reference to compare what happens with the system performance when the policies of offloading or aggregation are applied. Figure 5.1 shows the data rate observed with No Offload policy applied, where the number of LTE resource blocks (PRB) used is 25 that corresponds to 5 MHz channel.

Downlink UDP traffic has an average data rate of 12.44 Mbps being very similar to the TCP data rate achieved, which represents an average of 12.07 Mbps. The reason is that in DL the PDR values observed are 100%, i.e., there are almost no lost packets. In such case, the TCP and UDP performances are expected to be similar.

The performance in uplink is much lower compared to downlink. This is because, OAI limits the UL resources assigned even if there are available resources. Seeing in detail, the UDP traffic reaches data rates in average of 1.1 Mbps, while, in a similar way, average TCP traffic reaches 1.09 Mbps.

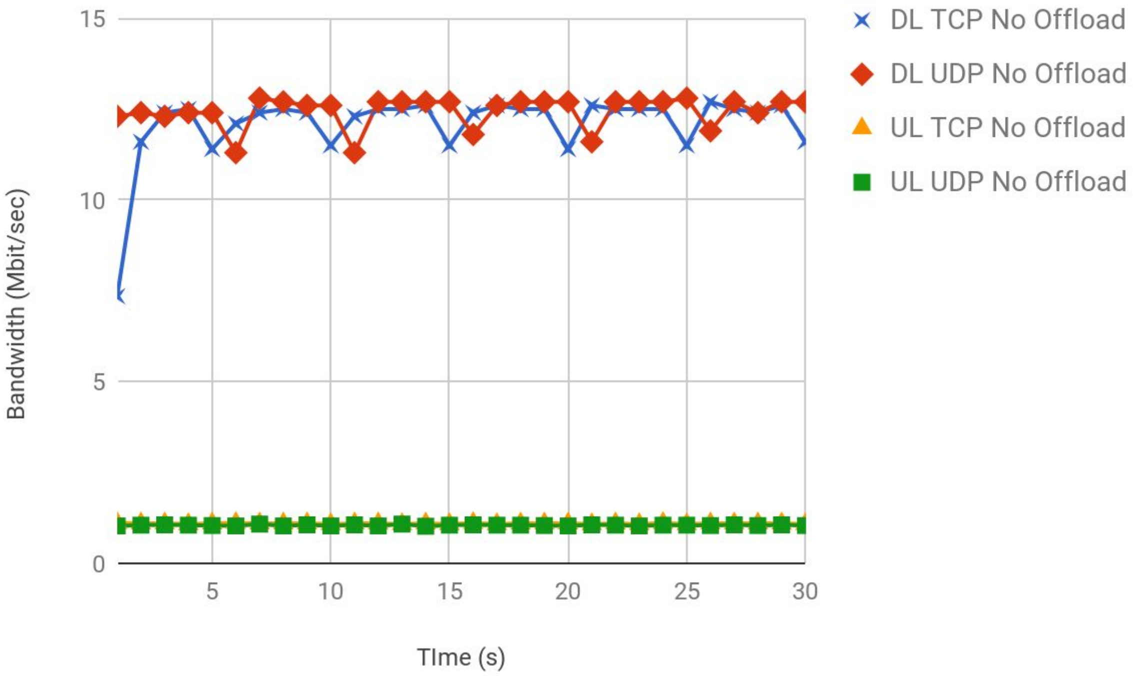


Figure 5.1 Performance: No offload policy with 25 PRBs

For the No offload policy, the number of resource blocks used was increased to 50 and as observed from figure 5.2, the data rate observed in downlink is almost double of 25 PRBs, which is expected as the bandwidth is doubled in this case. In case of uplink, data rate is same in comparison with a transmission with 25 PRBs, i.e., there is no improvement. The reason is that in OAI resource scheduling, the UL resources are limited to specific number of PRBs. In the figure, UL TCP and UDP results are overlapping.

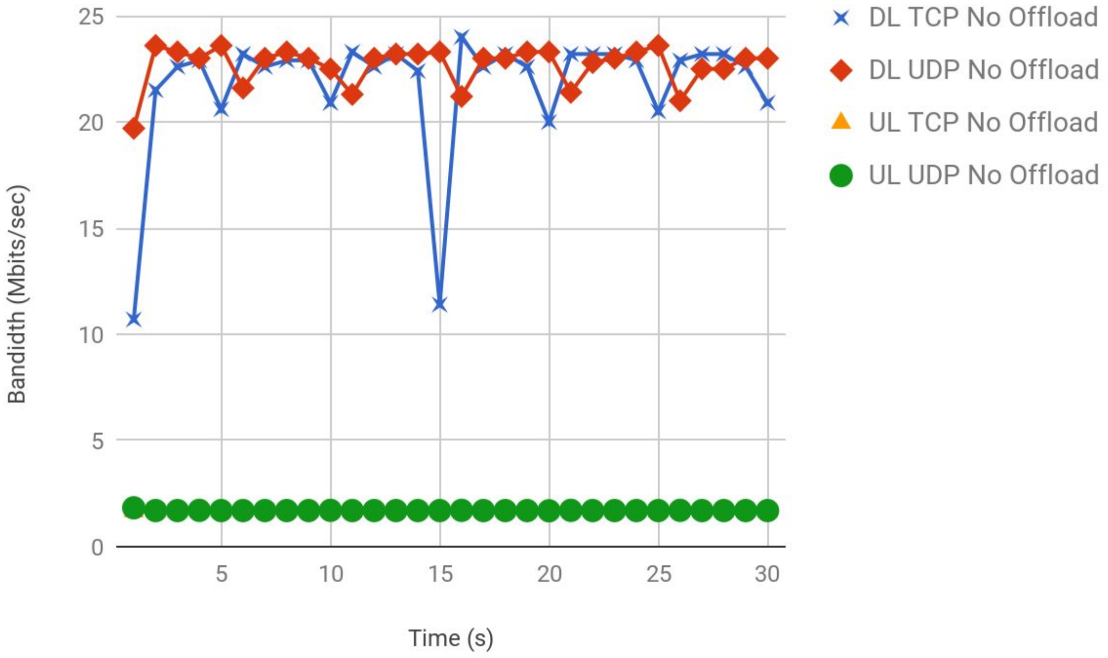


Figure 5.2 Performance: No offload policy with 50 PRBs

5.1.2 Offload policy

Next, system performance is evaluated when the offload policy is applied, i.e., in this scenario the data transmission is done exclusively on the Wi-Fi interface. Figure 5.3 shows the data rates reached with Offload policy applied.

Downlink UDP traffic has an average data rate of 27.27 Mbps, being higher than the TCP data rate that represents on average 22.87 Mbps. The reason is that Wi-Fi working on an unlicensed band, is open to external interference, which results in infrequent packet drops. In such cases, TCP reduces its congestion window, which results in less data rate then UDP.

The performance in uplink is very similar in comparison to the data rate of downlink. Seeing in detail, the UDP traffic reaches data rates in average of 24.05 Mbps, while on average TCP traffic reaches 21.74 Mbps. Note that, these values are similar to downlink performance, unlike the No offload case.

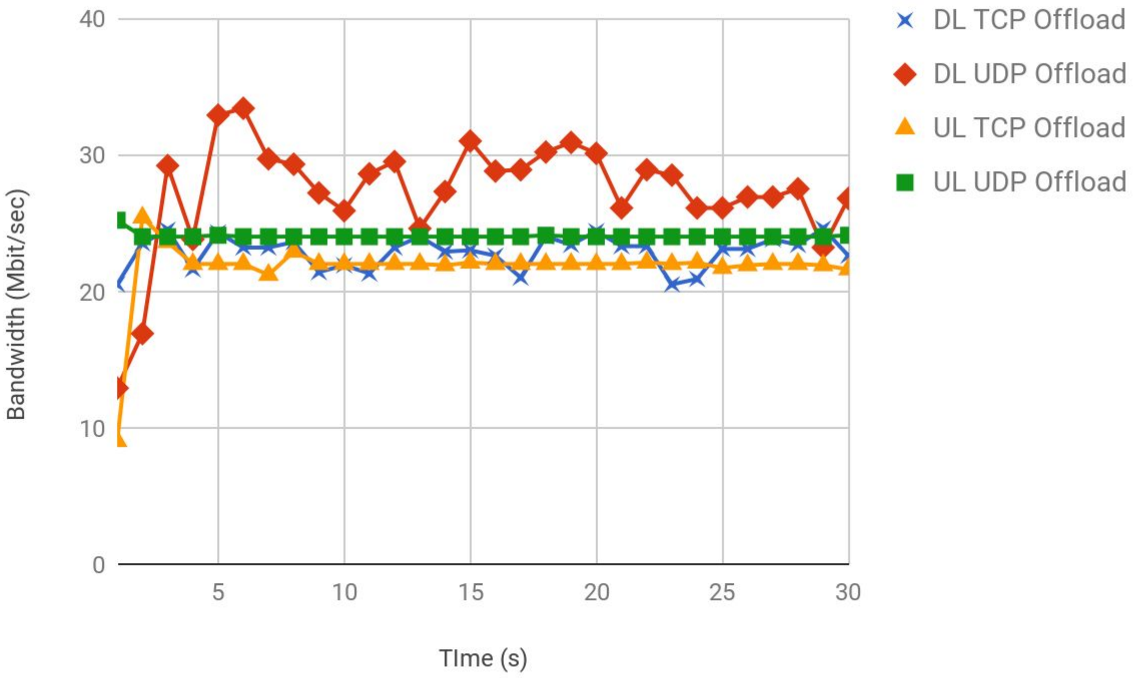


Figure 5.3 Performance: Offload policy

5.1.3 Aggregation policy

Next the system performance is evaluated when aggregation policy is applied, for which the data transmission is performed on both LTE and Wi-Fi interfaces. The interface selection is done with a simple policy to analyze the aggregation performance in detail. The policy is to send the even numbered frames through LTE and odd numbered frames through Wi-Fi. Figure 5.4 shows the data rates reached with aggregation policy applied.

Downlink UDP traffic has an average data rate of 25.41 Mbps, being higher than the TCP data rate which represents on average 8.85 Mbps. The much worse TCP performance can be explained with the out-of-order packets caused by different delays observed between interfaces. In this case, TCP at the receiver side does not acknowledge the individual packets if they arrive out of order. This makes the sender to reduce its window size, and, hence the lower data rate.

The performance in the uplink is much lower compared to the data rate of DL. Seeing in detail the UDP traffic reaches data rates in average of 2.94 Mbps, while the average TCP traffic reaches 1.73 Mbps. Although Wi-Fi has high uplink speeds, the LTE uplink speed limits the TCP performance.

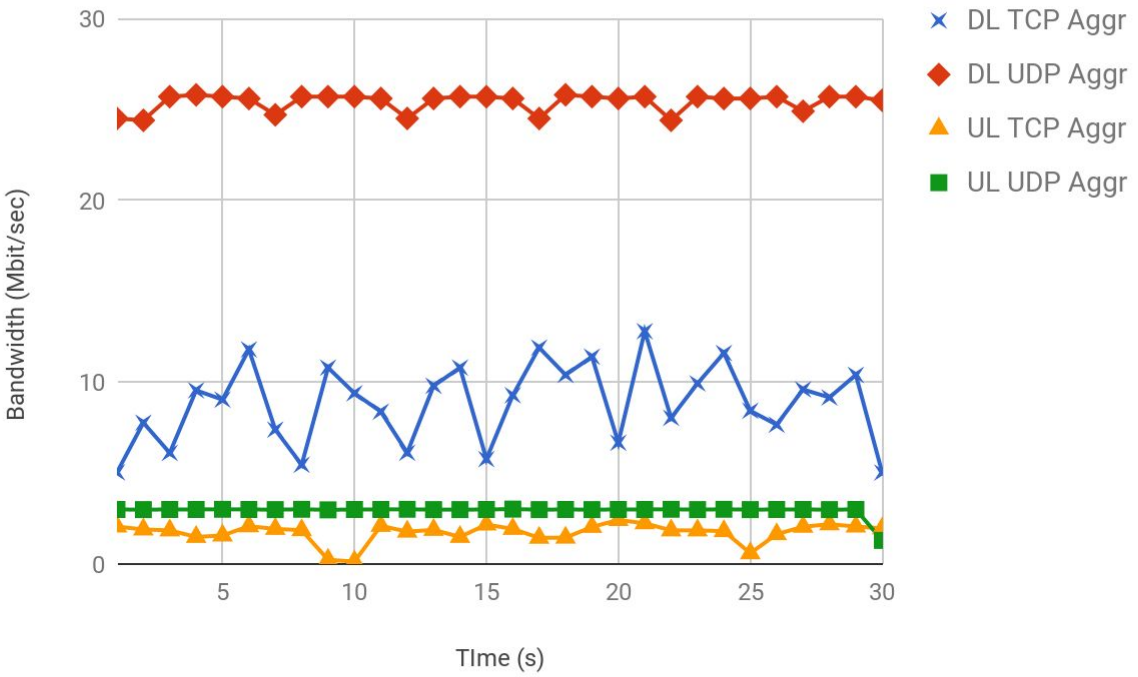


Figure 5.4 Performance: Aggregation policy for 25 PRBs

In the traffic aggregation policy, the resource block used was increased to 50 and as figure 5.5 shows, the performance obtained from the system, in all cases are doubled compared to the speeds achieved with the use of 25 PRBs. The reason is that since the LTE speed, which is the limiting factor, is doubled, the data rate achieved is also doubled.

Downlink UDP traffic has an average data rate of 49.65 Mbps being higher than the TCP data rate which represents on average 16.02 Mbps. Again here, the out-of-order packets create problem for TCP. According to iperf tool, the out-of-order packet percentage is around 50%, which is expected since half of the packets arrive earlier than the packet following them.

The performance in the uplink is lower compared to the data rate of DL, seeing in detail the UDP traffic reaches data rates on average of 4.54 Mbps while the average TCP traffic reaches 3.01 Mbps.

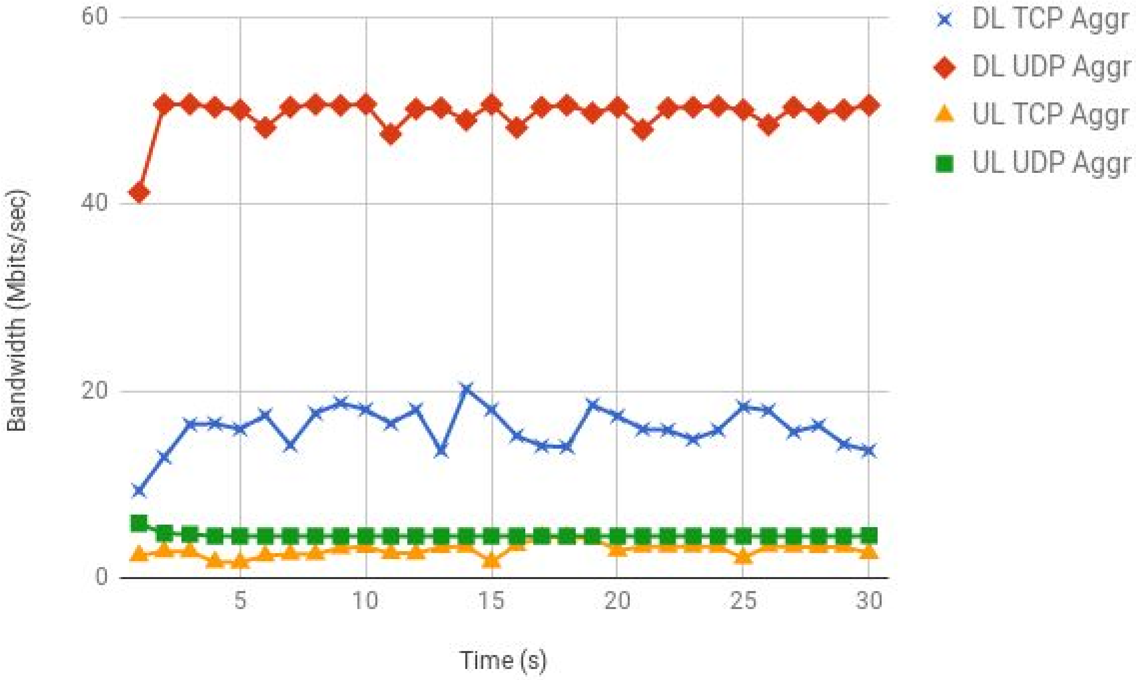


Figure 5.5 Performance: Aggregation policy with 50 PRBs

5.2 Inter policy performance

5.2.1 Downlink TCP

Figure 5.6 shows the data rate obtained in downlink with TCP traffic according to the different policies applied.

Undoubtedly, the highest data rate obtained in Downlink TCP traffic is in the data transmission on Wi-Fi interface (offload policy) with 22.87 Mbps, followed by the transmission made on LTE interface (offload policy) with 12.07 Mbps. Finally, the data rate obtained when sending traffic of alternated way, by the Wi-Fi and LTE interfaces (aggregation policy) with 8.85 Mbps is lower than the two previous cases. In practice, the low data rate obtained is due to the different delays that each interface has. The odd frames arrive at their destination after several even frames and TCP considers that the odd frames require retransmission, which degrades the data rate.

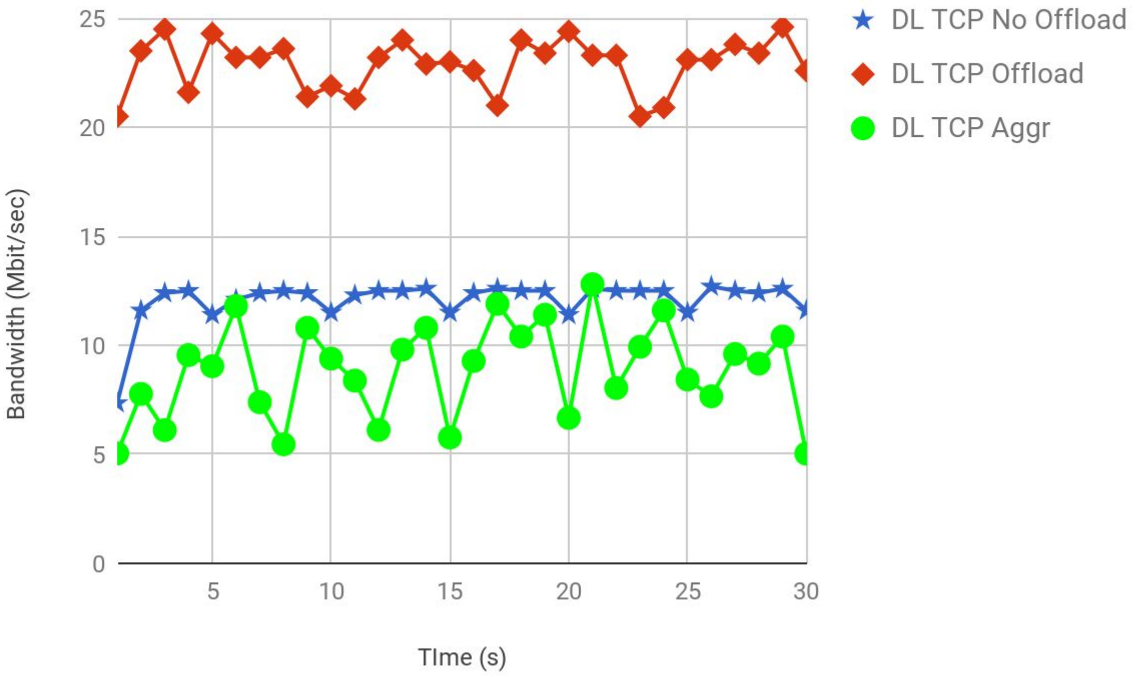


Figure 5.6 Performance: Downlink TCP with 25 PRBs for LTE

TCP is a protocol that considers the order of arrival of the packets, while UDP does not have control order of arrival packets. In section 5.2.2, a significant improvement had been evidenced in data rate through aggregation of traffic.

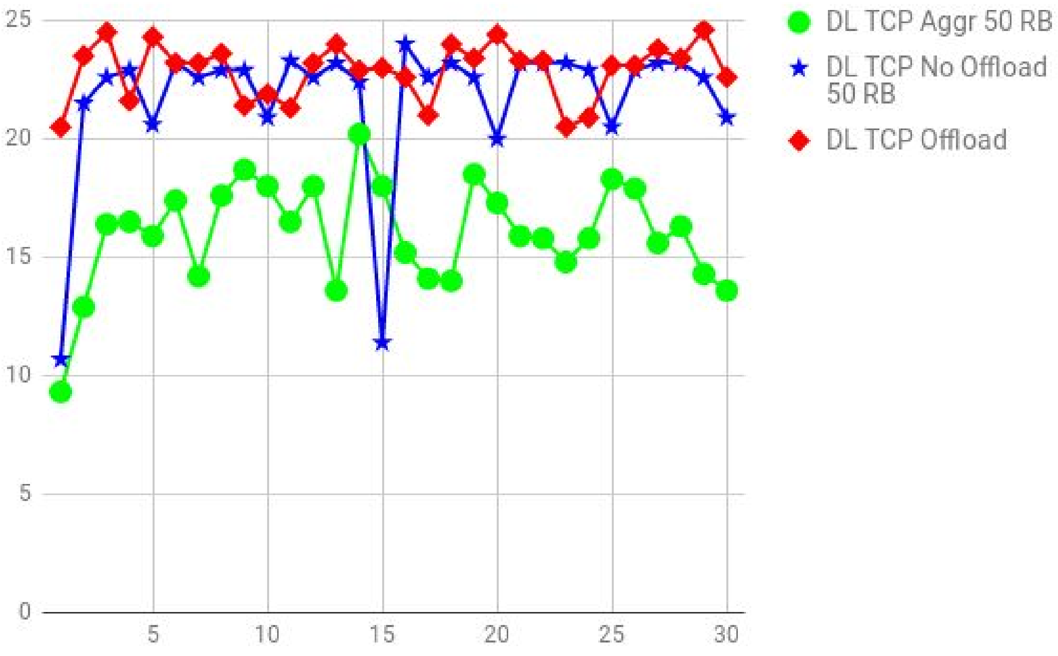


Figure 5.7 Performance: Downlink TCP with 50 PRBs for LTE

Figure 5.7 shows the performance obtained by using the 50 PRBs (10 MHz bandwidth) in no offload and aggregation policy, instead of 25 PRBs (5MHz bandwidth). In this case, the data rates are twice of those obtained with 25 PRBs (figure 5.6), which shows that the LTE speed is the limiting factor for the aggregation with TCP.

5.2.2 Downlink UDP

Figure 5.8 shows the data rate obtained in downlink with UDP traffic, according to the different policies applied.

In downlink UDP traffic, the referential data rate is 12.44 Mbps achieved in the transmission over LTE interface (no offload policy). The two policies implemented surpassed the benchmark performance, being the transmission over Wi-Fi the one that achieved the highest data rate with 27.27 Mbps, followed by the traffic aggregation transmission with 25.41 Mbps.

The UDP protocol does not perform packet order control, which allows a better data rate of 25.41 Mbps than the one obtained in the aggregation policy of TCP, i.e., 8.85 Mbps (section 5.2.1). As an example, 48% of datagrams received by UE arrived out of order according to iperf tool results.

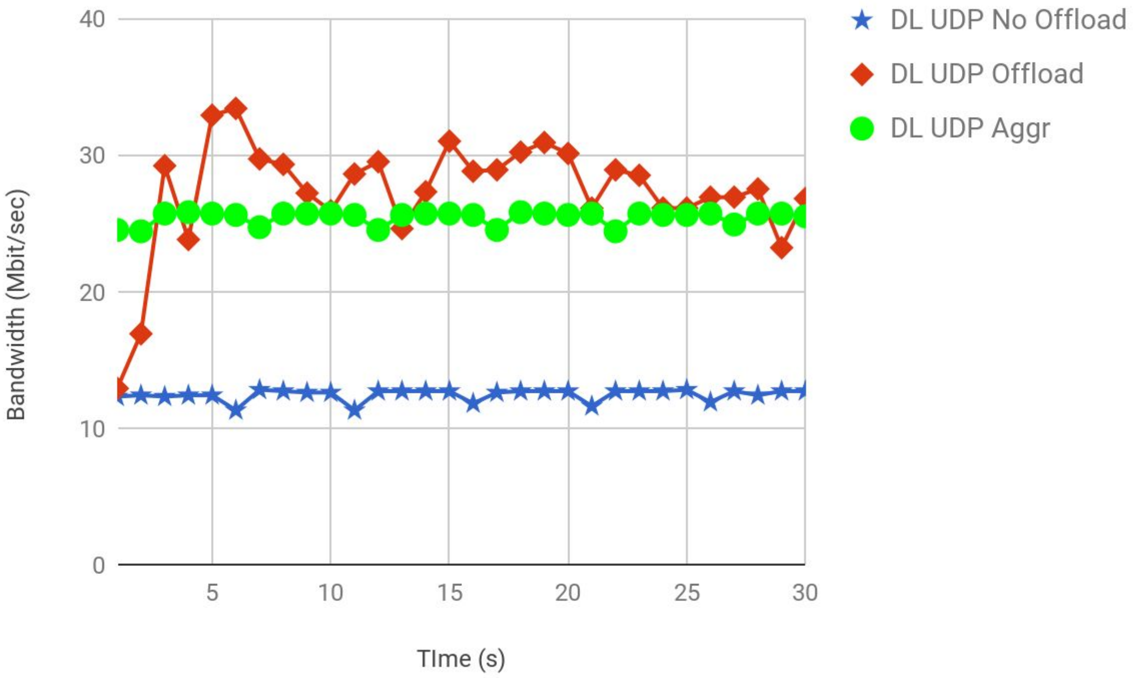


Figure 5.8 Performance: Downlink UDP with 25 PRBs for LTE

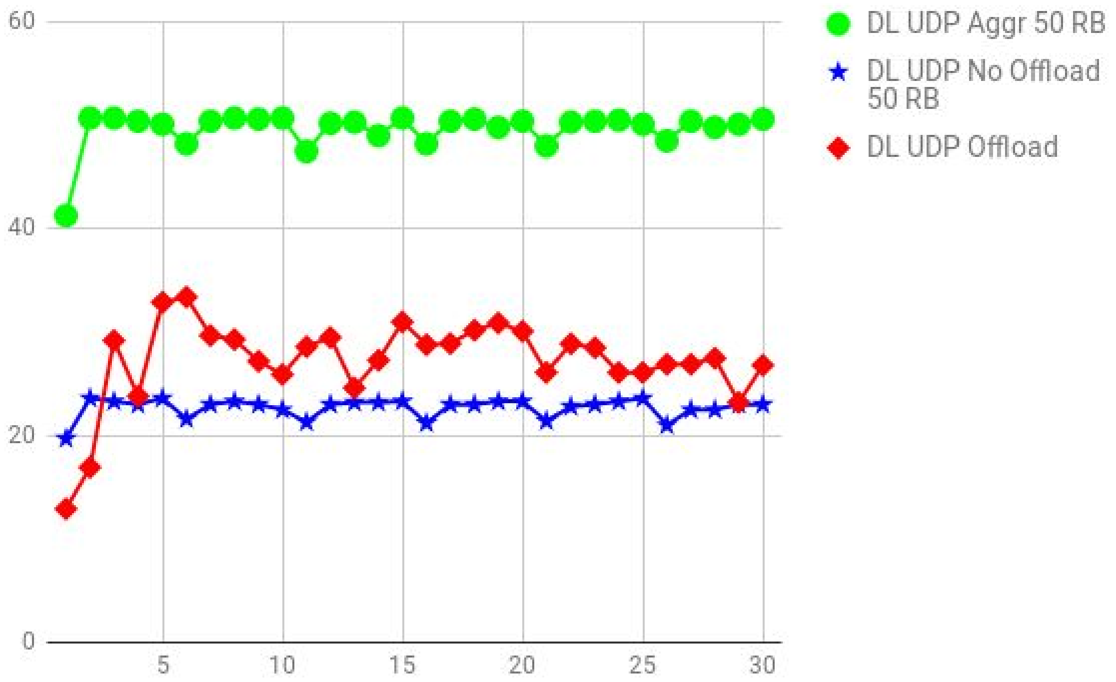


Figure 5.9 Performance: Downlink UDP with 50 PRBs for LTE

Figure 5.9 shows performance obtained using 50 PRBs (10 MHz bandwidth) for LTE. Data rates are twice of those obtained with 25 PRBs (see figure 5.8) as in the downlink TCP case. Note that close to 50 Mbps achieved by the aggregation is the highest achieved among all the tests in this thesis, which shows the promising feature of the aggregation policy.

5.2.3 Uplink TCP

Figure 5.10 shows the data rate obtained in uplink with TCP traffic, according to the different policies applied.

In uplink TCP traffic the referential bandwidth is 1.09Mbps, achieved in the transmission over LTE interface (no offload policy). The two policies implemented surpassed the benchmark performance, being the transmission over Wi-Fi the one that achieved the highest data rate with 21.71 Mbps, followed by the traffic aggregation transmission with 1.73 Mbps.

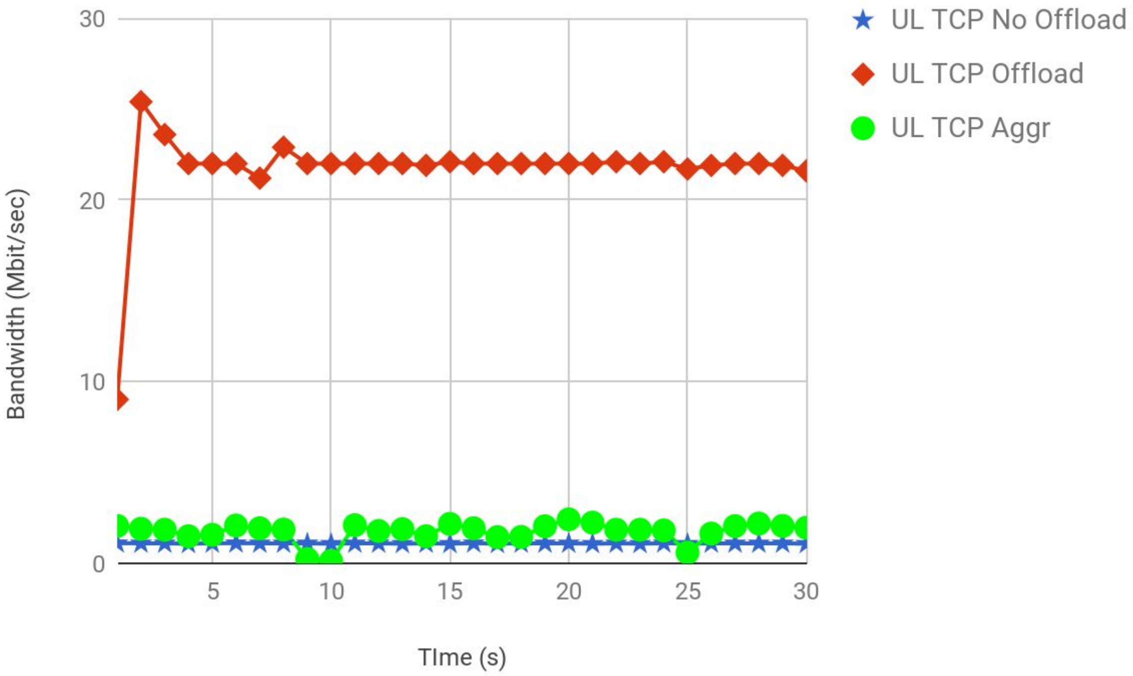


Figure 5.10 Performance: Uplink TCP with 25 PRBs for LTE

Figure 5.11 shows performance obtained with 50 PRBs (10 MHz) applied to no offload and aggregation policy. In the case of no offload policy, 1.68 Mbps was obtained being a very similar value to the one obtained with the use of 25 PRBs (5MHz). In aggregation policy, an average of 3.01 Mbps represents 73% of improvement compared to the use of 25 PRBs (5MHz).

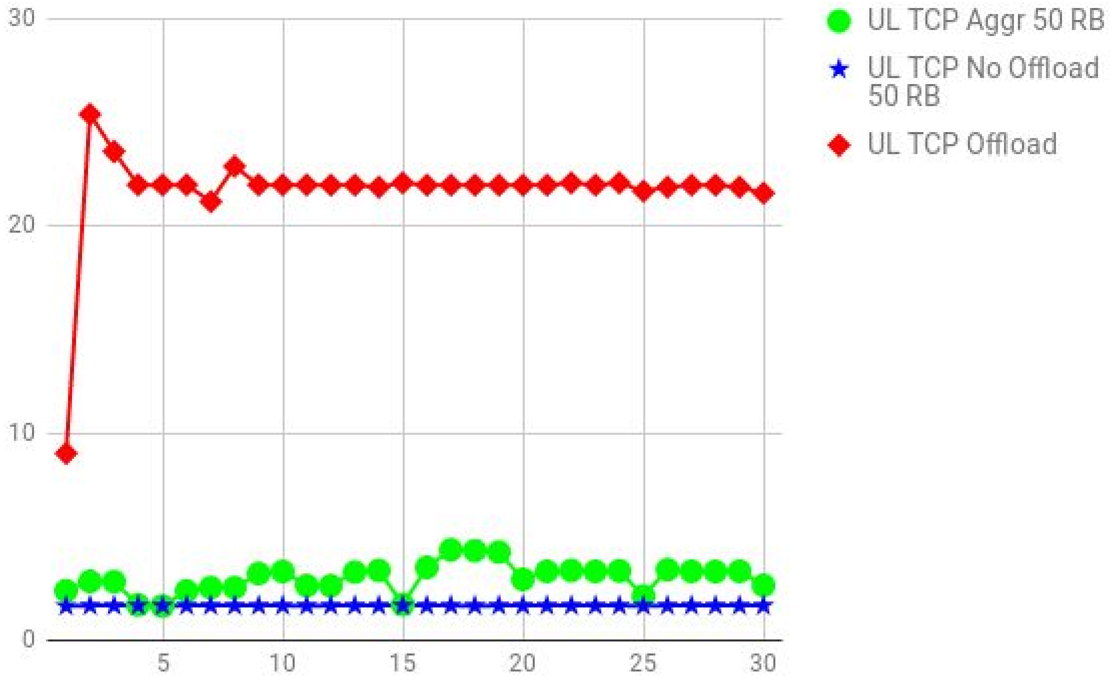


Figure 5.11 Performance: Uplink TCP with 50 PRBs for LTE

5.2.4 Uplink UDP

Figure 5.12 shows the data rate obtained in uplink with UDP traffic, according to the different policies applied.

In uplink UDP traffic, the referential bandwidth is 1.1Mbps, achieved in the transmission over LTE interface (no offload policy). The two policies implemented surpassed the benchmark performance, being the transmission over Wi-Fi the one that achieved the highest data rate with 24.05 Mbps, followed by the traffic aggregation transmission with 2.94 Mbps.

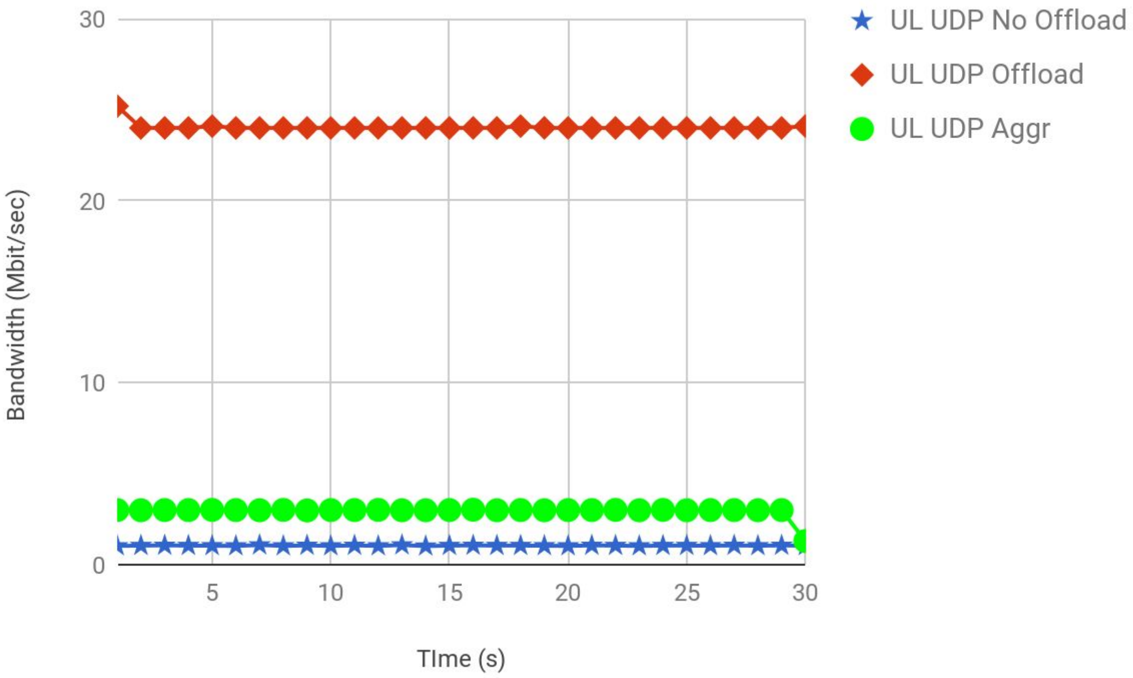


Figure 5.12 Performance: Uplink UDP with 25 PRBs for LTE

Figure 5.13 shows performance obtained with 50 PRBs (10 MHz) applied to no offload and aggregation policy. In the case of no offload policy 1.70 Mbps was obtained being a very similar value to obtained with the use of 25RB (5MHz), there is no improvement. In aggregation policy 4.54 Mbps was obtained, it represents 128% of improvement compared to the use of 25 PRBs (5MHz).

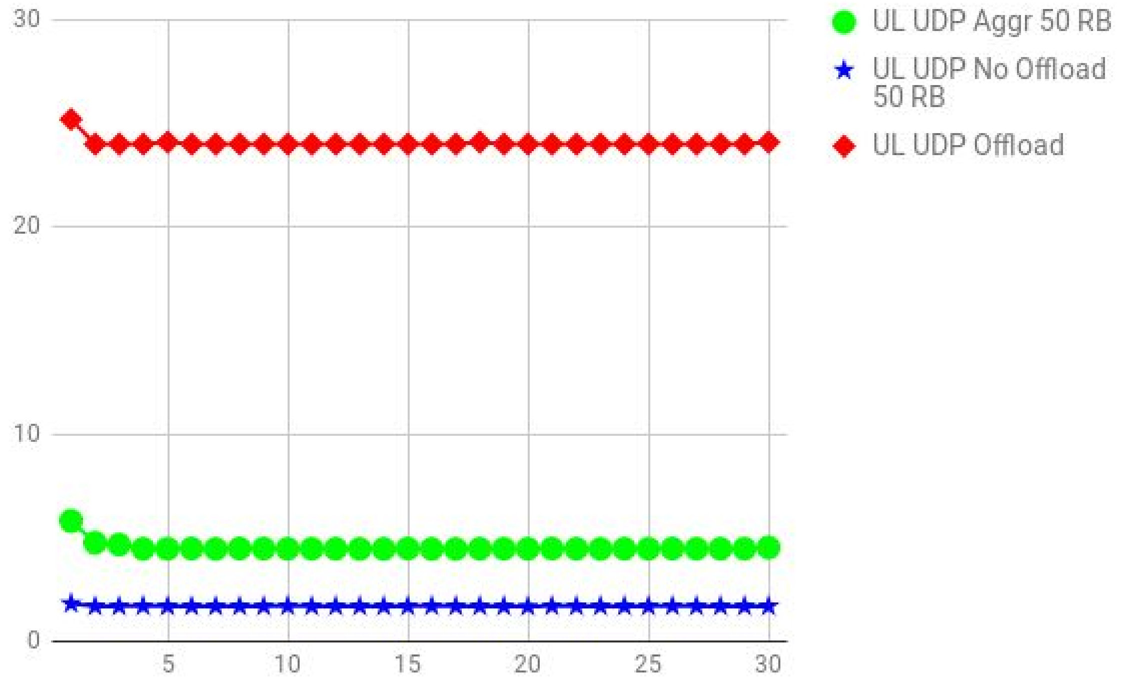


Figure 5.13 Performance: Uplink UDP with 50 PRBs for LTE

5.3 Adding delay artificially

Finally, the delay effect between the Wi-Fi and LTE interfaces is evaluated when the aggregation policy is applied to TCP. The netem tool allows adding delays to wlan and eth interfaces related to UE/eNB machines respectively. In each interface half of the difference between LTE and Wi-Fi link RTTs is applied to make them have similar delays.

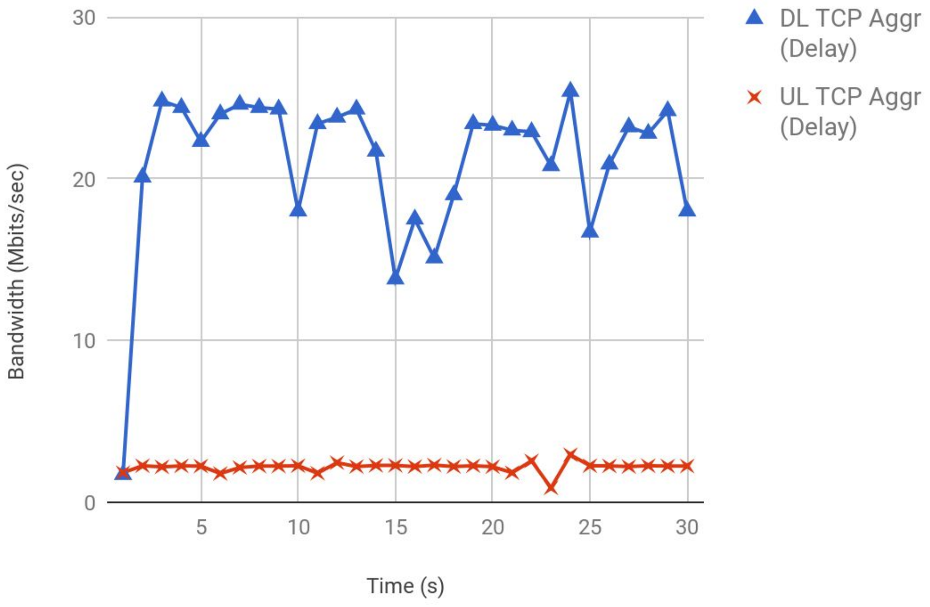


Figure 5.14 Performance: Aggregation TCP throughput with artificial delay adjustment

Figure 5.14 shows a better performance in downlink than the one obtained without delay applied to LTE/Wi-Fi interfaces. Data rate increases from 8.85 Mbps to 21.06 Mbps. This shows the crucial impact of delay difference between interfaces on TCP performance. However, in uplink no significant data rate changes are seen.

6 Conclusions and future work

In this project a prototype was implemented to obtain a very tight coupling between LTE and Wi-Fi technologies, based on open source and commodity hardware. Three functionalities were developed on open-source Open Air Interface LTE project to enable, disable and interact with the properties of the lower layers of both technologies.

Functionalities implemented are:

1) No offload traffic: in this functionality the data traffic remains on the LTE network without interaction of the Wi-Fi.

2) Offload traffic: in this functionality data traffic is transmitted over Wi-Fi. PDCP PDUs are sent to adaptation layer where a header is included with information necessary to uniquely identify the frame, then the frame passes to the Ethernet layer that includes the corresponding headers. Finally, the frame arrives at the Wi-Fi AP where the IEEE 802.3 headers are replaced by the IEEE 802.11 headers, to be transmitted by the Wi-Fi interface.

3) Traffic aggregation: in this functionality the frames are transmitted alternately in the LTE and Wi-Fi interfaces, the decision is based on a simple policy: even numbered frames are sent through LTE, and odd numbered frames are sent through Wi- Fi, i.e. with a transmission ratio of 50% over Wi-Fi, and 50% over LTE.

Aggregation and offloading process are managed by the eNB, therefore, the core network remains intact without any modification. The switch or split of the bears according to the chosen policy is done in PDCP layer, which is the common layer between LTE and Wi-Fi.

In traffic aggregation scenario, a substantial improvement in the data rate is evidenced for UDP traffic. However, the data rate does not increase for TCP with aggregation due to the difference of delays between the Wi-Fi and LTE links, which causes the performance degradation by the arrival of packets in a non-sequential way. The datagrams received out-of-order are around 50%. However, because UDP has no control of the order arrival packets, the aggregation increases the UDP performance significantly. We provided and evaluated a solution to mitigate the delay difference problem which improved the TCP performance for aggregation policy.

The uplink performance of Open Air Interface system is independent of the resource blocks configured, which limits the results obtained in uplink with the aggregation of traffic implemented in this research.

A future research work can be the implementation of selection process of transmission interface, based on performance measurement reports of the Wi-Fi and LTE channels, with the purpose that RRC configure the UE. At the same time, the eNB sends an RRC reconfiguration message to the UE to state which policy it should use.

Bibliography

[1] CISCO, “Cisco Visual Networking Index : Forecast and Methodology , 2014 – 2019,” 2015.

[2] Ericsson, “Ericsson Mobility Report,” 2014.

[3] CISCO, “Cisco Visual Networking Index : Global Mobile Data Traffic Forecast Update , 2012 – 2017,” 2017.

[4] X. Lagrange, “Very Tight Coupling between LTE and Wi-Fi for Advanced Offloading Procedures,” 2014.

[5] H. Takagi and B. Walke, *Spectrum Requirement Planning in Wireless Communications*. 2008.

[6] R. Comes, B. F. Álvarez, F. Casadevall, R. Ferrús, J. Pérez, and O. Sallent, *LTE: Nuevas tendencias en comunicaciones móviles*. 2010.

[7] R. Flickenger, C. Aichele, S. Buttrich, and L. Drewett, *Wireless Networking in the Developing World*, Second edi. 2007.

[8] I. The Institute of Electrical and Electronics Engineers, “Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” 2012.

[9] B. Jeon and H. J. Son, “LTE / Wi-Fi Multipath Aggregation,” *Netvision Telecom*, 2015.

[10] ETSI, “TS 136 300 - V13.2.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (3GPP TS 36.300 version 13.2.0 Release 13),” 2016.

[11] P. Sharma, A. Brahmakshatriya, T. V. P. S, B. R. Tamma, and A. Franklin, “LWIR : LTE-WLAN Integration at RLC Layer with Virtual WLAN Scheduler for Efficient Aggregation,” 2016.

[12] T. Valerrian, S. Patro, B. Reddy, and A. Franklin, “Tight coupling of LTE Wi-Fi Radio Access Networks - A Testbed Evaluation,” *Networked Wirel. Syst. Lab*, 2016.

[13] Y. Khadraoui, X. Lagrange, and A. Gravey, “Very Tight Coupling Between LTE and WiFi : From Theory To Practice,” *Wirel. Days*, pp. 7–9, 2016.

[14] M. Gast, *802.11 Wireless Networks The Definitive Guide*, 2005.

[15] M. Litzenburger, H. Bakker, S. Kaminski, and K. Keil, “Very Tight Coupling of Wireless LANs and UMTS Networks : A Technical Challenge and an Opportunity for Mobile Operators,” *Alcatel Res. Innov.*, 2003.