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

LTE SELF-BACKHAULING: IMPLEMENTATION AND EVALUATION

Master's Thesis
Submitted to the Faculty of the
Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by

JOSÉ UMBERTO GAMBOA SÁNCHEZ

In partial fulfillment
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: **LTE Self-backhauling: Implementation and Evaluation**

Author: **José Umberto Gamboa Sánchez**

Advisor: **Dr. Ilker Demirkol**

Abstract:

Long Term Evolution (LTE) mobile networks are becoming more available for people all around the world, as it is the most deployed broadband communication technology. To address the exponential data requirements projected for the future, a targeted solution for LTE is network densification, i.e., deploying small cells (cells with reduced radius compared to macro cells), which will enable better frequency reuse. However, a challenge here is the backhauling of these small cell evolved-NodeBs (eNBs), since the fiber-based backhauling is costly and is not physically feasible for many cases. For this, a recent idea is to use LTE self-backhauling, where an eNB can relay its data to another eNB through the use of LTE technology.

In this thesis, we develop and evaluate an implementation of LTE Self-Backhauling building on an open-source software and commodity hardware (regular PCs and low-cost Software-defined Radios) for the LTE system. For this, we implement a self-backhauled eNodeB (B-eNB), which connects to an Anchor-eNB (A-eNB) using an LTE UE. Through physical experimentation using off-the-shelf UEs, we show that the method proposed is viable and can improve the network coverage and capacity.

Keywords: LTE, 4G/5G mobile networks, eNB, EPC, OpenAirInterface, Self-Backhauling.

Dedicated to my parents and my family

Acknowledgments

First of all I would like to thank GOD, who has always blessed me. I think this thesis wouldn't be possible without the supervision and ideas of my advisor, Ilker Demirkol. I would like to thank him for all his support and cooperation which lead me to this point. Working with him in LTE OAI, has given me a lot. It helped me to understand better and enhance my interests in 4G and future 5G networks.

I would like to thank to my laboratory partners for their support and cooperation. I would like to thank Zainul Abideen who helped me since the first day with the understanding, installation, and configuration of the OAI project. To Khaled Hijjeh for listening, supporting, and motivating all my ideas about my thesis. To Antonio Uribe and Leticia Lemus who were good friends to me, making me had a great time here. And to Diego Ibarra, who started the thesis project with me and helped me along this journey.

Finally, I am extremely thankful to my parents, my sister, my brother, and all of the rest of my family in Peru who have always supported me. Without their support I wouldn't be here in Barcelona finishing my Master's degree.

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

telecom BCN

Approval Record

Written by		Reviewed and Approved by	
Date	July 17, 2017	Date	July 17, 2017
Name	José Gamboa	Name	Dr. Ilker Demirkol
Position	Thesis Author	Position	Thesis Supervisor

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last years, LTE mobile communication has been the most deployed broadband communication technology. This technology is growing fast, the number of mobile phone users, nowadays, are close to 5 billions users [27]. There are many projections on exponential increase in traffic demand in cellular networks, which is a challenge for the mobile operators.

Moreover, due to the fast increase in number of the mobile phone users and their traffic demands, some base stations, in LTE known as evolved NodeB (eNB), are working at their full capacity. This means that they are saturated, in consequence there must be more deployment of eNBs to increase the capacity and the radio coverage of the LTE network.

In LTE, by design, it is known that a fixed eNB is physically connected to the core network through a transport network. This connection is known as backhaul link, and the deployment of this network is an issue, because the cost of its deployment has a lot of significance in the overall operation cost of the network. Moreover, there are places that an eNB can not be deployed because transport network connection is not possible due to the environment.

For the wired connection in backhaul, fiber is used for the deployment of the access link in LTE, to reach the high velocities required in this network. However, for scenarios such as rural areas, fiber is not always feasible, in these cases, a wireless solution is a better choice for the access link in LTE, and also is cost-effective.

One solution for wireless backhaul is the microwave [12] and another solution is Millimeter Wave (mmWave) [28], bands between 30 and 300 GHz, where the available bandwidths are wider than the actual bandwidth of cellular networks [9]. The latter technology is suitable for LTE backhaul and future 5G networks. However, the deployment of mmWave for cellular networks has many technical obstacles [21], mmWave signals are extremely susceptible to shadowing [21]. Moreover, the operation cost is high since the mmWave system is fraught with considerable complexities [30].

An alternative and novel wireless backhauling solution is self-backhauling [4], which is studied in this thesis. Self-backhauling enables the wireless backhauling of an eNB (named self-backhauled eNodeB (B-eNB)) using the existing LTE radio interface of another eNB (named Anchor eNodeB (A-eNB)) as a backhaul link. The concept of this solution is shown in Figure 1.1.

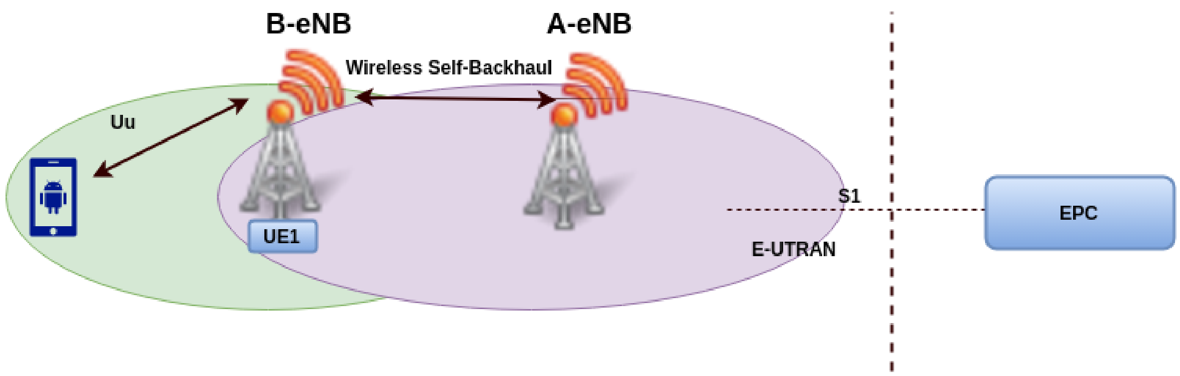


Figure 1.1: Self-Backhauling Network

Using the existing LTE radio interface will provide a better cost-efficiency solution, since they are sharing the same O&M systems (Operation and maintenance) simplifying the system management [3], also a higher spectrum utilization, since the reuse of time, frequency and space resources between access and backhaul link [3]. However, using in-band self-backhauling a new type of interference called access-backhaul interference or self-interference will be created (shown in Figure 1.2), since access and backhaul link share the same carrier frequency, this means that B-eNB transmits and receives in the same band, the transmitted signal interferes with the received signal. And to mitigate this new interference, a sophisticated (complex) scheduling of channel resources between access and backhaul will be required [4]. On the other hand, self-interference does not occur when using out-band self-backhauling, where access and backhaul link operate on separate carrier frequencies, in consequence a better performance is obtained, but the drawback is the extra transceiver costs.

Some use cases for self-backhauling LTE network would be core-isolated eNBs which are eNBs that has no connection to the core network; small cells to increase the capacity of the macro-cell of the network; and moving cells to use moving eNBs for transport applications[1].

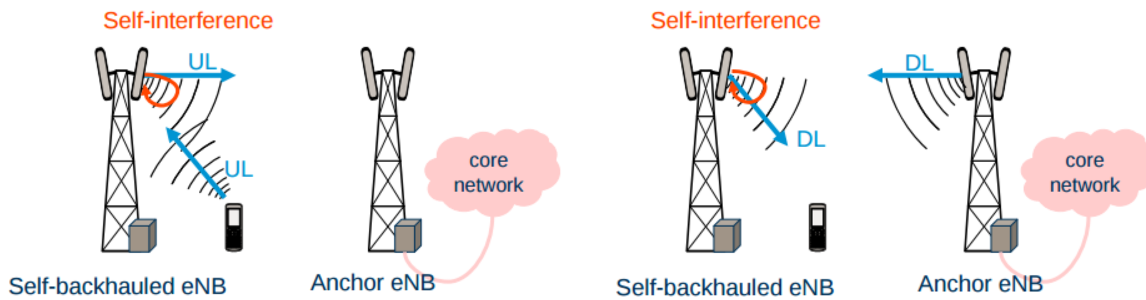


Figure 1.2: Self-Interference at Self-Backhauling Network [5]

In this thesis, we develop and evaluate an implementation of LTE Self-Backhauling building on an open-source software for the LTE system. For this, we used the OpenAirInterface (OAI) project, which enables the setup of an open LTE network on a generic purpose PC [13][26] requiring only a small form-factor Software Defined Radio (SDR). OAI provides an open-source ecosystem for the core network, which is the Evolved Packet Core (EPC), and the access network, which is the Evolved Universal Terrestrial Radio Access Network (E-UTRAN). Using the OAI project it is possible to build an LTE base station and core network on one or more PCs, and connect commercial User Equipments (UEs) using an SDR, in order to test the LTE network in real time. Comparing with commercial existing solutions, the solution provided by this project is cheaper since low cost PCs and SDRs are used, instead of using real eNBs and EPC.

One of the major challenges encountered during the project was the double GTP encapsulation when the first setup was realized, this first setup is only using B-eNB and A-eNB as traditional eNBs. In order to solve this double GTP encapsulation problem, several potential solutions are presented and one of the solutions is implemented and show to resolve this problem.

For the validation of the results of this project, physical evaluations were done to compare several performance metrics between a UE connecting to the conventional LTE network setup and the same UE connecting through a Self-Backhauling network. For the self-backhauling network, two solutions were presented and evaluated: the first was in-band self-backhauling network, where the self-backhauled eNB (B-eNB) shares the same central frequency and bandwidth as the backhauling eNB (A-eNB) and the second was out-band self-backhauling network, where they are assigned non-overlapping frequency spectrum. The performance of the self-backhauling network and conventional approach with the OAI setup were evaluated by measuring downlink and uplink throughput using the tools iperf and speedtest application. Through physical experimentation using off-the-shelf UEs, we show that the method proposed is viable, and it is shown that in terms of performance using a self-backhauling network is a good choice to increase the capacity of the network and the radio coverage.

Chapter 2

Background

2.1. Evolved Packet System

The Evolved Packet System (EPS) is introduced by 3GPP standardization group as the evolution of the 3G/UMTS standard. EPS is purely IP based [11], it routes the IP traffic from a gateway in the PDN to the UE using the concept of EPS bearers [25][18].

2.1.1. Long Term Evolution (LTE)

LTE is introduced in 3GPP Release 8 as the radio access evolution of the Universal Mobile Telecommunications System (UMTS) through the Evolved Universal Terrestrial Radio Access Network (E-UTRAN). LTE is a packet-switched system, different from old mobile technologies that are circuit-switched systems. LTE is the radio access part of the EPS, which provides wireless Internet Protocol (IP) connection between user equipment (UE) and the packed data network (PDN); and the System Architecture Evolution (SAE), which is the Evolved Packet Core (EPC), is the evolution of the core network. LTE and SAE together comprise the Evolved Packet System (EPS)

2.1.2. EPS Architecture

EPS provides access to internet to the end user with IP connectivity to a PDN. In figure 2.1 is shown the overall network architecture of the EPS. EPS network comprises of the core network EPC, which consists of many logical nodes that will be explained in the next section, and the access network E-UTRAN, which consist in just one type of node called evolved NodeB (eNB), that connects to the UEs. [25].

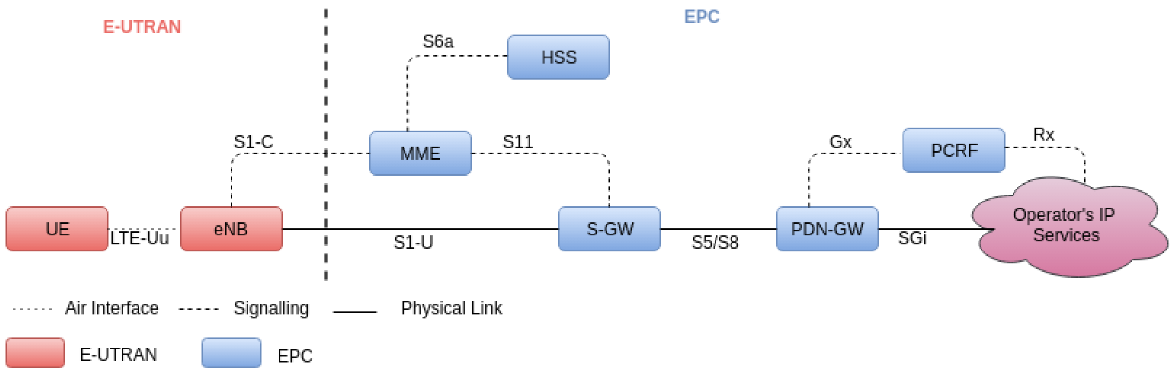


Figure 2.1: EPS Network Architecture

2.1.3. Elements of EPC

EPC is the core network of EPS, responsible of the complete control of the UE and the establishment of the bearers. The logical nodes shown in Figure 2.1 are discussed in more detail below: [24]

Home Subscriber Server (HSS) hosts database that contains the user subscription data of the EPS. It provides user authentication and access authorization, also has the identity of MME to which a user is attached or registered. In addition, HSS holds the information about the PDNs that the user can connect. It is based on the Home Location Register (HLR) and Authentication Centre (AuC).

Mobility Management Entity (MME) is the main element within the control plane of the EPS, it handles the signaling between the UE and the core network (CN). The protocols that are between UE and the CN are known as Non Access Stratum (NAS) protocols. It provides the initial handshake process with the UE through the eNB by verifying the user data in the HSS.

Serving Gateway (SGW) deals with the user plane. It transports all user IP data traffic to the external networks by routing the incoming and outgoing IP packets. It is logically connected to the PDN Gateway.

PDN Gateway (PGW) handles the connection between the EPC and the external IP networks. These networks are called Packet Data Network (PDN). It is the responsible of providing the IP address allocation to the UE, it also provides policy control and charging.

Policy Control and Charging Rules (PCRF) is responsible of making the policy control decisions, and sends Quality of Service (QoS) setting information for each user's subscription profile.

2.1.4. Evolved Universal Terrestrial Radio Access Network (E-UTRAN)

The E-UTRAN represents the access network of the EPS, it handles the traffic between UE and the EPC. The E-UTRAN consists in only one component called Evolved NodeB (eNB), which is connected to the EPC through S1 physical interface. This S1 interface is divided into two parts: one part is called S1-C interface which is the signaling procedure between eNB and MME, and the other part is S1-U interface which is the connection between eNB and SGW. It can also be connected to neighboring eNBs through X2 interface for handover purposes. The E-UTRAN architecture is shown in Figure 2.2.

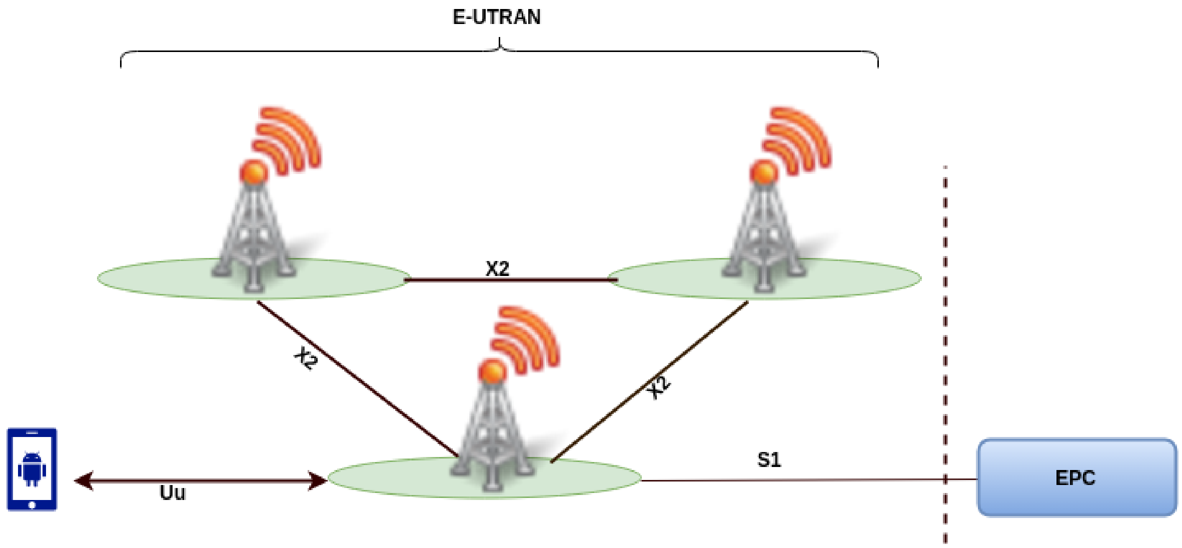


Figure 2.2: E-UTRAN Architecture

2.1.5. Protocol Architecture

User Plane

In user plane, an IP packet for the UE is encapsulated in an EPC using a 3GPP specific protocol. This protocol is called GPRS Tunneling Protocol - User Plane (GTP-U) which is used over the core network interfaces S1 and S5/S8. After the GTP-U encapsulation, the IP packet is tunneled between the PGW and the eNB for transmission to the UE [25]. The overall user plane protocol stack is shown in Figure 2.3.

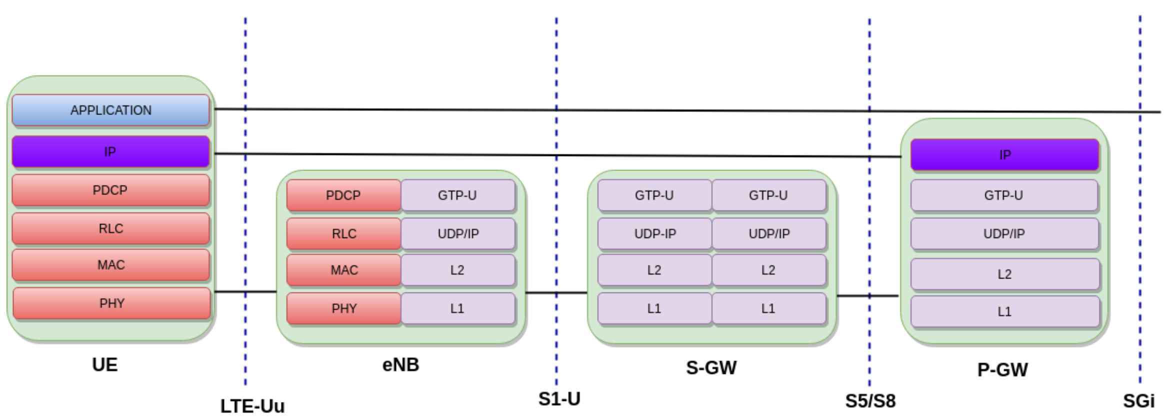


Figure 2.3: EPS User Plane Protocol Stack

Control Plane

The control plane protocol stack between UE and MME is shown in Figure 2.4.

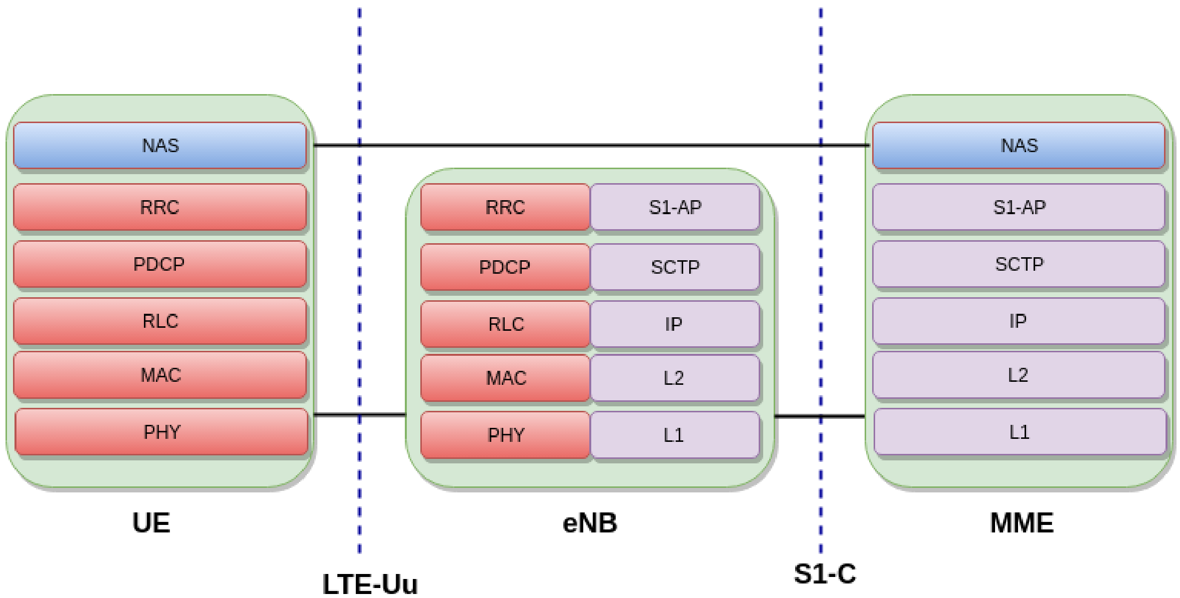


Figure 2.4: UE-MME Control Plane Communication Protocol Stack

2.1.6. Protocol Stack Definition

Physical Layer (PHY) carries all the information from the MAC transport channels over the air interface to physical channels[7]. It also handles the coding/decoding, modulation/demodulation, and multi-antenna processing of the signal. [31].

Medium Access Layer (MAC) is responsible for mapping between logical channels and transport channels, offering a set of logical channels to the RLC sublayer that the MAC multiplexes into the physical layer transport channels[23]. It is also responsible of the scheduling for uplink and downlink, and the data multiplexing from different radio bearers. There is only one MAC entity per UE [31].

Radio Link Control (RLC) is responsible to transport the PDCP's PDUs. It can operate in three different modes: Transparent Mode (TM), Unacknowledged Mode (UM), and

Acknowledged Mode (AM). Depending on the mode of operation, which depends of the reliability provided, can provide HARQ error correction, segmentation/concatenation of PDUs, reordering for in-sequence delivery, duplicate detection, etc. [23][7]

Packet Data Convergence Control (PDCP) is responsible of [7]:

- Header compression and decompression of IP data,
- Transfer of data (user plane or control plane),
- Maintenance of PDCP Sequence Numbers (SNs),
- In-Sequence delivery of upper layer PDUs at re-establishment of lower layers,
- Duplicate packet detection,
- Ciphering and deciphering of user plane data and control plane data,
- Handover data habdling,
- Integrity protection and validation.

Radio Resource Control (RRC) is responsible of [7]:

- The broadcasted system information related to the access stratum (AS) and transport of the non access stratum (NAS) messages.
- Paging, establishment and release of the RRC connection between UE and E-UTRAN.
- Security key management.
- Establishment, configuration, preservation and release of point to point Radio Bearers.
- Handover.
- QoS.

Non Access Stratum (NAS) is a protocol between the UE and the MME on the network side (outside of E-UTRAN). It performs authentication of the UE and security control to establish and maintain IP connectivity between the UE and a PDN GW. [23][7]

GPRS Tunneling Protocol (GTP) is a group of IP-based communications protocols used to carry GPRS in GSM, UMTS and LTE networks. It uses UDP/IP as transport protocol and can be descomposed into separate protocols (GTP-U and GTP-C) [35].

- **GTP-U**: It is used in user plane to carry user data packets in GPRS LTE networks.
- **GTP-C**: It is used in the control plane for signaling (bearer activation, deletion and modification) in GPRS LTE networks.

S1 Application Protocol (S1-AP) is used ont the control plane. The S1-AP messages are sent between the eNB and the MME, this messages are encapsulated by SCTP/IP. S1-AP provides S1 bearer set-up and paging initiation which are signalling services[31].

Stream Control Transmission Protocol (SCTP) ensures the required reliable delivery of the signalling messages. Only one SCTP association is established between one eNB and one MME[31].

2.1.7. LTE Channel Quality Indicator

The channel quality information (CQI), as its name implies, is an indicator carrying the information on how good or bad is the channel quality between UE and the eNB. In OAI project, when eNB runs, it shows the CQI level when an UE is connected. In LTE, there are 15 different CQI value ranging from 0 to 15. Its values and corresponding MCS settings are shown in Table 2.1.

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

Table 2.1: 4-bit CQI Table

2.2. Heterogeneous Networks (HetNets)

Heterogeneous network (Figure 2.5) is the mix of macro cells and small cells to effectively bringing the network closer to the user. In order to reduce the site-to-site distance in the macro-network, smalls cells are introduced to existing macro-eNBs, through the addition of low-power base stations (eNBs, HeNBs, or Relay Nodes(RN)) or Remote Radio Heads (RRH). [32]

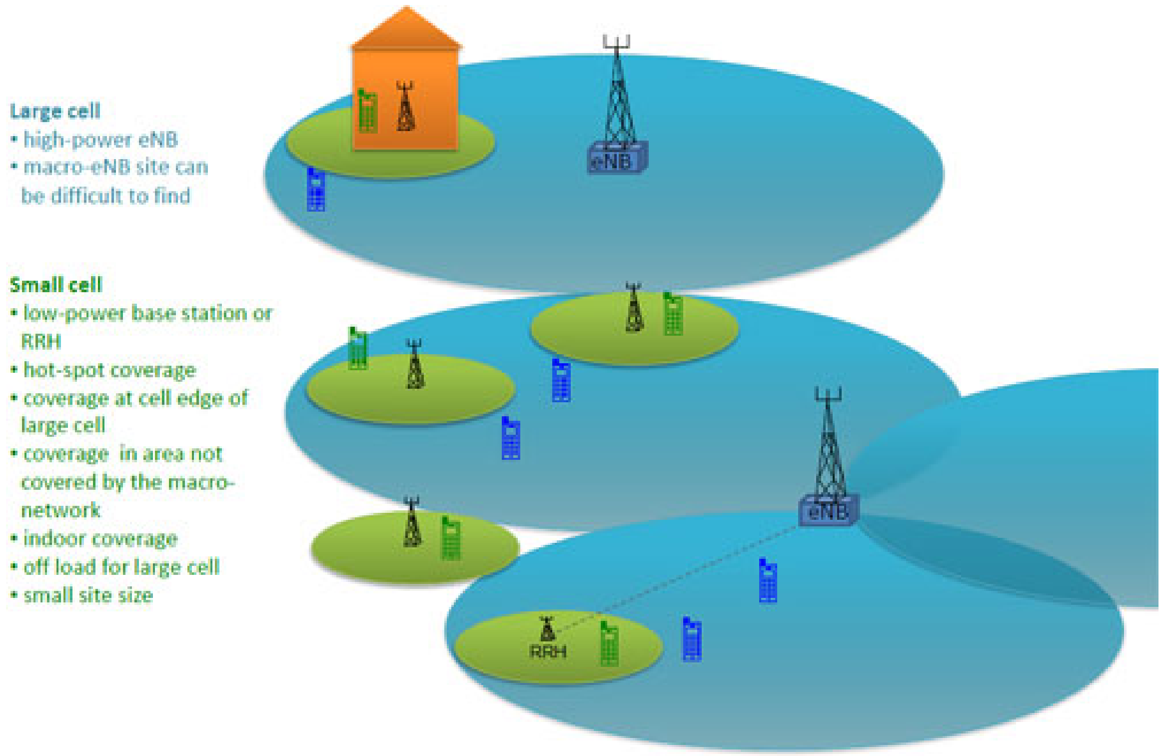


Figure 2.5: Illustration of a HetNet with large and small cells [32]

In HetNets, the cells of different sizes are referred as macro-cells, micro-cells, pico-cells and femto-cells; these cells sizes are listed in order of decreasing base station power. In this list, micro-cells, pico-cells and femto-cells are considered small cells, since they are small compared to the macro cell coverage. The introducing of a heterogeneous network makes the network planning more complex. In a network where the macro-cell and the small-cell use the same frequency channel, the UE connects to the cell with the strongest received DL signal (SSDL). In order to ensure that small cells can serve enough users, the coverage area of the small cell is increased through the use of SSDL offset, this is called Cell Range Extension (CRE). However, the effect of the CRE increase the interference on the DL experienced by the UE that is in the coverage of the small cell. To mitigate this interference problem, a number of features added to the 3GPP LTE can be used. [32]

Inter-cell Interference Coordination (ICIC)

ICIC was introduced in Release 8. In this concept, via the X2 interface, the eNBs can communicate using ICIC in order to mitigate inter-cell interference at the cell edge for UEs. In Release 10, enhanced ICIC (eICIC) was introduced. The principal change is the addition of time domain ICIC through the use of Almost Blank Subframes (ABS), which transmit control channels with reduced power, and allow that the UEs that are at the edge, typically in the CRE region of the small cells, to receive DL information for both control and user data. In Release 11, ICIC was evolved to further enhanced ICIC (feICIC). Here its focus is on the interference handling by the UE through ICIC for control signals, which enables further cell range extension. [32]

Carrier Aggregation (CA)

CA was introduced in Release 10, in order to increase the total bandwidth available to UEs, and thereby increase the bitrate, which is the transferring data measured in bits per second.

Coordinated Multi Point (CoMP)

CoMP is used in order to provide the proper coverage at the cell edges. Here a number of transmission/reception points (signal from two or more base stations) can be coordinated to provide service to the UE, with this information, the UE located in the CRE can use the best UL in the small cell and the best DL in the macro-cell.

2.3. Open Air Interface (OAI) Platform

OAI is developed by OpenAirInterface Software Alliance (OSA) [15], created by Eurecom [6]. It is an open-source hardware and software for wireless technology platforms (simulation, emulation, and realtime) for deployment of a simulate network with high level of realism. OAI provides an open-source-software-based implementation of LTE network (based on release 8 and partially release 10), following the standard protocols of 3GPP for the Access Network and the Core Network. It includes E-UTRAN (eNB and UE) and EPC (MME, HSS, SGW, and PGW), which are shown in Figure 2.6.[10]

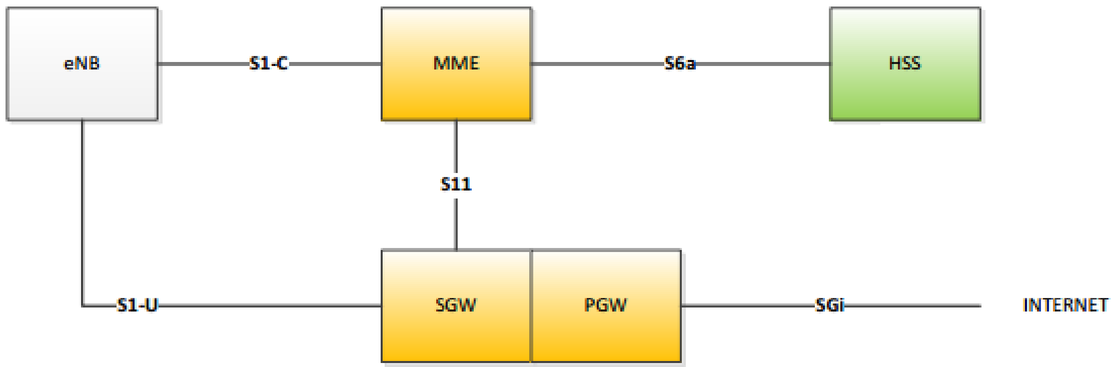


Figure 2.6: OAI LTE Network [10]

As Figure 2.6 shows, in the scenario of OAI EPC, SGW and PGW are merged together into one entity called SPGW, so there is no S5/S8 interface between SGW and PGW, as the conventional EPS.

Software

The source code of the OAI Platform is divided in two parts. The part of the EPC is called openair-cn, and the part of the E-UTRAN is called openairinterface5g. The source code is found in the Git repository of OAI [10], and is organized as follows [8]:

- **Cmake-targets:** Openair build system.
- **Common:** Common code in all layers.
- **Openair1:** Source code for Layer 1.
- **Openair2:** Source code for Layer 2.
- **Openair3:** Middleware code.
- **Openair-cn:** Source code for Core Network.

- **Targets:** Specific code for executables.

Hardware

The software of OAI can be used with standard RF laboratory equipment (low cost devices) for real-world experimentation and validation. In this section will be explained the possible scenarios that can be used with OAI, and the hardware requirements for the setup.

2.3.1. Deployment scenarios

In OAI, different scenarios can deployed as follows [13]:

- Commercial UE <-> OAI eNB + Commercial EPC.
- Commercial UE <-> OAI eNB + OAI EPC.
- Commercial UE <-> Commercial eNB + OAI EPC.
- OAI UE <-> Commercial eNB + OAI EPC (experimental).
- OAI UE <-> Commercial eNB + Commercial EPC (experimental).
- OAI UE <-> OAI eNB + Commercial EPC (experimental).
- OAI UE <-> OAI eNB + OAI EPC.
- OAI UE <-> OAI eNB.

In addition, when OAI eNB and OAI EPC is used, it can be setup on different host or on the same host, but it is recommended to setup them in different host machines due to possible conflicting packages/kernel. In this project, the scenario used is Commercial UE <-> OAI eNB + OAI EPC on different hosts [29].

2.3.2. Host Machines and Processors

According to hardware requirements in OAI [14], the host machines should fulfill certain requirements to be compatible with OAI project. The following are processor families that have been successfully tested in OAI:

- Generation 3/4/5/6 Intel Core i5, i7.
- Generation 2/3/4 Intel Xeon.
- Intel Atom Rangeley, E38xx, x5-z8300.

2.3.3. Supported RF

OAI supports the following SDR hardware:

- NI/Ettus USRP B200/B210 USB3 radio card, requiring a PC with a free USB3 port.
- BladeRF over USB3 port.
- LimeSDR over USB3 port.
- EURECOM EXPRESSMIMO2 PCIe card, requiring a PC with a free 8/16-way PCIe slot.

In this project, we used the USRP B200, which is a low-cost Single Input Single Output (SISO) SDR, that has a frequency range from 700 MHz to 6 GHz [20]. It belongs to Ettus Bus Series, with requirements of USB 3.0 to SuperSpeed transfer samples to eNB. If USB 2.0 is used, the sample transfer rate is very slow and the eNB will stop. The USRP B200 that is used in this project is shown in Figure 2.7.

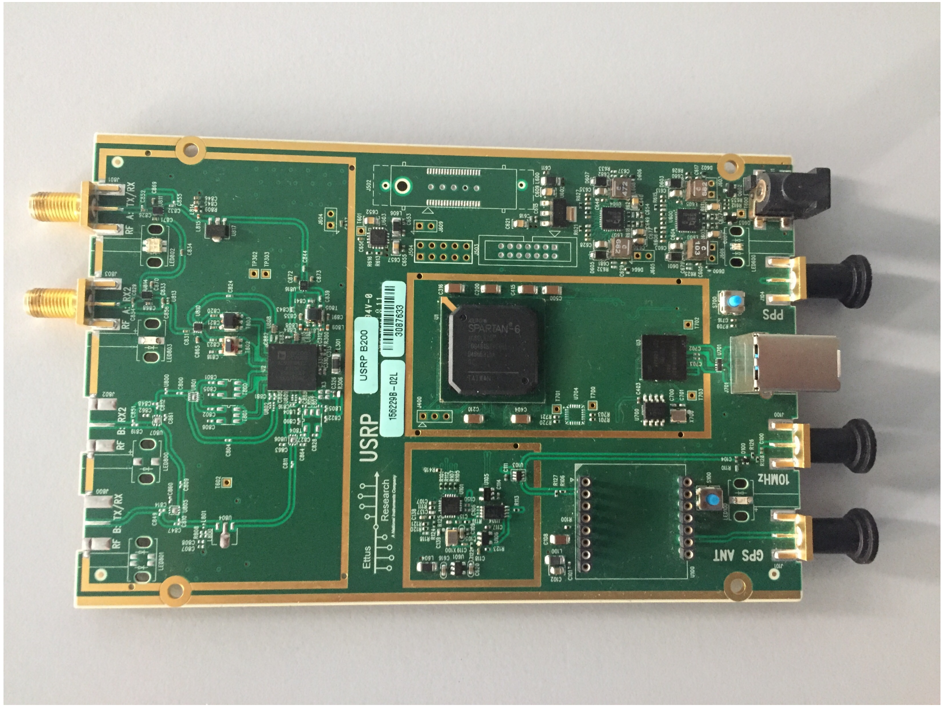


Figure 2.7: USRP B200

2.3.4. User Equipment

In UMTS and 3GPP, a user equipment (UE) can be any device used directly by an end-user to communicate. It can be an OAI UE (software-implementation of a UE), a mobile phone, or a PC/Laptop using a USB Dongle LTE. In this project for mobile phone, we used a Nexus 5 Smartphone (shown in Figure 2.8), and for Dongle LTE we used a Huawei E3372 (shown in Figure 2.9).

Figure 2.8: Nexus 5 Smartphone

Figure 2.9: USB Dongle LTE: Huawei E3372

2.3.5. SIM Card

IMSI

International Mobile Subscriber Identity (IMSI) is a unique code of identification for each user in the mobile network. It is presented as a 15 digit number, the first 3 digits are the Mobile Country Code (MCC), then the next digits are the Mobile Network Code, in this case it depends on the value of the MCC, either 2 digits for European standard or 3 digits for North American standard, in this project the European standard is used, being 2 digits the length of the MCC. The remaining digits are the Mobile Subscription Identification Number (MSIN) which length is 9 or 10 digits depending on the MNC length, in this project the MSIN length is 10 digits. An IMSI code used in this project is shown as an example in the Table 2.2.

MCC	208 (France)
MNC	93 (new MNO MNC)
MSIN	0000000001

Table 2.2: IMSI: 208930000000001

ICCID

A SIM card contains its unique Integrated Circuit Card Identifier (ICCID), it identifies each SIM chip internationally.

IMEI

International Mobile Equipment Identity (IMEI) is a unique number to identify 3GPP mobile phone. It is usually found printed inside the battery compartment of the phone. The IMEI is stored in the HSS to identify valid devices of the users.

Chapter 3

The State of the Art

In this section, we will present solutions proposed in the literature to extend the radio coverage or increase the capacity of LTE networks.

3.1. Evolved User Equipments (eUE) [1]

In this solution, proposed by Apostolaras et al. [1], the eUEs are presented as active network elements to enable reliable multi-hop operation. These eUEs create a virtual link air-interface to be able to forward L2/MAC packets with low latency. The low latency communication is achieved by introducing a full protocol implementation mechanism for MAC/L2 packet forwarding that exploits buffer aware scheduling [1]. The virtual link air-interface extends the classical point-to-point physical links of the radio access networks systems, as shown in Figure 3.1. Basically, eUE stores incoming packets and relay the traffic to the eNB through multi-hop connection [1].

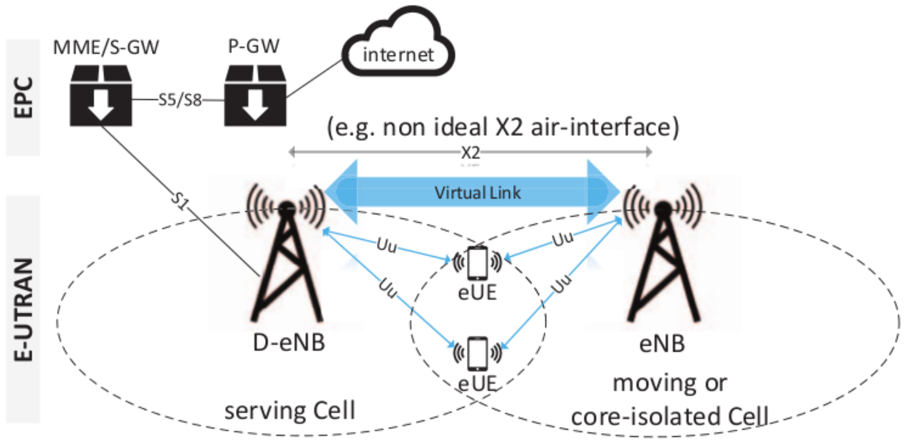


Figure 3.1: LTE Network with Evolved UEs. [1]

3.1.1. Use Cases

In Figure 3.2, it is shown the network topology and the uses cases introduced by the eUE-assisted forwarding.

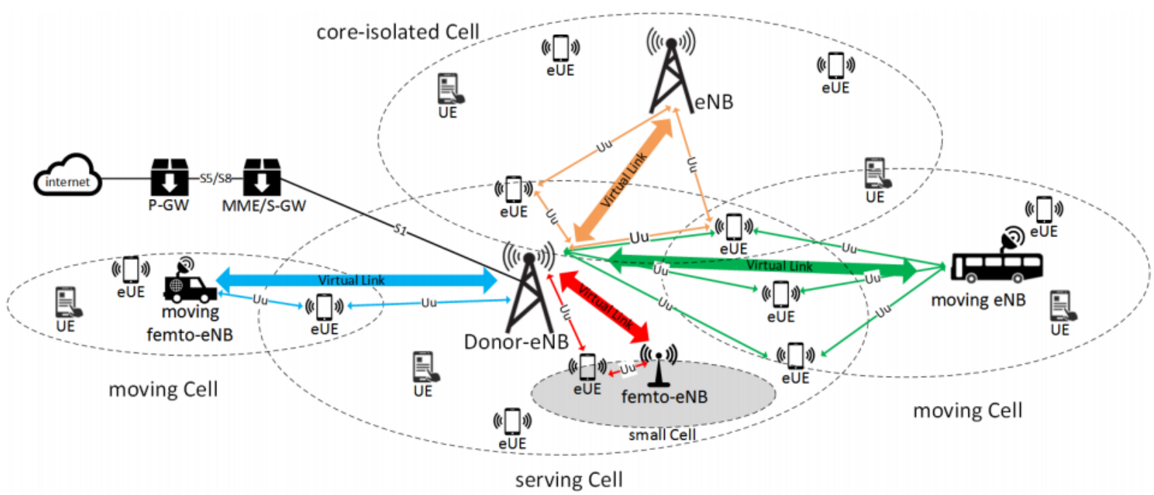


Figure 3.2: Network Topology and Use Cases

- **Core-isolated eNBs**
  - eUEs enable wireless backhauling to core-isolated eNBS.
  - eUEs are enabled as a service by the eNBs to relay traffic.

- **Moving Cells**
    - Public Safety and Private Mobile Radio.
    - Intelligent Transport Systems (ITS) applications.
- **Small Cells**
    - eUEs can communicate to multiple eNBs by realizing a CoMP on the downlink reception.
    - eUEs allows eNBs to re-establish X2-interface.

3.1.2. Performance Evaluation

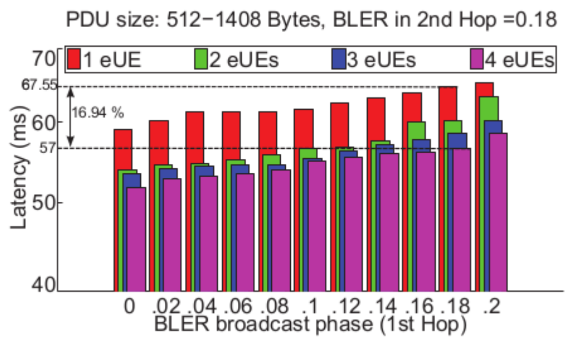
The OpenAirInterface (OAI) [15] was used in order to evaluate the performance, the distributed synchronization procedures and the 3GPP protocol operations for eNBs and eUEs (full implementation code is available online [22]).

The scenario consist of two eNBs and four eUEs located in an area of $500m^2$. The system configuration used by Apostolaras et al. is summarized in Table 3.1.

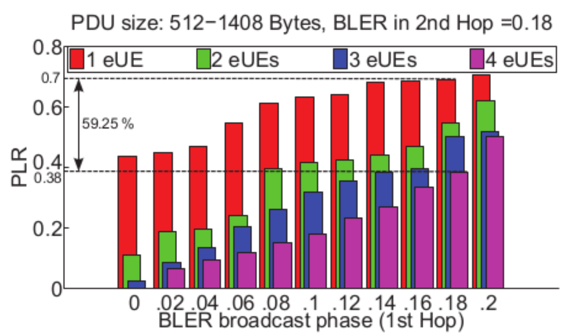
Parameter	**Value**
Carrier Freq.	1.9 GHz
Bandwidth	5MHz
Frame Duration	10ms
TTI	1ms
eUEs	1, 2, 3, 4
Traffic Type	UDP
Fading	AWGN Ch.
Pathloss	-50dB
Pathloss Exp.	2-67
Mobility	Random

Table 3.1: System configuration setup [1]

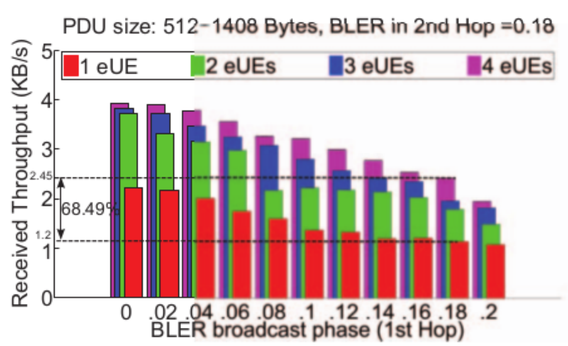
The results of the system configuration of Table 3.1 is shown in Figure 3.3. As the Table 3.1 indicates, they are using AWGN channel as a parameter, and OAISIM is used for the setup of this channel according to OAI, which is a system emulator that allows simulation and emulation of an OpenAirLTE network [16].



(a) Latency.



(b) Packet Loss Rate.



(c) Received Throughput.

Figure 3.3: OAI Measurements Results of LTE using eUEs. [1]

As it is shown in Figure 3.3, when the number of eUEs increases, the latency and packet loss rate (PLR) are reduced, and the throughput increases. In the case of 4 eUEs, in Figure 3.3-a, it is shown that the latency is improved up to 16.94%. Figure 3.3-b shows that the PLR is reduced up to 59.25%, and finally Figure 3.3-c shows that the throughput is improved up to 68.49%. As it is seen, when more eUEs are used to for multiple eNB communication, the performance of the network is improved.

3.2. Enhanced Evolved NodeB (2eNB) [19]

The concept of enhanced evolved NodeB (e2NB) is introduced by Romain Favraud and Navid Nikaein [19], to enable wireless mesh backhaul link between e2NBs. The e2NB uses the existing LTE air interface to establish communication between e2NBs. The difference between an ordinary eNB and a e2NB are its components, shown in Figure 3.4.

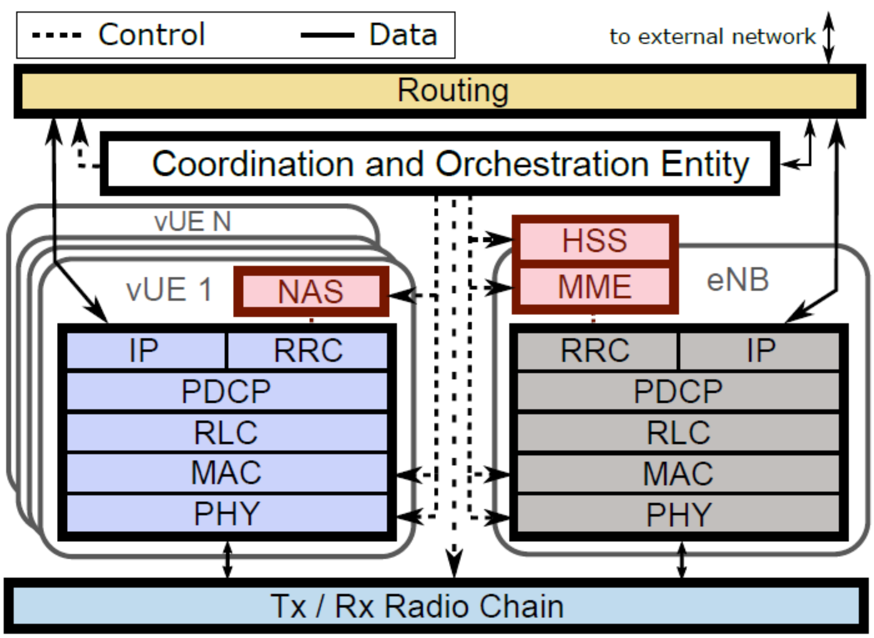


Figure 3.4: Enhanced Evolved NobeB [19]

- **eNB:** Provides the same operations as a normal eNB.
- **MME and HSS:** Allow that the e2NB can work in standalone functionality, and this

components can interact with the embedded eNB [19]. The HSS provides the same operations as a normal HSS, that is the database of authorized users on the network.

- **Virtual UEs:** Establish the communication between e2NBs.
- **Coordination and Orchestration Entity (COE):** It manages the entire life-cycle of vUEs, provides them IMEI and a SIM service to be authenticated by the others e2NBs. It determines the IP address and routing algorithms. It controls the access to the radio required by the embedded eNB and vUEs of the e2NB. [19]
- **Routing and data forwarding:** Enables fast routing and data forwarding of IP packets between eNB and vUE PDCP layer [19]. e2NB can be seen as an endpoint, contrary to normal eNBs.

3.2.1. Performance Evaluation

The OpenAirInterface (OAI) was used to build an emulation platform [13][26]. The network architecture for the connection between two e2NBs is shown in Figure 3.5, where links 1 and 2 are the links between the two e2NBs and their respective vUEs, and links 3 and 4 are the connection with their respective real UEs.

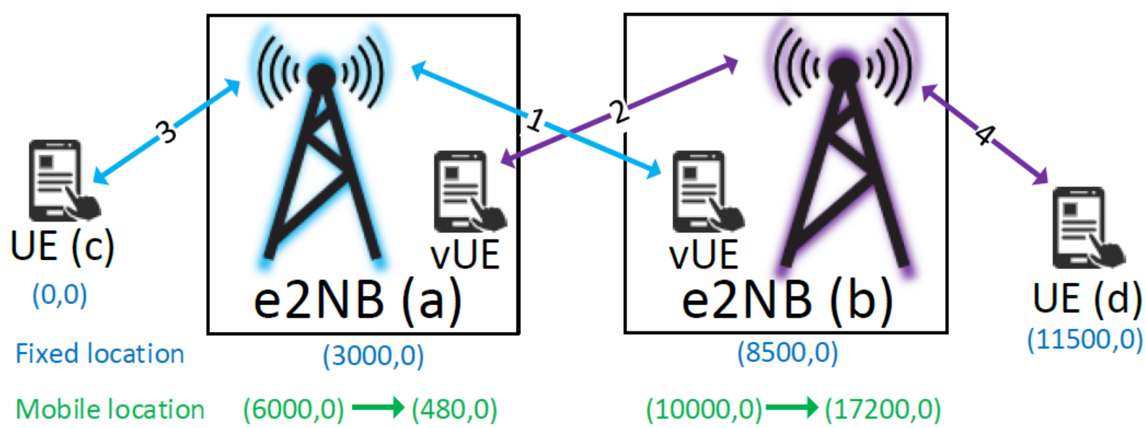


Figure 3.5: Network Architecture e2NB [19]

The network architecture of Figure 3.5 is used to perform three experiments with the emulation parameters shown in Table 3.2.

Parameter	Value
Carrier Freq.	0.9 GHz
Bandwidth	5MHz
Pathloss at 1Km	-91dB
Pathloss Exp.	3
Noise Model	AWGN
Trans. Mode	1
Antenna	Omni 0dBi
Max. Tx Pwr. (dBm)	eNB 24.7 - vUE 23
Max. MCS	DL 26 - UL 16
RLC Mode	UM
RLC reorder timer	35ms
SR Periodicity	2ms - even SF only
Packet IDT	uniform 10-50 ms
Packet size	uniform 64-1408 bytes

Table 3.2: Emulation Parameters [1]

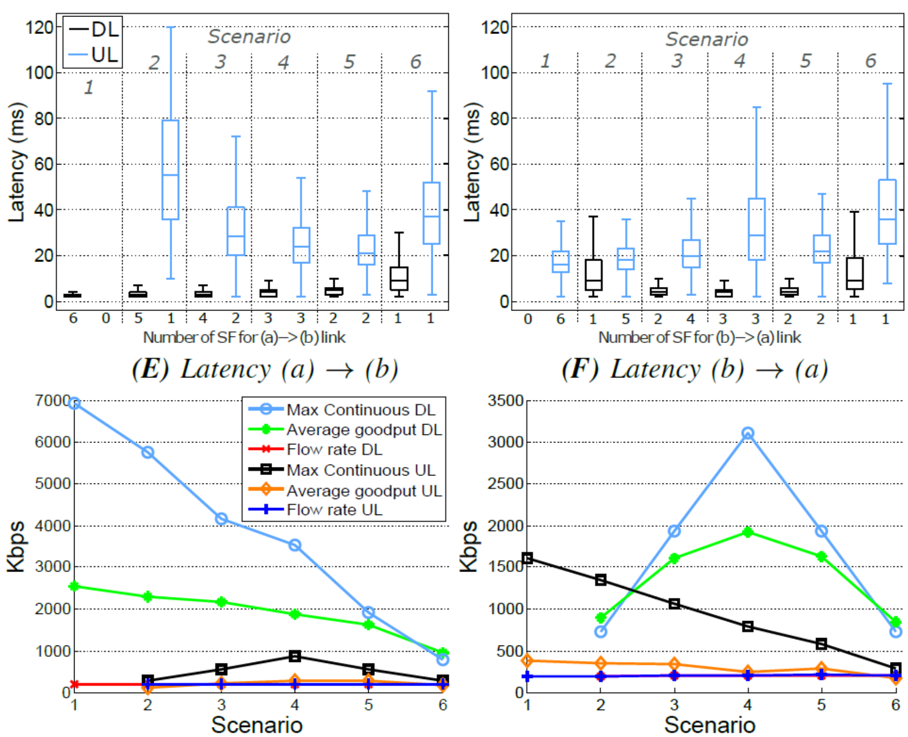
Fixed Cells

In this scenario, the performance of the backhaul link between the two e2NBs (3.5-(a) and 3.5-(b)) is evaluated without any connected UE. Table 3.3 shows the different scenarios, with different subframes (SF) allocation for each link, that were used in this experiment.

Scenario	DL SF (a) –> (b)	DL SF (b) –> (a)	Total DL/UL SFs
1	1, 2, 3, 6, 7, 8	none	6/6
2	1, 2, 6, 7, 8	3	6/6
3	1, 2, 6, 7	3, 8	6/6
4	1, 2, 6	3, 7, 8	6/6
5	1, 6	3, 8	4/4
6	6	8	2/2

Table 3.3: Subframes Allocations [1]

Here each e2NB is connected to each other using their respective vUEs through links 1 and 2. In this experiment, when scenario 1 is used, the only link that is established is the link 1, due to the SF allocation.



(E) Latency (a) → (b)

(F) Latency (b) → (a)

(G) Data rates (a) → (b)

(H) Data rates (b) → (a)

Figure 3.6: Performance of inter-e2NB backhaul link on a fixed scenario [19]

As shown in Figure 3.6, (E) and (F) are box plots that represents the results of the packet latency of UL and DL flows for each scenario. Then (G) and (H) represents the results of average gooput, traffic flow data rate for the variable bit rate (VBR) flow, and the maximum data rate for links from (a) to (b) and from (b) to (a). It can be seen that when the number of available DL or UL SFs increases, the latency, the goodput and the maximum data rate are improved. Nevertheless, it is seen that the latency over a DL path is lower than the latency over the UL path, this means that the DL provides much higher goodput.

Moving Cells

In this experiment, the e2NBs of Figure 3.5 are moving in opposite directions starting from 4Km to 17.2Km of distance, and both with a maximum speed of 20m/s [19]. Both e2NBs reach their destination after 400 seconds. For SF allocation, Table 3.3 (also see Figure 3.5) is used. Here two types of traffic are considered: VBR and VoIP G729. For VBR traffic, the packet size and the IDT is the same as the ones defined in Table 3.2.

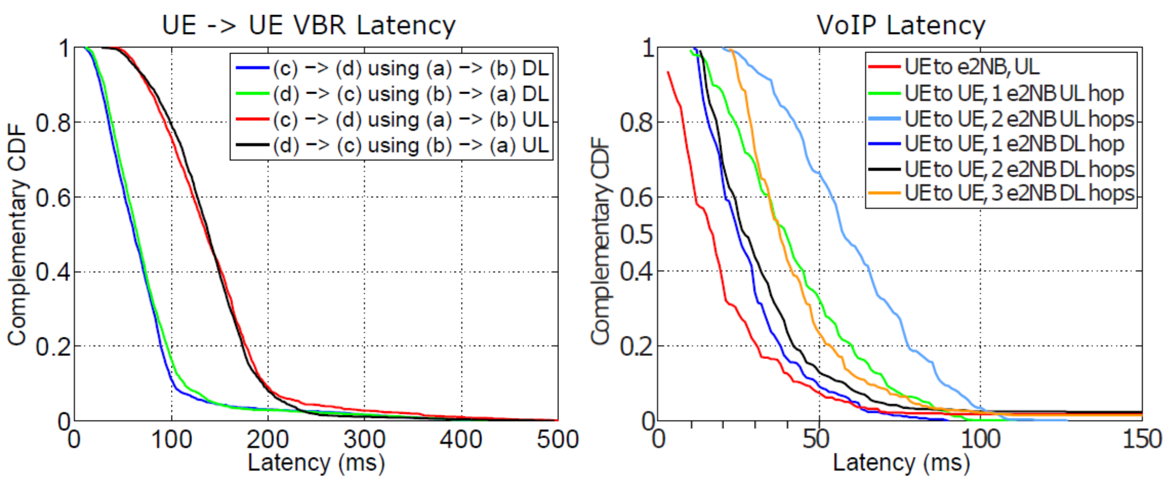


Figure 3.7: Performance of inter-e2NB backhaul link on a moving scenario [19]

It can be seen in Figure 3.7 that the DL path has a lower latency and a better goodput than the UL, especially for VBR traffic. Also, it is seen that the VBR traffic of the DL path after 15.5Km is not good, and the UL path is not good after 10Km. According to Romain Favraud and Navid Nikaein [19], it is caused due to two factors, one is the lack of capacity caused by the adaptive modulation and coding scheme (MCS), and the other is the transmission errors due to channel degradation. In addition, it is seen that using DL or UL path, the LTE QoS requirement of 100ms latency for VoIP is fulfilled throughout the whole experiment [19].

Multi-hop Operation

In this last experiment, the two e2NBs are fixed and connected to UEs (c) and (d) respectively (see Figure 3.5). To forward data to the destination in a multi-hop scenario, static routes are added. In addition, as the experiment of moving cells, VBR and VoIP G729 traffic are generated.

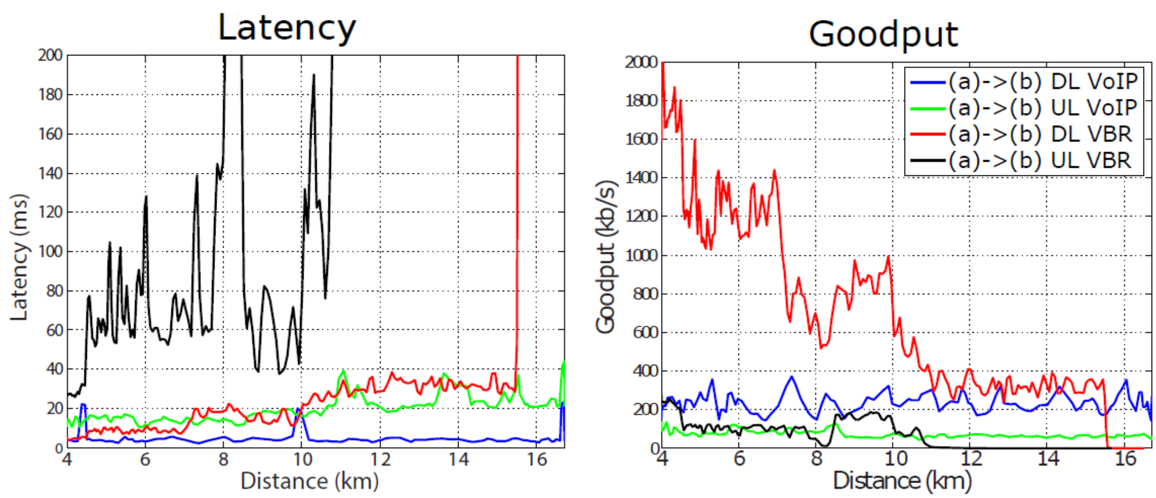


Figure 3.8: Latency of VBR flow and VoIP over multi-hop [19]

In Figure 3.8, it is seen that in VBR end-to-end latency, UL path is almost the double of DL path. And as the previous experiment, it is seen that the LTE QoS requirement of 100ms latency for VoIP is met throughout the whole experiment[19].

3.3. Relay Nodes

In order to improve and extend the radio coverage area or increase the capacity of 4G networks, the concept of relaying was defined for LTE-Advanced in 3GPP Release 10 of the LTE specifications by the introduction of Relay Nodes (RNs) [33][34]. Figure 3.9 illustrates the concept, which includes the following terminology used in 3GPP for a Relay Network:

- **Donor eNB:** The source where the RN receives its signal.
- **Donor cell:** The coverage area of the eNB.
- **Relay Node:** Lower power base station that works as an eNB.
- **Relay cell:** The coverage area of the RN.
- **Backhaul link:** The link between the donor eNB and the RN.
- **Access link:** The link between the RN and the UE.
- **Direct link:** The link between the donor eNB and the UE.

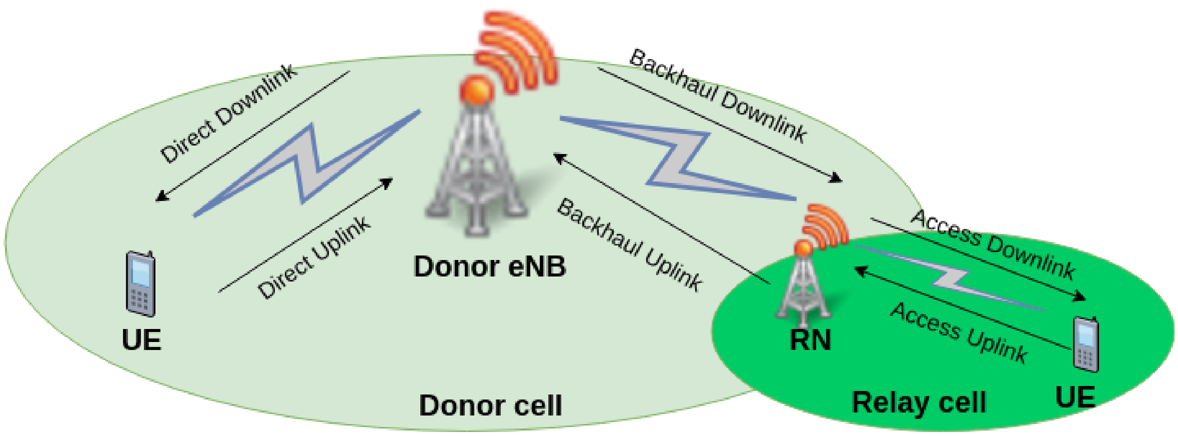


Figure 3.9: Relay Network

As shown in Figure 3.9, the RN is connected to the donor eNB using the existing LTE radio interface to extend the radio coverage of the LTE network.

Relaying strategies can be categorized by the protocol layer functionality of the RN. [17]

3.3.1. Layer 1 RN (Repeater)

In 3GPP Release 8, a relay as a form of repeater was introduced, which is known as wireless repeater or Layer 1 Relay Node (L1 RN) [2]. This repeater receives the signal of the eNB and amplifies it and transmits it again to the destination. The process of the repeater is performed at Layer 1 (physical layer-PHY). L1 relay is a good solution for mitigating coverage holes [2]. It is cost effective and has low latency compared to the other relay scenarios. However, as it amplifies the signal coming from the donor eNB, it also amplifies the noise and the interference, and transmit them together to the destination with the desired signal. In the following figures, the protocol stack for the control plane (Figure 3.10) and user plane (Figure 3.11) for L1 RN solution are shown [17].

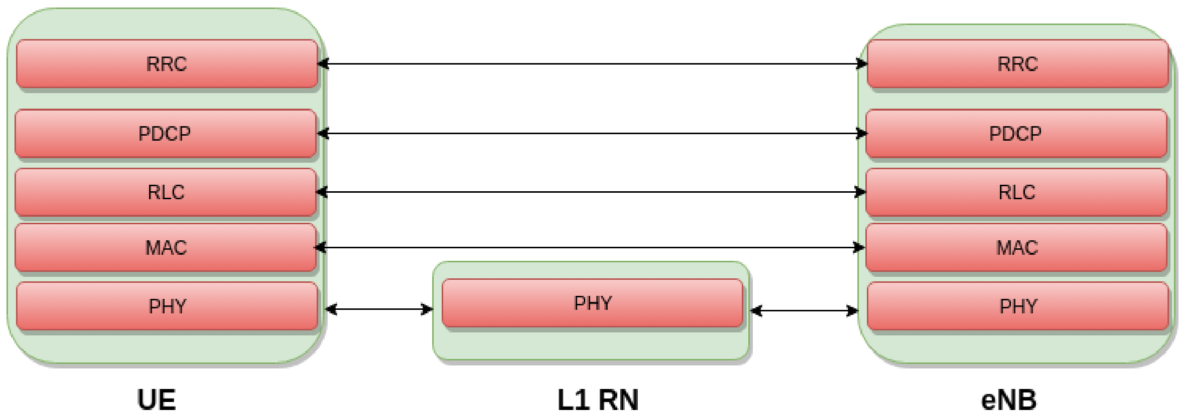


Figure 3.10: Protocol stack L1 RN - Control Plane

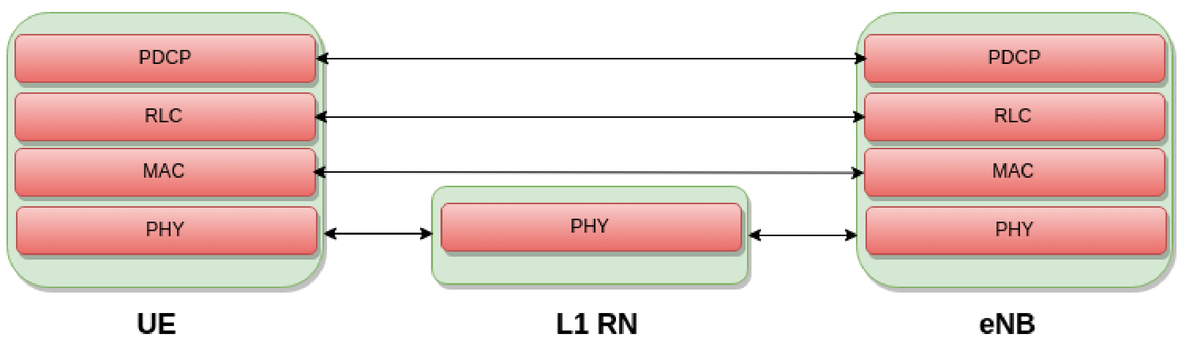


Figure 3.11: Protocol stack L1 RN - User Plane

3.3.2. Layer 2 RN

In Layer 2 Relay Node solution, RN forwards user plane and control plane traffic in the sublayers PDCP, RLC and MAC [2], as shown in Figures 3.12 and 3.13. Here delay is introduced, since the relay decodes and re-encode before sending the received data, but there is no noise in the data that is forwarded by the L2 RN.

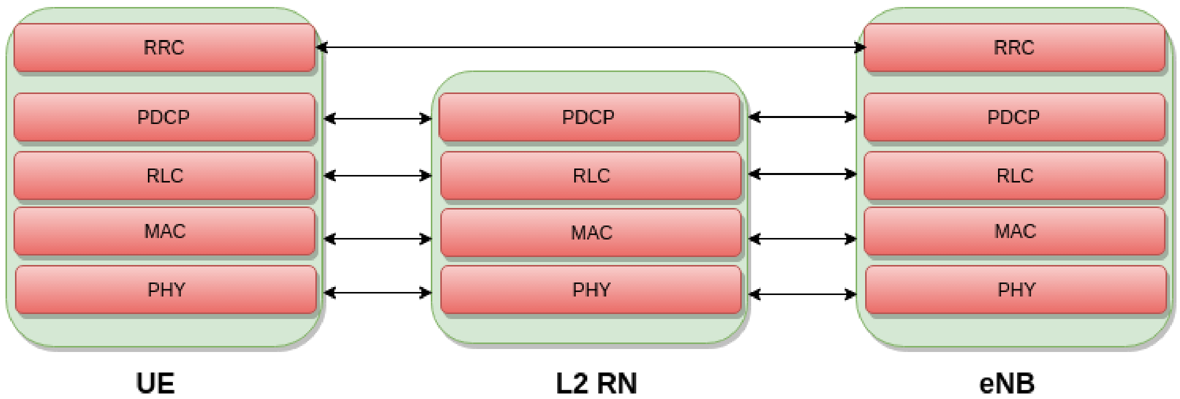


Figure 3.12: Protocol stack L2 RN - Control Plane

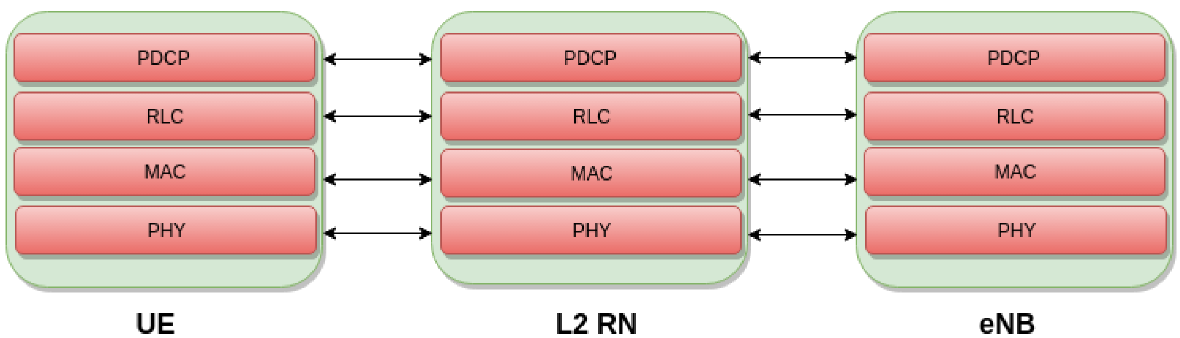


Figure 3.13: Protocol stack L2 RN - User Plane

3.3.3. Layer 3 RN: Wireless Router

In L3 RN, known as wireless router [4], is similar to the L2 RN, but L3 RN also forwards IP packets. A solution in this layer called self-backhauling, was proposed by Hoymman et al. [4]. This self-backhauling relay has the same functionality as the eNB, but it transmits a lower power and has a smaller cell size than the eNB. The relay must support LTE radio interface protocols, since it is connected to the eNB through LTE radio interface for communication [2]. The control plane and user plane traffic forward is shown in Figures 3.14 and 3.15. This solution is also evaluated by A. Mourad in Interdigital [3]. Even though Hoymman and A. Mourad have researched about self-backhauling solutions, they do not have results of an implemented self-backhauling network, only potential solutions for future deployment.

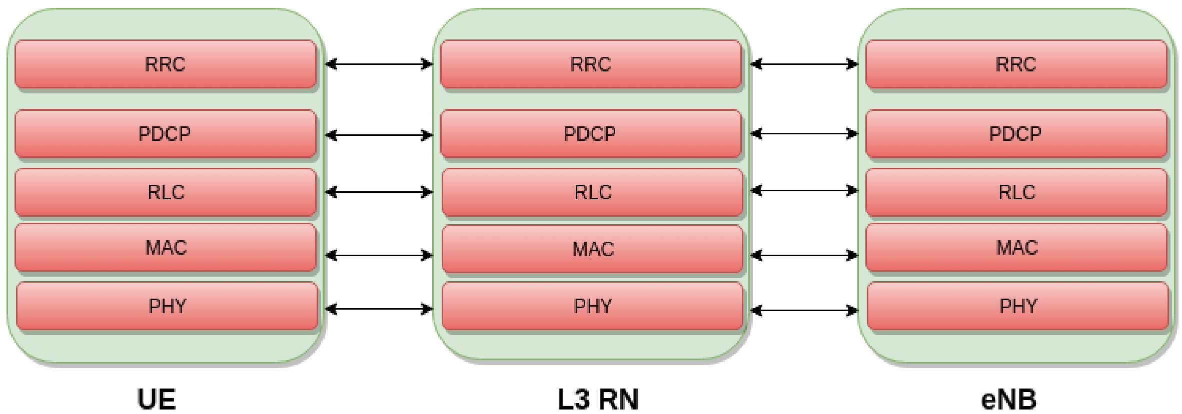


Figure 3.14: Protocol stack L3 Self-Backhauling RN- Control Plane

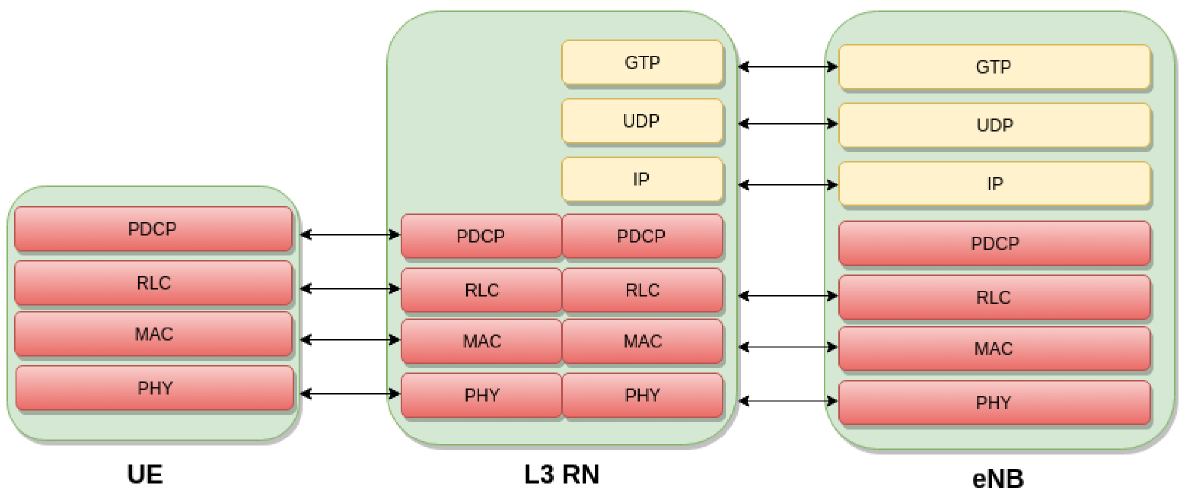


Figure 3.15: Protocol stack L3 RN Self-Backhauling - User Plane

Chapter 4

Implementation

For OAI installation, Linux PCs need to run Ubuntu 14.04 LTS 64-bit. This version is supported and recommended by the OAI project, based on the experience at EURECOM. Other Linux distributions are not recommended because OAI needs lot of packages and is very sensitive to version numbers, linux kernel, etc. The OAI project gives a tutorial for the setup for a conventional LTE network, as explained in Chapter 2. The final scenario that is used in this project is based on the tutorial of Commercial UE <-> OAI eNB + OAI EPC on different hosts [29], which is the setup shown in Figure 4.1. Hence, the installation of eNBs and EPC should be done following this tutorial. In the following, we present the specific configuration and setup details.

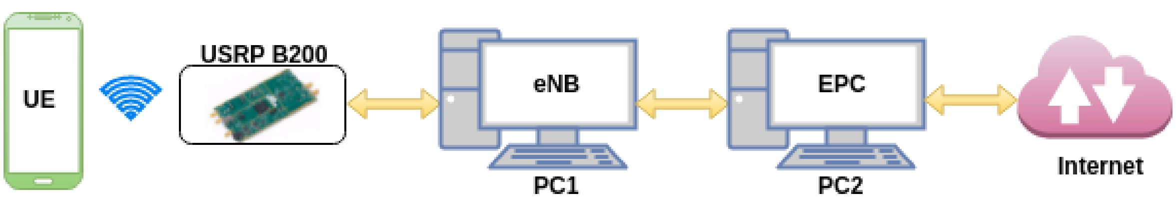


Figure 4.1: Conventional Architecture of LTE Network

4.1. General System Requirements

The Linux PC that is used as an eNB (PC1 in Figure 4.1) needs to perform lots of calculations and receives frames from UEs through USRP B200 connected to USB 3.0 in real time. For this reason, eNB needs a low-latency kernel. The kernel version used for eNB in this project is 3.19.0-61-lowlatency. Then for the EPC (PC2 in Figure 4.1) requires a generic kernel version 4.7.x to create a GTP tunnel for SPGW, this tunnel is used to deliver UE IP packets coming from eNB towards P-GW. The kernel version used for EPC in this project is 4.7.1.

4.2. Conventional LTE Architecture Setup

For conventional LTE architecture setup, as shown in Figure 4.1, we use PC1 for eNB and PC2 for EPC, which are connected through an Ethernet cable, and the connection between PC1 and USRP B200 is via USB 3.0 for real time data traffic purposes. For the installation of the OAI software, it can be obtained from the EURECOM GitLab server as mentioned in Chapter 2 [10]. It is important to use the correct branch, there are two main branches (master and develop), the develop branch is the one recommended by the OAI team. The detailed setup for conventional LTE is shown in Figure 4.2.

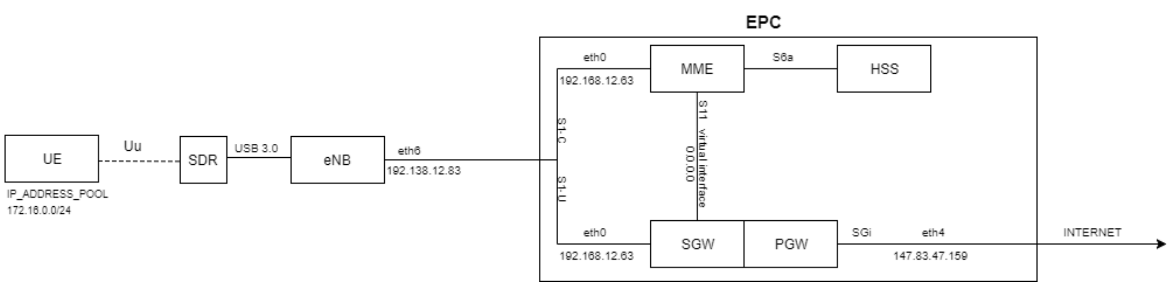


Figure 4.2: Detailed Conventional LTE setup with ethernet interfaces and IP addresses

The UE SIM card used in this project is configured with the information shown in Table 4.1.

MCC (Mobile Country Code)	208
MNC (Mobile Network Code)	93
TAC (Tracking Area Code)	1
IMSI (International Mobile Subscriber Identity)	208930000000001
OP (Operator Key)	11111111111111111111111111111111 (32 digits)

Table 4.1: UE SIM card configuration

The configuration files for eNB, HSS, MME and SPGW have to be modified in order to setup the LTE network. For the configuration of the eNB, the parameters to be changed in the file "enb.band7.tm1.usrpb210.conf" are shown in Table 4.2.

eNB_ID	0xe00;
downlink_frequency	2670000000L;
N_RB_DL	25;
tx_gain	90;
rx_gain	125;
tracking_area_code	"1";
mobile_country_code	"208";
mobile_network_code	"93";
mme_ip_address (ipv4)	"192.168.12.63";
ENB_INTERFACE_NAME_FOR_S1_MME	"eth6";
ENB_IPV4_ADDRESS_FOR_S1_MME	"192.168.12.83/24";
ENB_INTERFACE_NAME_FOR_S1_MME	"eth6";
ENB_INTERFACE_NAME_FOR_S1_MME	"192.168.12.83/24";
ENB_PORT_FOR_S1U	2152;

Table 4.2: eNB parameters to be changed for conventional LTE network

For the configuration of the EPC, the parameters of HSS, MME, and SPGW to be changed are shown in following subsections.

HSS

In the HSS, as it is a database, MySQL is installed during the HSS build. Here two files have to be configured: “hss.conf” and “hss_fd.conf”;. These parameters that have to be changed are shown in Table 4.3 and Table 4.4 respectively.

MYSQL_server	“127.0.0.1”;
MYSQL_user	“root”;
MYSQL_pass	“linux”;
MYSQL_db	“oai_db”;
OPERATOR_key	“11111111111111111111111111111111”;

Table 4.3: Parameters to be changed in “hss.conf”;

Identity	“hss.openair4G.eur”;
Realm	“openair4G.eur”;

Table 4.4: Parameters to be changed in “hss_fd.conf”;

MME

In the MME, like HSS, two files have to be configured: “mme.conf” and “mme_fd.conf”. These parameters that have to be changed are shown in Table 4.5 and Table 4.6 respectively.

MAXENB	2; #power of 2
MAXUE	16; #power of 2
GUMMEI_LIST	({MCC="208" ; MNC="92"; MME_GID="4" ; MME_CODE="1"; });
TAI_LIST	({MCC="208" ; MNC="93"; TAC = "1"; });
MME_INTERFACE_NAME_FOR_S1_MME	"eth3";
MME_IPV4_ADDRESS_FOR_S1_MME	"192.168.12.63/24";
MME_INTERFACE_NAME_FOR_S11_MME	"lo";
MME_IPV4_ADDRESS_FOR_S11_MME	"127.0.11.1/8";
MME_PORT_FOR_S11_MME	2123;
SGW_IPV4_ADDRESS_FOR_S11	"127.0.11.2/8";

Table 4.5: Parameters to be changed in "mme.conf"

Identity	"younes.openair4G.eur";
Realm	"openair4G.eur";
ConnectPeer	"hss.openair4G.eur" { ConnectTo = "127.0.0.1"; No_SCTP ; No_IPv6; Prefer_TCP; No_TLS; port = 3868; realm = "openair4G.eur";};

Table 4.6: Parameters to be changed in "mme_fd.conf"

SPGW

As mentioned in Chapter 2, S-GW and P-GW are merged together, and the configuration file that has to be configured is "spgw.conf". The parameters to be changed are shown in Table 4.7 and Table 4.8.

SGW_INTERFACE_NAME_FOR_S11	"lo";
SGW_IPV4_ADDRESS_FOR_S11	"127.0.11.2/8";
SGW_INTERFACE_NAME_FOR_S1U_S12_S4_UP	"eth3";
SGW_IPV4_ADDRESS_FOR_S1U_S12_S4_UP	"192.168.12.63/24";
SGW_IPV4_PORT_FOR_S1U_S12_S4_UP	2152;
SGW_INTERFACE_NAME_FOR_S5_S8_UP	"none";
SGW_IPV4_ADDRESS_FOR_S5_S8_UP	"0.0.0.0/24";

Table 4.7: Parameters of S-GW to be changed in "spgw.conf"

PGW_INTERFACE_NAME_FOR_S5_S8	"none";
PGW_IPV4_ADDRESS_FOR_S5_S8	"0.0.0.0/24";
PGW_INTERFACE_NAME_FOR_SGI	"eth4";
PGW_IPV4_ADDRESS_FOR_SGI	"147.83.47.159/24";
PGW_MASQUERADE_SGI	"yes";
UE_TCP_MSS_CLAMPING	"no";
IPV4_LIST	("172.16.0.0/24";);
DEFAULT_DNS_IPV4_ADDRESS	"8.8.8.8";
DEFAULT_DNS_SEC_IPV4_ADDRESS	"8.8.4.4";
UE_MTU	1400;

Table 4.8: Parameters of P-GW to be changed in "spgw.conf";

After all the changes done in the configuration files, the user needs to be registered in the database (HSS). To do that, we use MySQL to add the user to the user table "oai_db.users";, and also the tables "oai_db.mmeidentity" and "oai_db.pdn" have to be updated.

Then HSS, MME, and SPGW are compiled and run. To have a successful connection between MME and HSS, in the terminal "STATE_OPEN" message has to be shown as illustrated in Figure 4.3. SPGW creates a GTP tunnel to connect the eNB and the P-GW, and this tunnel can be seen through the command "ifconfig" as an interface called "gtp0" (Figure 4.4).

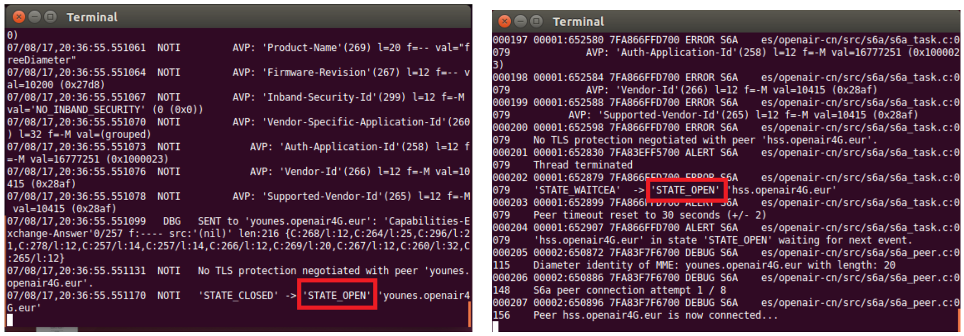


Figure 4.3: State Open on HSS (left) and MME (rigth) terminal

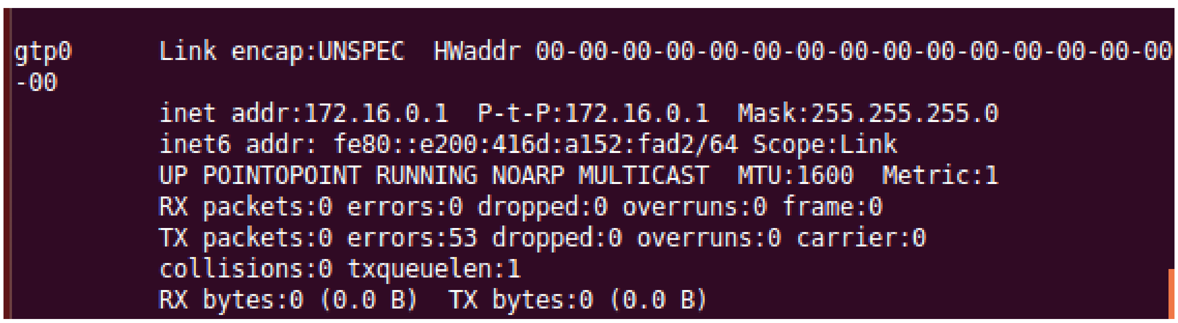


Figure 4.4: gtp0 interface

When eNB is compiled and run, MME will show the established connection between eNB and EPC by indicating the local address of the EPC and the remote peer address of the eNB, and also number of eNBs connected in the EPC as shown in Figure 4.5 and Figure 4.6 respectively.

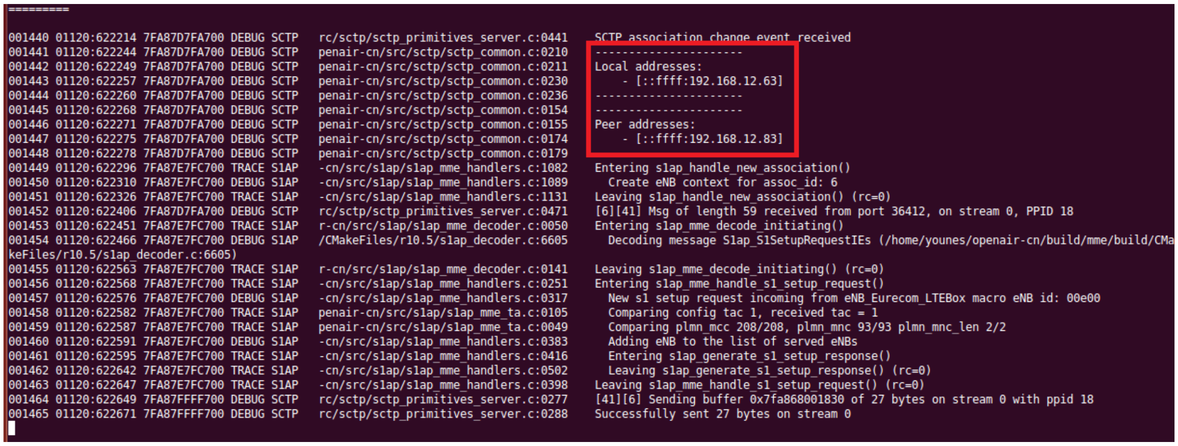


Figure 4.5: Addresses of eNB and EPC on MME terminal

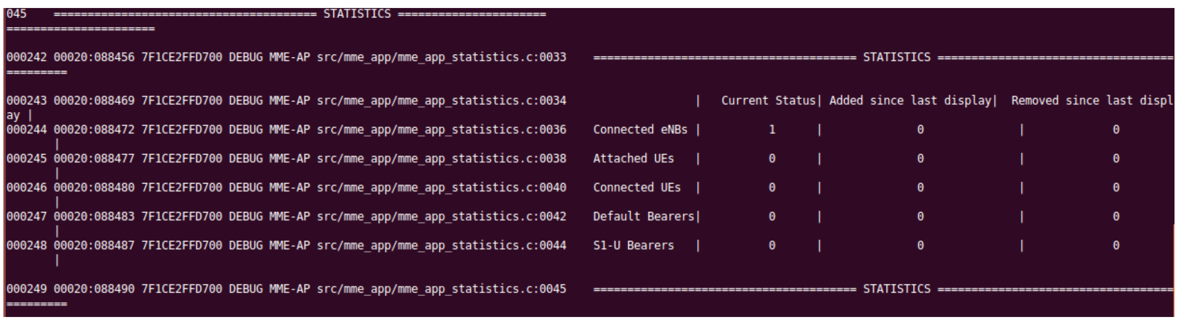


Figure 4.6: Number of eNBs connnected to the MME

To connect the UE to the LTE network, the APN information in the UE (Nexus 5) has to be changed with the parameters that are shown in Table 4.9 in the path **Settings -> Mobile Network Settings -> Access Point Names -> Add a new apn**.

Name	"eur"
APN field	"oai.ipv4"
Bearer	"LTE"

Table 4.9: Parameters of APN to be changed in UE

After configuring the UE and connected to the LTE network, the status of Figure 4.6 will change to the one shown in Figure 4.7. And the UE will indicate successful connection to the

LTE network as shown in Figure 4.8

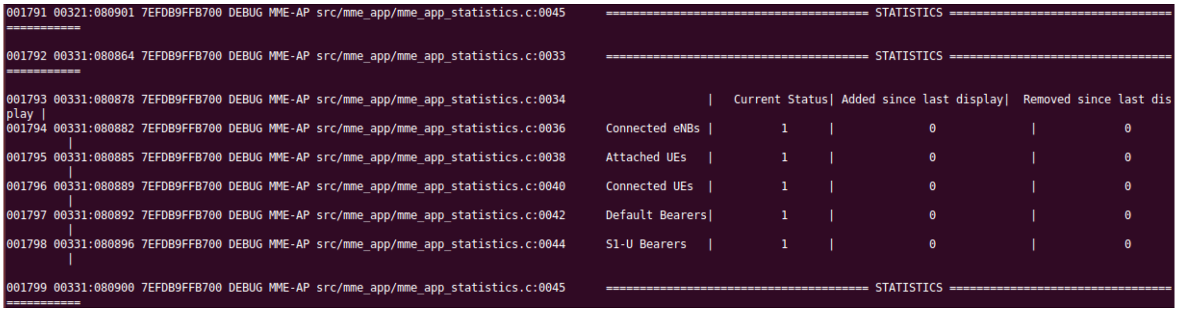


Figure 4.7: Number of UEs and eNBs connected to the MME

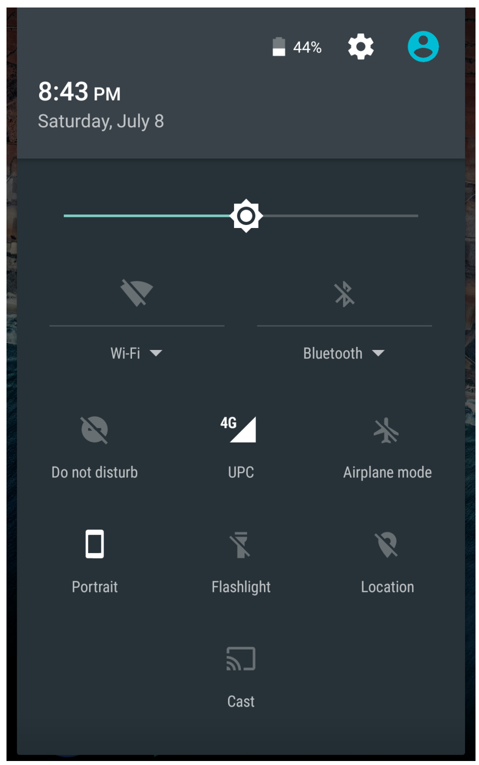


Figure 4.8: UE connected to the LTE network

When Dongle LTE is used as a UE, the configuration of the APN can be changed through the graphical interface that can be accessed using an Internet browser with the IP address 192.168.8.1 as shown in Figure 4.9. And the successful connection to the LTE network is shown in Figure 4.10

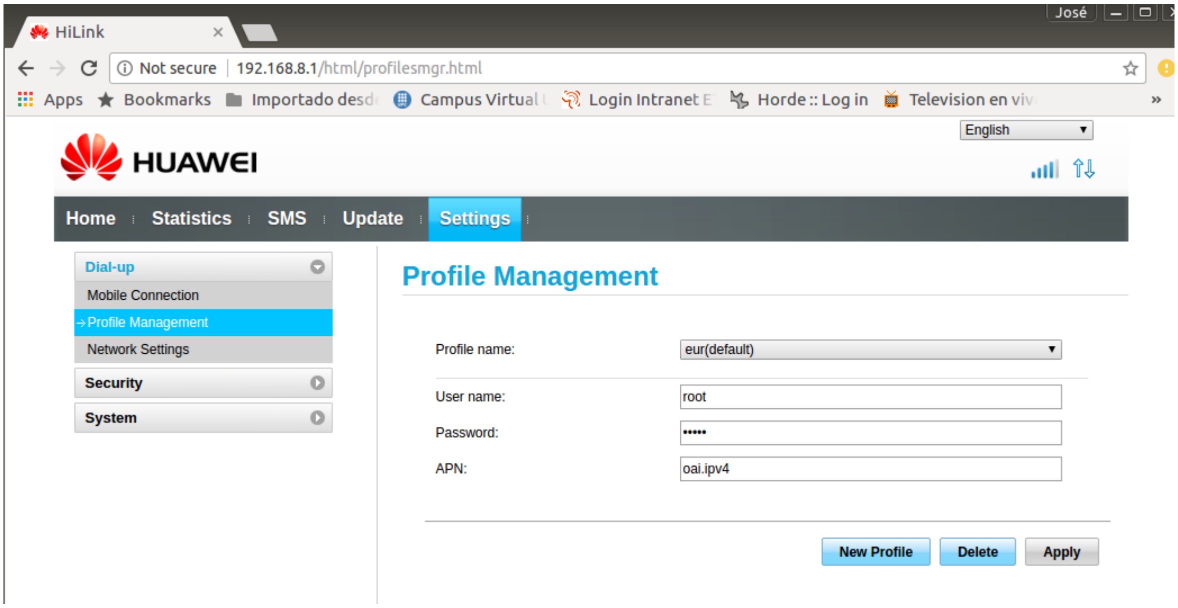


Figure 4.9: APN configuration Dongle LTE

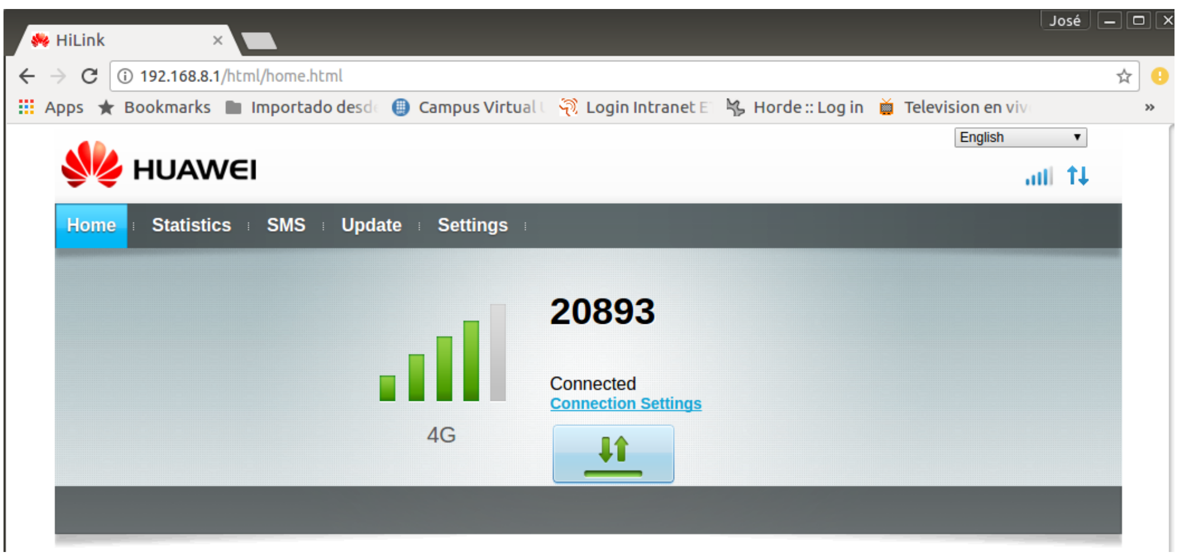


Figure 4.10: Successful connection Dongle LTE

4.3. Self-Backhauling Architecture Setup

For the self-backhauling architecture, we used the previous traditional LTE architecture setup for the A-eNB, which is the following: a low cost USRP B200 (Universal Software Defined Radio) is used and connected to a PC1 that works as an A-eNB, this A-eNB is connected via Ethernet to PC2 that works as an EPC. For the Self-Backhauled network, a second USRP B200 is used and connected to a PC3 that works as the B-eNB. The B-eNB is connected to a dongle LTE, which allows the B-eNB to connect to the A-eNB network as an user equipment, to provides the backhaul link to the EPC. And finally an UE is connected via LTE radio interface through the USRP B200 that is connected to the B-eNB. This implementation is shown in Figure 4.11.

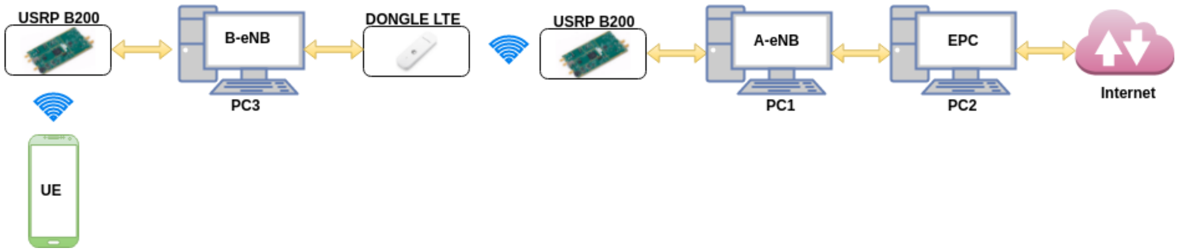


Figure 4.11: Self-Backhauling First Setup with OAI project

The detailed setup of the self-backhauling network is shown in Figures 4.12 and 4.13.

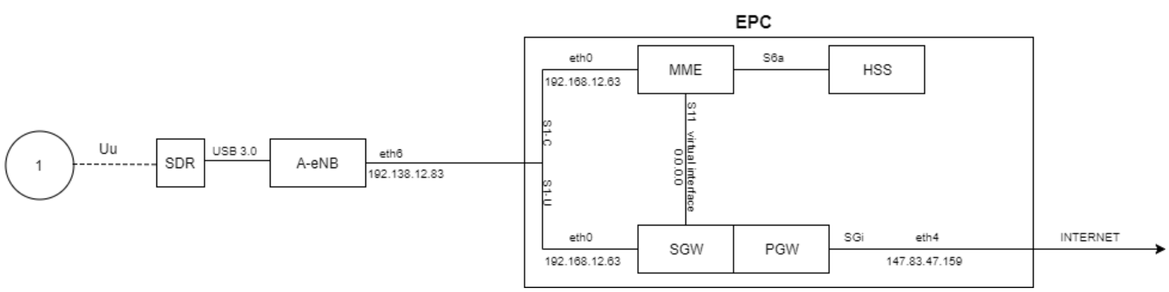


Figure 4.12: Detailed Traditional Network part of the Self-Backhauling network architecture

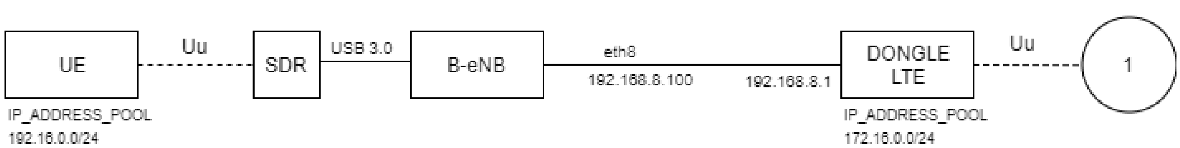


Figure 4.13: Detailed Self-Backhauling Network

B-eNB

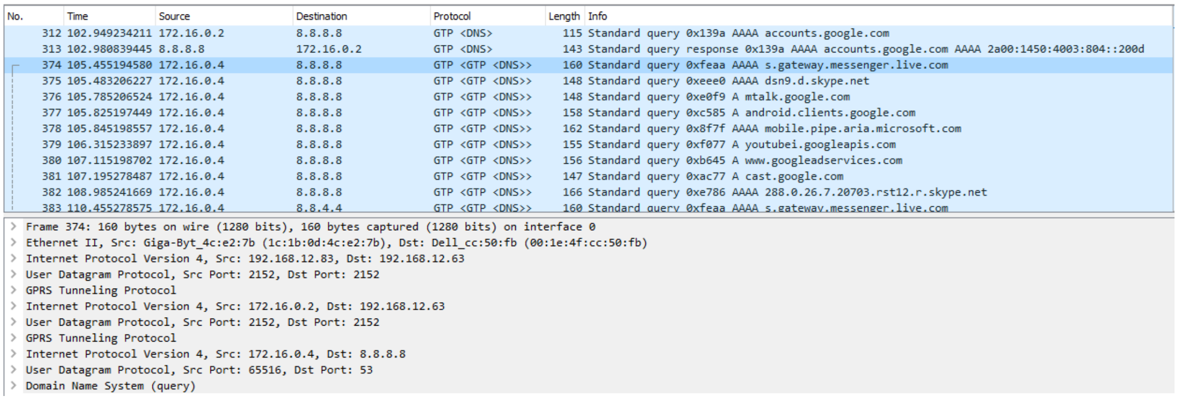
eNB_ID	0xe10;
downlink_frequency	2640000000L;
N_RB_DL	25;
tx_gain	90;
rx_gain	125;
tracking_area_code	“1”;
mobile_country_code	“208”;
mobile_network_code	“93”;
mme_ip_address (ipv4)	“192.168.12.63”;
ENB_INTERFACE_NAME_FOR_S1_MME	“eth8”;
ENB_IPV4_ADDRESS_FOR_S1_MME	“192.168.8.100/24”;
ENB_INTERFACE_NAME_FOR_S1_MME	“eth8”;
ENB_INTERFACE_NAME_FOR_S1_MME	“192.168.8.100/24”;
ENB_PORT_FOR_S1U	2152;

Table 4.10: B-eNB parameters to be changed for Self-Backhauling LTE network First Setup

As it is showed in Table 4.10, there are some parameters that are different from the one used in Table 4.2, such as "eNB_ID" which must be different from the eNB from conventional LTE network to not have any conflict with the IDs, another parameters are the IPs and interfaces which are configurated to use the ones given to the B-eNB by the Dongle LTE to route the incoming and outcoming packets.

4.4. Double GTP encapsulation problem for Self-Backhauling

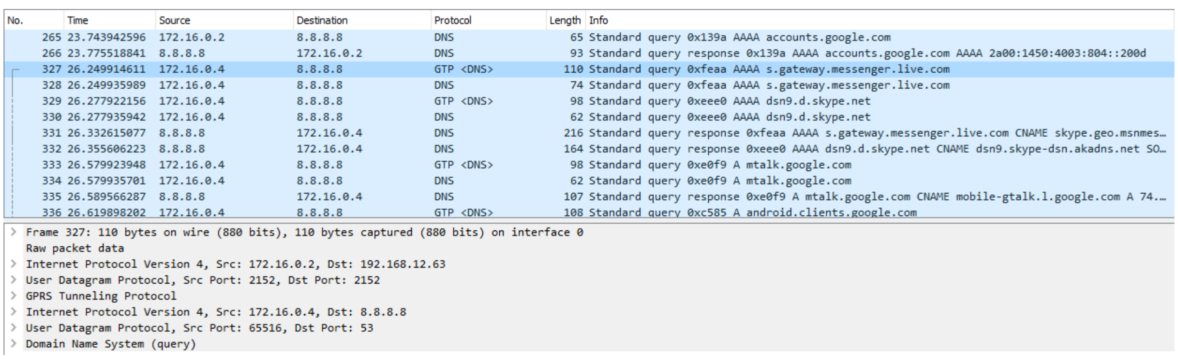
After compiling and running all the setups, a problem was found, the UE was not able to connect to the Internet. After detailed investigation, it was found that since B-eNB does the GTP encapsulation of the data received from UE to send the data to A-eNB, and the A-eNB does not recognize which packets are from a normal UE or from B-eNB, A-eNB was realizing another GTP encapsulation of the incoming packets from B-eNB, and we obtained a double GTP encapsulation that is shown in the wireshark captures in Figures 4.14, 4.15 and 4.16 .



No.	Time	Source	Destination	Protocol	Length	Info
312	102.949234211	172.16.0.2	8.8.8.8	GTP <DNS>	115	Standard query 0x139a AAAA accounts.google.com
313	102.980839445	8.8.8.8	172.16.0.2	GTP <DNS>	143	Standard query response 0x139a AAAA accounts.google.com AAAA 2a00:1450:4003:804::200d
374	105.455194580	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	160	Standard query 0xfeaa AAAA s.gateway.messenger.live.com
375	105.483206227	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	148	Standard query 0xeee0 AAAA dsn9.d.skype.net
376	105.785206524	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	148	Standard query 0xe0f9 A mtalk.google.com
377	105.825197449	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	158	Standard query 0xc585 A android.clients.google.com
378	105.845198557	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	162	Standard query 0x8f7f AAAA mobile.pipe.aria.microsoft.com
379	106.315233897	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	155	Standard query 0xf077 A youtubei.googleapis.com
380	107.115198702	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	156	Standard query 0xb645 A www.googleadservices.com
381	107.195278487	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	147	Standard query 0xac77 A cast.google.com
382	108.985241669	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	166	Standard query 0xe786 AAAA 288.0.26.7.20703.rst12.r.skype.net
383	110.455278575	172.16.0.4	8.8.4.4	GTP <GTP <DNS>>	160	Standard query 0xfeaa AAAA s.gateway.messenger.live.com

Frame 374: 160 bytes on wire (1280 bits), 160 bytes captured (1280 bits) on interface 0
Ethernet II, Src: Giga-Byt_4c:e2:7b (1c:1b:0d:4c:e2:7b), Dst: Dell_cc:50:fb (00:1e:4f:cc:50:fb)
Internet Protocol Version 4, Src: 192.168.12.83, Dst: 192.168.12.63
User Datagram Protocol, Src Port: 2152, Dst Port: 2152
GPRS Tunneling Protocol
Internet Protocol Version 4, Src: 172.16.0.2, Dst: 192.168.12.63
User Datagram Protocol, Src Port: 2152, Dst Port: 2152
GPRS Tunneling Protocol
Internet Protocol Version 4, Src: 172.16.0.4, Dst: 8.8.8.8
User Datagram Protocol, Src Port: 65516, Dst Port: 53
Domain Name System (query)

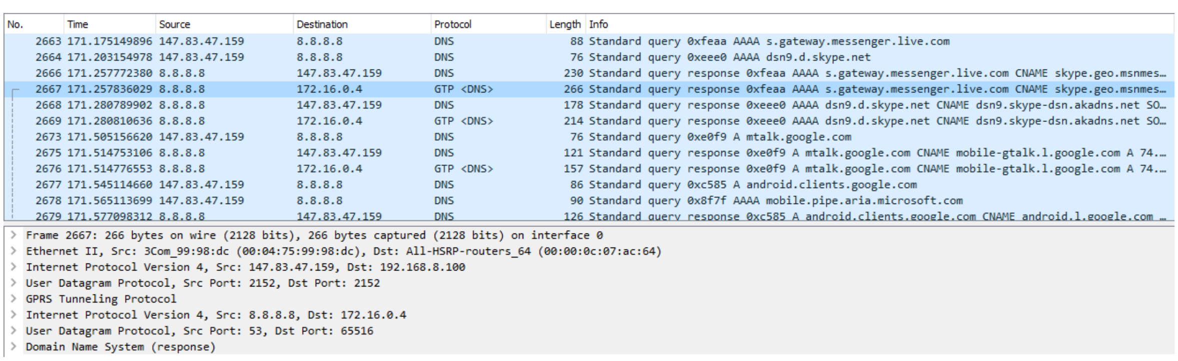
Figure 4.14: Wireshark capture of incoming packets in S-GW



No.	Time	Source	Destination	Protocol	Length	Info
265	23.743942596	172.16.0.2	8.8.8.8	DNS	65	Standard query 0x139a AAAA accounts.google.com
266	23.775518841	8.8.8.8	172.16.0.2	DNS	93	Standard query response 0x139a AAAA accounts.google.com AAAA 2a00:1450:4003:804::200d
327	26.249914611	172.16.0.4	8.8.8.8	GTP <DNS>	110	Standard query 0xfeaa AAAA s.gateway.messenger.live.com
328	26.249935989	172.16.0.4	8.8.8.8	DNS	74	Standard query 0xfeaa AAAA s.gateway.messenger.live.com
329	26.277922156	172.16.0.4	8.8.8.8	GTP <DNS>	98	Standard query 0xeee0 AAAA dsn9.d.skype.net
330	26.277935942	172.16.0.4	8.8.8.8	DNS	62	Standard query 0xeee0 AAAA dsn9.d.skype.net
331	26.332615077	8.8.8.8	172.16.0.4	DNS	216	Standard query response 0xfeaa AAAA s.gateway.messenger.live.com CNAME skype.geo.msnmes…
332	26.355606223	8.8.8.8	172.16.0.4	DNS	164	Standard query response 0xeee0 AAAA dsn9.d.skype.net CNAME dsn9.skype-dsn.akadns.net SO…
333	26.579923948	172.16.0.4	8.8.8.8	GTP <DNS>	98	Standard query 0xe0f9 A mtalk.google.com
334	26.579935701	172.16.0.4	8.8.8.8	DNS	62	Standard query 0xe0f9 A mtalk.google.com
335	26.589566287	8.8.8.8	172.16.0.4	DNS	107	Standard query response 0xe0f9 A mtalk.google.com CNAME mobile-gtalk.l.google.com A 74.…
336	26.619898202	172.16.0.4	8.8.8.8	GTP <DNS>	108	Standard query 0xc585 A android.clients.google.com

Frame 327: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
Raw packet data
Internet Protocol Version 4, Src: 172.16.0.2, Dst: 192.168.12.63
User Datagram Protocol, Src Port: 2152, Dst Port: 2152
GPRS Tunneling Protocol
Internet Protocol Version 4, Src: 172.16.0.4, Dst: 8.8.8.8
User Datagram Protocol, Src Port: 65516, Dst Port: 53
Domain Name System (query)

Figure 4.15: Wireshark capture of incoming packets in P-GW



No.	Time	Source	Destination	Protocol	Length	Info
2663	171.175149896	147.83.47.159	8.8.8.8	DNS	88	Standard query 0xfeaa AAAA s.gateway.messenger.live.com
2664	171.203154978	147.83.47.159	8.8.8.8	DNS	76	Standard query 0xeee0 AAAA dsn9.d.skype.net
2666	171.257772380	8.8.8.8	147.83.47.159	DNS	230	Standard query response 0xfeaa AAAA s.gateway.messenger.live.com CNAME skype.geo.msnmes…
2667	171.257836029	8.8.8.8	172.16.0.4	GTP <DNS>	266	Standard query response 0xfeaa AAAA s.gateway.messenger.live.com CNAME skype.geo.msnmes…
2668	171.280789902	8.8.8.8	147.83.47.159	DNS	178	Standard query response 0xeee0 AAAA dsn9.d.skype.net CNAME dsn9.skype-dsn.akadns.net SO…
2669	171.280810636	8.8.8.8	172.16.0.4	GTP <DNS>	214	Standard query response 0xeee0 AAAA dsn9.d.skype.net CNAME dsn9.skype-dsn.akadns.net SO…
2673	171.505156620	147.83.47.159	8.8.8.8	DNS	76	Standard query 0xe0f9 A mtalk.google.com
2675	171.514753106	8.8.8.8	147.83.47.159	DNS	121	Standard query response 0xe0f9 A mtalk.google.com CNAME mobile-gtalk.l.google.com A 74.…
2676	171.514776553	8.8.8.8	172.16.0.4	GTP <DNS>	157	Standard query response 0xe0f9 A mtalk.google.com CNAME mobile-gtalk.l.google.com A 74.…
2677	171.545114660	147.83.47.159	8.8.8.8	DNS	86	Standard query 0xc585 A android.clients.google.com
2678	171.565113699	147.83.47.159	8.8.8.8	DNS	90	Standard query 0x8f7f AAAA mobile.pipe.aria.microsoft.com
2679	171.577098312	8.8.8.8	147.83.47.159	DNS	126	Standard query response 0xc585 A android.clients.google.com CNAME android.l.google.com …

Frame 2667: 266 bytes on wire (2128 bits), 266 bytes captured (2128 bits) on interface 0
Ethernet II, Src: 3Com_99:98:dc (00:04:75:99:98:dc), Dst: All-HSRP-routers_64 (00:00:0c:07:ac:64)
Internet Protocol Version 4, Src: 147.83.47.159, Dst: 192.168.8.100
User Datagram Protocol, Src Port: 2152, Dst Port: 2152
GPRS Tunneling Protocol
Internet Protocol Version 4, Src: 8.8.8.8, Dst: 172.16.0.4
User Datagram Protocol, Src Port: 53, Dst Port: 65516
Domain Name System (response)

Figure 4.16: Wireshark capture of incoming packets in SGi

Figure 4.14 shows the packets that A-eNB sends to the S-GW, Figure 4.15 shows the packets that S-GW use gtp decapsulation to send to the P-GW, and Figure 4.16 shows the packets that are sent through SGi interface to the Internet. As we can see, there is a double GTP encapsulation and when the S-GW does the gtp decapsulation, it only decapsulates one of them, having GTP packets in the data when it is sent to the Internet. This process will be better explained by showing the user plane in Figure 4.17

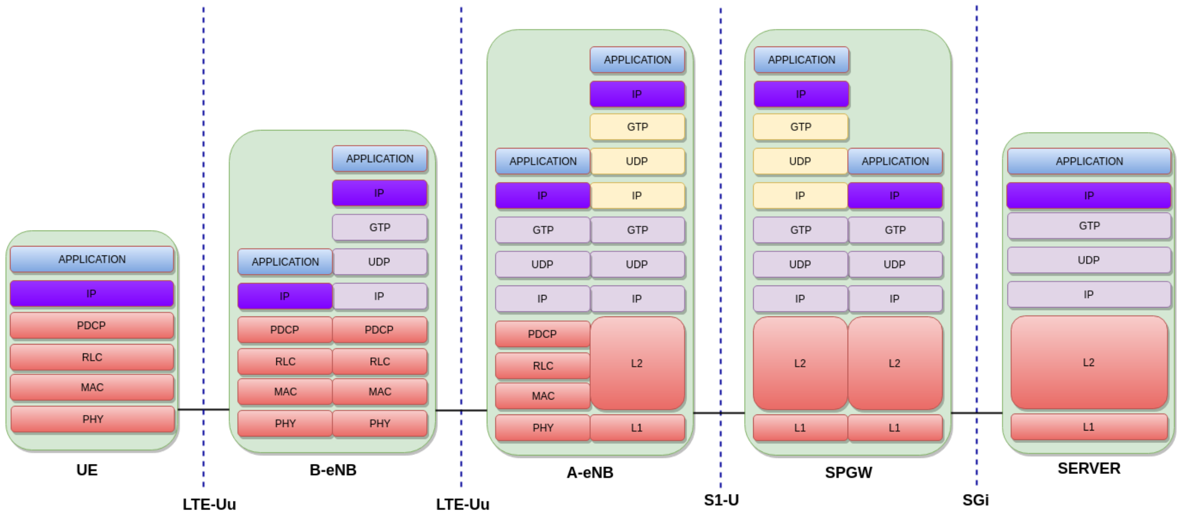


Figure 4.17: User Plane of LTE network with double GTP problem

First let's talk about possible solutions to mitigate the problem of double gtp encapsulation/decapsulation:

- **Update the A-eNB code**, to recognize which are packet from a normal UE and which are from the B-eNB, if the packets are from the B-eNB it will only relay them without any GTP encapsulation to the EPC, but if the packets are from a normal UE they will be relay with GTP encapsulation to the EPC.
- **Update SGW**, to be able to remove the second GTP encapsulation.
- **Use an SGW at B-eNB**, to remove the GTP encapsulation before sending the packets to the A-eNB.

Hence, we focused on the last option and implemented a solution based on this idea. To implement this solution it was not enough to put an SGW in B-eNB, because using OAI, SGW and PGW are merged together, and also the SPGW must be connected to one MME, so a virtual MME and a virtual SPGW were added to the B-eNB. In consequence, the final setup is shown in Figure 4.18.

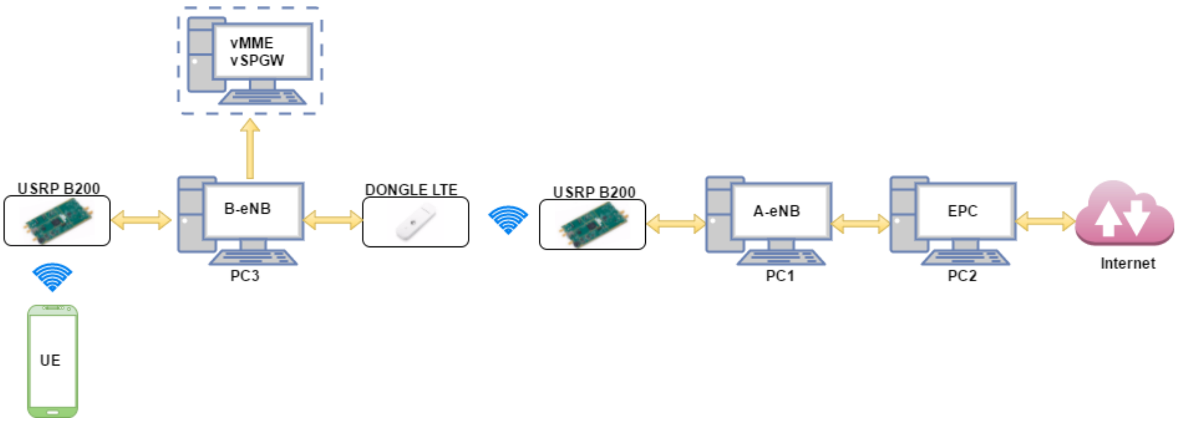


Figure 4.18: Self-Backhauling Final Setup with OAI project

In the detailed description of the self-backhauling LTE network, we are still using the traditional LTE network from Figure 4.12 for A-eNB and its EPC connection, but for the self-backhauling part of the network we are using the detailed description shown in Figure 4.19

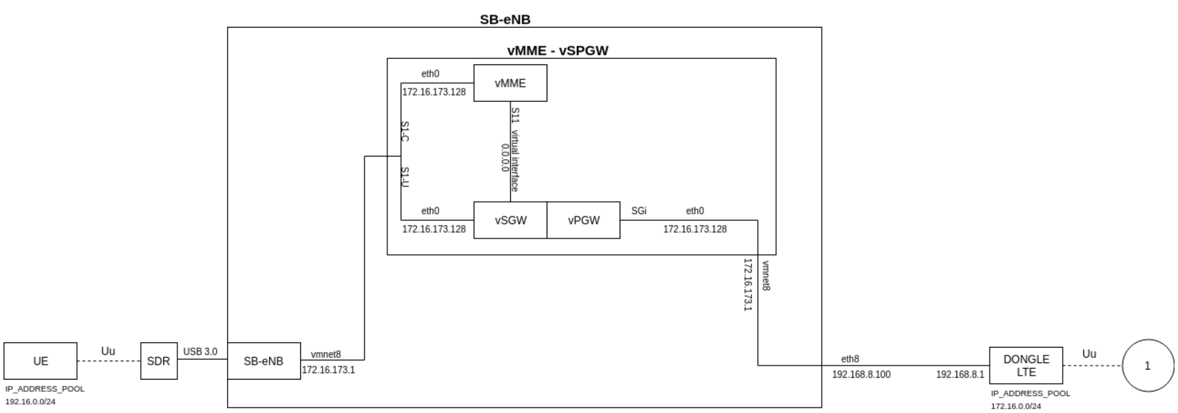


Figure 4.19: Detailed Self-Backhauling Network with vMME and vSPGW

In B-eNB as we are using a virtual machine to setup vMME and vSPGW, the B-eNB configuration file has to be changed as shown in Table 4.11 to direct the SPGW and MME connections to the new virtualized SPGW and MME accordingly.

mme_ip_address (ipv4)	"172.16.173.128";
ENB_INTERFACE_NAME_FOR_S1_MME	"vmnet8";
ENB_IPV4_ADDRESS_FOR_S1_MME	"172.16.173.1/24";
ENB_INTERFACE_NAME_FOR_S1_MME	"vmnet8";
ENB_INTERFACE_NAME_FOR_S1_MME	"172.16.173.1/24";
ENB_PORT_FOR_S1U	2152;

Table 4.11: B-eNB parameters to be changed for Self-Backhauling LTE network Final Setup

vMME

In the vMME, as the traditional MME, two files have to be configured: "mme.conf" and "mme_fd.conf";. These parameters that have to be changed are shown in Table 4.12 and Table 4.13 respectively.

MAXENB	2;
MAXUE	16;
GUMMEI_LIST	({MCC="208" ; MNC="92"; MME_GID="4" ; MME_CODE="1"; });
TAI_LIST	({MCC="208" ; MNC="93"; TAC = "1"; });
MME_INTERFACE_NAME_FOR_S1_MME	"eth0";
MME_IPV4_ADDRESS_FOR_S1_MME	"172.16.173.128/24";
MME_INTERFACE_NAME_FOR_S11_MME	"lo";
MME_IPV4_ADDRESS_FOR_S11_MME	"127.0.11.1/8";
MME_PORT_FOR_S11_MME	2123;
SGW_IPV4_ADDRESS_FOR_S11	"127.0.11.2/8";

Table 4.12: Parameters to be changed in "mme.conf" for vMME

Identity	“nano.openair4G.eur”;
Realm	“openair4G.eur”;
ConnectPeer	“hss.openair4G.eur” { ConnectTo = “192.168.12.63”; No_SCTP ; No_IPv6; Prefer_TCP; No_TLS; port = 3868; realm = “openair4G.eur”;};

Table 4.13: Parameters to be changed in “mme_fd.conf” for vMME

As we can see in Table 4.13, to connect to the HSS from the traditional network, the “ConnectPeer” parameter is pointing to the IP of the EPC.

vSPGW

As the traditional SPGW, the configuration file that has to be configured is “spgw.conf”. The parameters to be changed are shown in Table 4.14 and Table 4.15.

SGW_INTERFACE_NAME_FOR_S11	“lo”;
SGW_IPV4_ADDRESS_FOR_S11	“127.0.11.2/8”;
SGW_INTERFACE_NAME_FOR_S1U_S12_S4_UP	“eth0”;
SGW_IPV4_ADDRESS_FOR_S1U_S12_S4_UP	“172.16.173.128/24”;
SGW_IPV4_PORT_FOR_S1U_S12_S4_UP	2152;
SGW_INTERFACE_NAME_FOR_S5_S8_UP	“none”;
SGW_IPV4_ADDRESS_FOR_S5_S8_UP	“0.0.0.0/24”;

Table 4.14: Parameters of S-GW to be changed in “spgw.conf” for vSPGW

PGW_INTERFACE_NAME_FOR_S5_S8	“none”;
PGW_IPV4_ADDRESS_FOR_S5_S8	“0.0.0.0/24”;
PGW_INTERFACE_NAME_FOR_SGI	“etho”;
PGW_IPV4_ADDRESS_FOR_SGI	“172.16.173.128/24”;
PGW_MASQUERADE_SGI	“yes”;
UE_TCP_MSS_CLAMPING	“no”;
IPV4_LIST	(“192.16.0.0/24”);
DEFAULT_DNS_IPV4_ADDRESS	“8.8.8.8”;
DEFAULT_DNS_SEC_IPV4_ADDRESS	“8.8.4.4”;
UE_MTU	1400;

Table 4.15: Parameters of P-GW to be changed in “spgw.conf” for vSPGW

In this final setup, before sending the packets from the B-eNB to the A-eNB, they are sent to the vMME and vSPGW to do the GTP decapsulation, and when the data is sent to the A-eNB, it is seen as normal data from an UE, so when A-eNB does the GTP encapsulation, the final result will not have the double gtp encapsulation/decapsulation problem. This process will be better explained by showing the updated user plane in Figure 4.20.

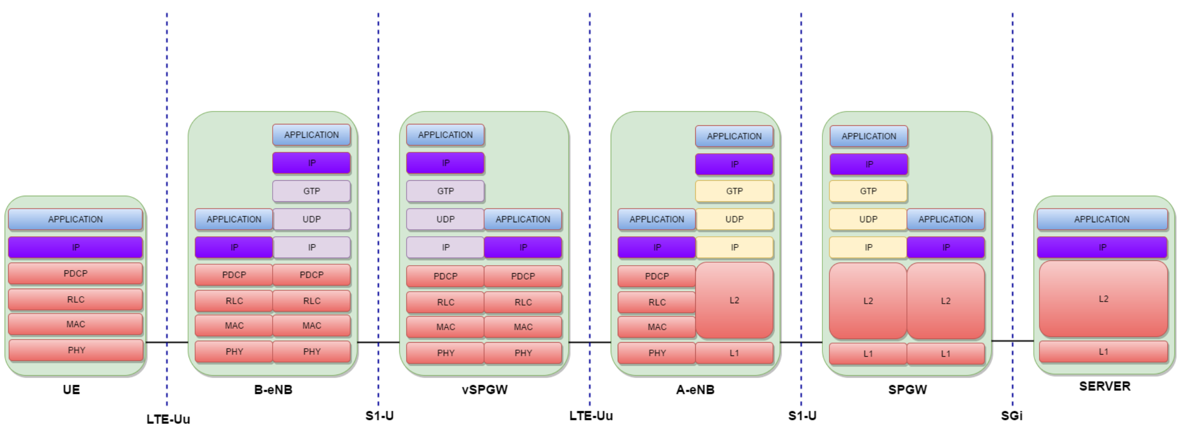


Figure 4.20: User Plane of the final Self-Backhauling LTE network setup

Chapter 5

Results

In order to obtain results for the implemented setup we used Band 7 which is composed of frequencies between 2620 and 2690 MHz. We compared the throughput observed by a UE connected to the conventional LTE network and the same UE connected through the implemented self-backhauling network. The throughput is measured in terms of uplink and downlink bitrate using iperf and speedtest Android application, respectively. The reason we used speedtest Android application for downlink bitrate is because in the self-backhauling network since the UE is connected to the vMME and vSPGW of the B-eNB and not to the MME and SPGW of the conventional LTE network, the vSPGW does the NAT of that UE and sends the traffic with the IP of the dongle LTE given by the EPC, so the EPC does not know the IP of the final UE, it only sees traffic coming from dongle LTE. For the iperf test we used TCP connection with the parameters shown in Table 5.1. Multiple tests (around 20) were made, since the difference between each result is insignificant the variance is not shown, only the average of the results are shown in this chapter.

Server	iperf3 -B 172.16.0.1 -s;
Client	iperf3 -c 172.16.0.1 -M 1400;

Table 5.1: Iperf parameters for server and client

Figure 5.1 shows the frequencies and bandwidth used for conventional LTE network and in-band self-backhauling network evaluations, where both A-eNB and B-eNB uses the same central frequencies. This represents the case, where the operator have limited spectrum or decides to do the set-up in this specific way. Figure 5.2 shows the frequencies and bandwidth used for out-band self-backhauling network evaluations.

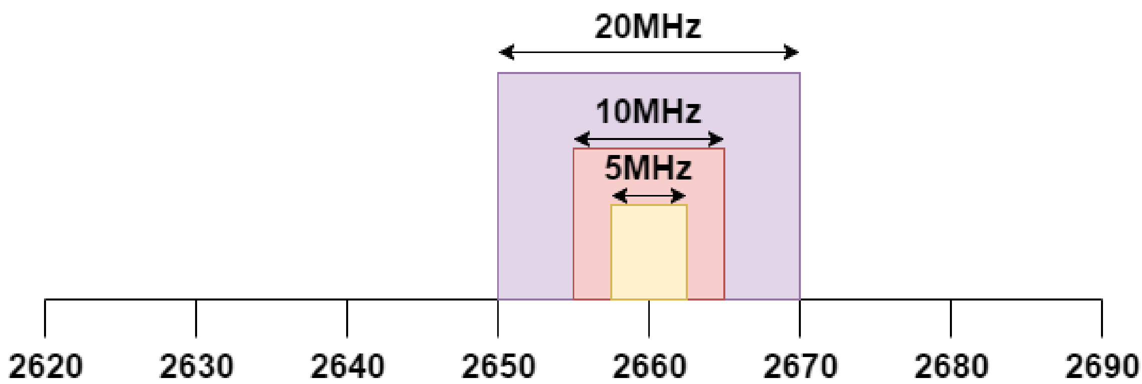


Figure 5.1: Spectrum used for conventional LTE network and In-Band Self-Backhauling tests

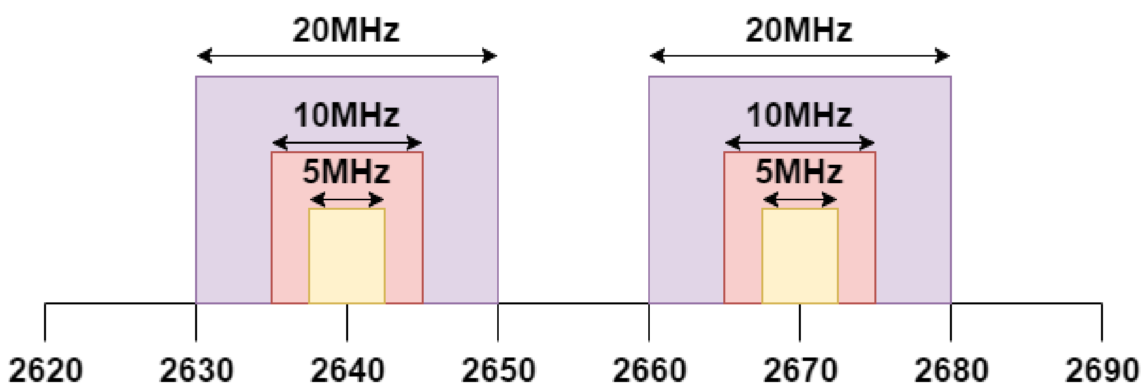


Figure 5.2: Spectrum used for Out-Band tests

Figures 5.3, 5.4 and 5.5 show the implemented setup in a real environment using the components explained in chapter 4.

Figure 5.3: Conventional LTE network implemented in a real environment

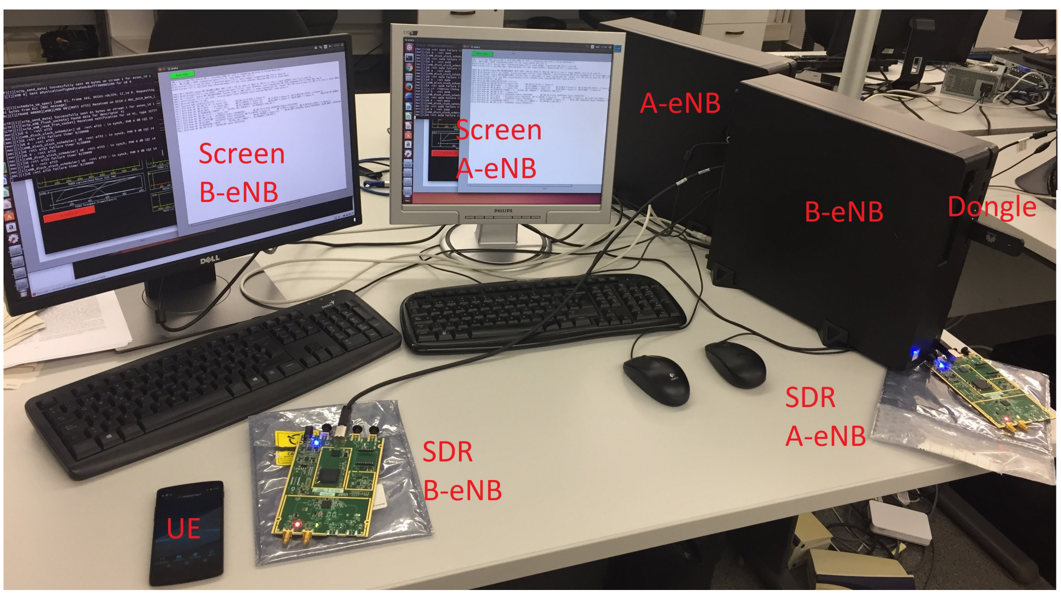


Figure 5.4: Self-Backhauling LTE network implemented in a real environment

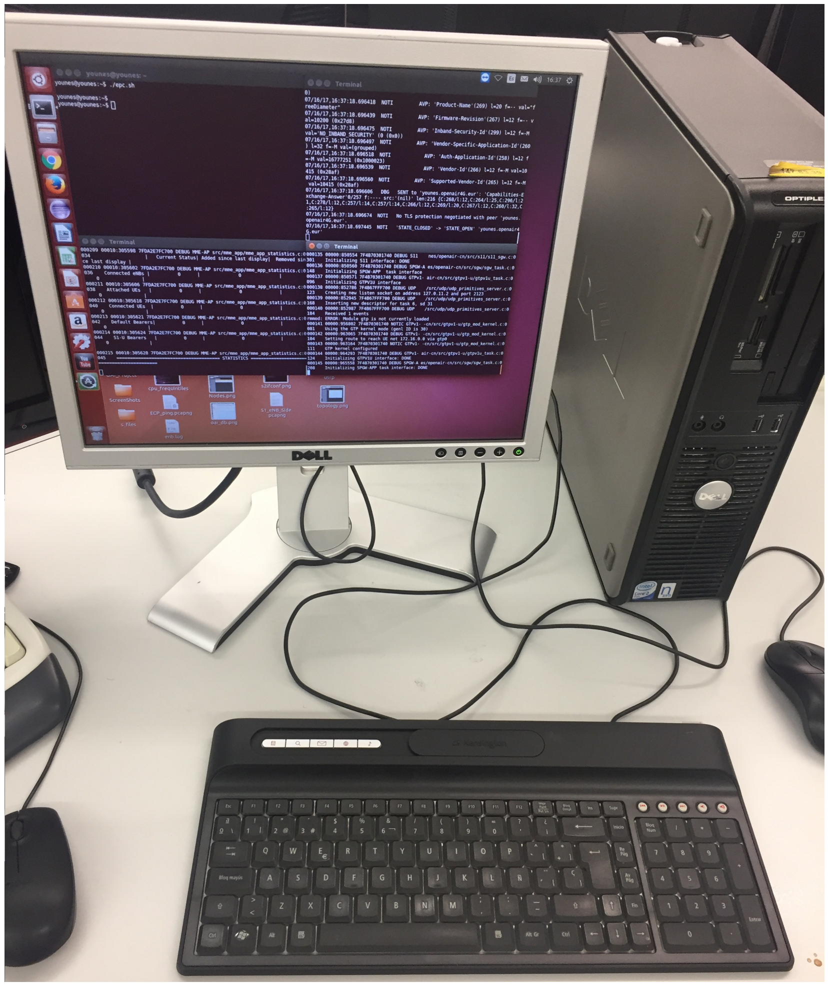


Figure 5.5: EPC implemented in a real environment

As we can see in Figure 5.4, there are two PCs, one for A-eNB and the other for B-eNB. In the B-eNB there is a Dongle LTE connected to the USB port, as explained before, this will allow B-eNB to work as a UE. There are two SDRs, we can see in the figure, one is for A-eNB and the other for B-eNB, and finally the UE that is close to the SDR of the B-eNB is the one used for the test in this project. The EPC is implemented in a separate PC shown in Figure 5.5.

5.1. Conventional LTE Network

We vary the "Tx" and "Rx" gain parameters of A-eNB to change the size of the cell, and evaluate the conventional LTE network setup under different gain values. By default, the Tx and Rx gain values provided by the OAI implementation are 90 and 125, respectively. Figure 5.6 illustrates the relative radio coverage observed for various gain value tuples. This coverage information was obtained by checking the position of the UE that would have the coverage of A-eNB.

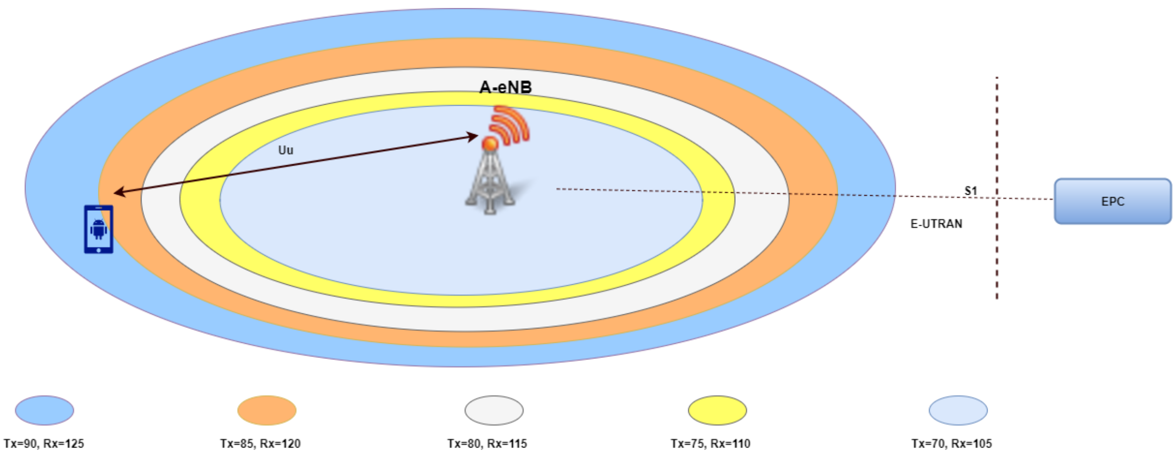


Figure 5.6: Radio coverage for conventional LTE network when changing Tx and Rx Gain parameters

For the rest of the evaluations, we fix the UE location and evaluate the throughput results of the network for UL and DL for varying gain value tuples, which are shown in Figures 5.7 and 5.8, respectively.

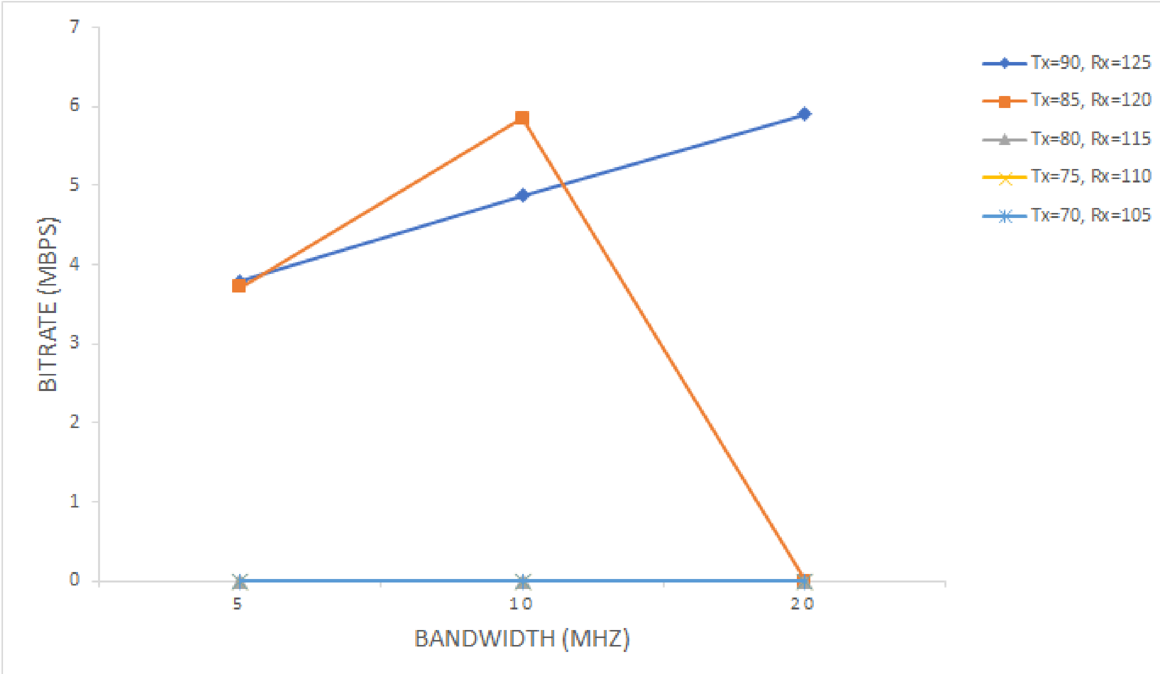


Figure 5.7: Results of Uplink with connection between UE and traditional LTE network

As we can see in Figure 5.7, the UE only has results when "Tx" gain is 90 and "Rx" gain is 125 using 5MHz, 10MHz and 20MHz of badwidth, and when "Tx" gain equal to 85 and "Rx" gain equal to 120 using 5MHz and 10MHz of bandwidth, however the rest of the results are zero, in which case the UE can not access the LTE network because it is out of the coverage as shown in Figure 5.6.

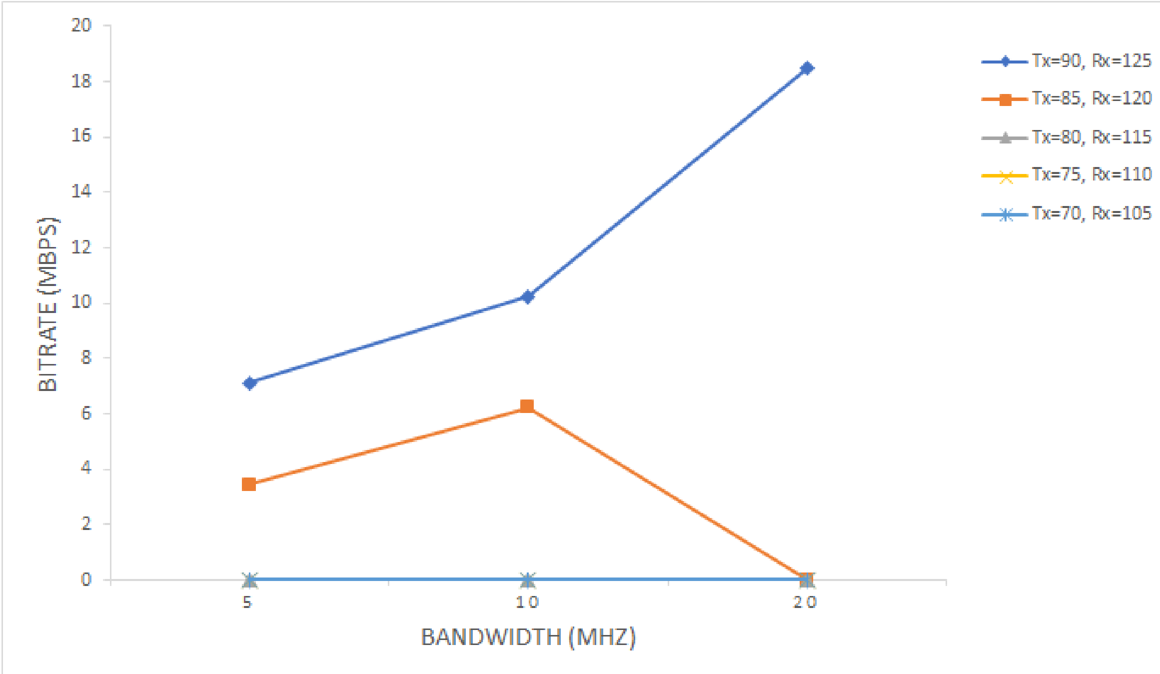


Figure 5.8: Results of Downlink with connection between UE and traditional LTE network

In Figure 5.8, as Figure 5.7, the UE only has results when “Tx” gain is 90 and “Rx” gain is 125, in the case of “Tx” gain equal to 85 and “Rx” gain equal to 120 the UE only has results when it uses 5MHz and 10MHz of bandwidth.

5.2. Self-Backhauling Network

Figure 5.9 illustrates how the changing of the “Tx” and “Rx” gain parameters affects the radio coverage of the self-backhauling LTE network setup. For all the tests done in self-backhauling network, in B-eNB “Tx” gain is 75 and “Rx” gain is 110.

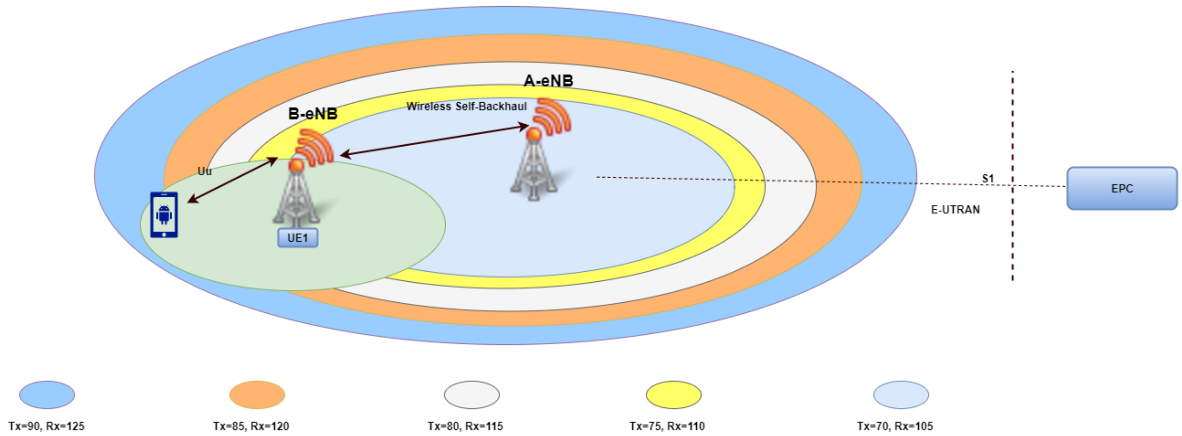


Figure 5.9: Radio coverage for self-backhauling LTE network when changing Tx and Rx Gain parameters

5.2.1. Self-Backhauling In-Band

Figures 5.10 and 5.11 show the throughput results of the self-backhauling in-band network setup.

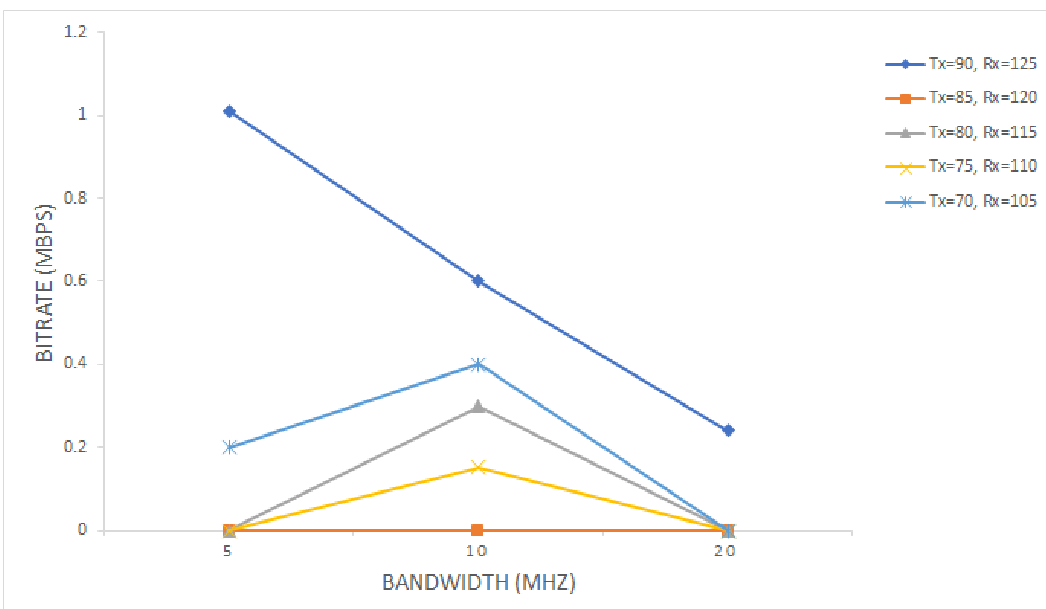


Figure 5.10: Results of Uplink with connection between UE and Self-Backhauling LTE In-Band network

As we can see in Figure 5.10, the UL throughput is very low, this happens because of the self-interference that exists when in-band solution is used.

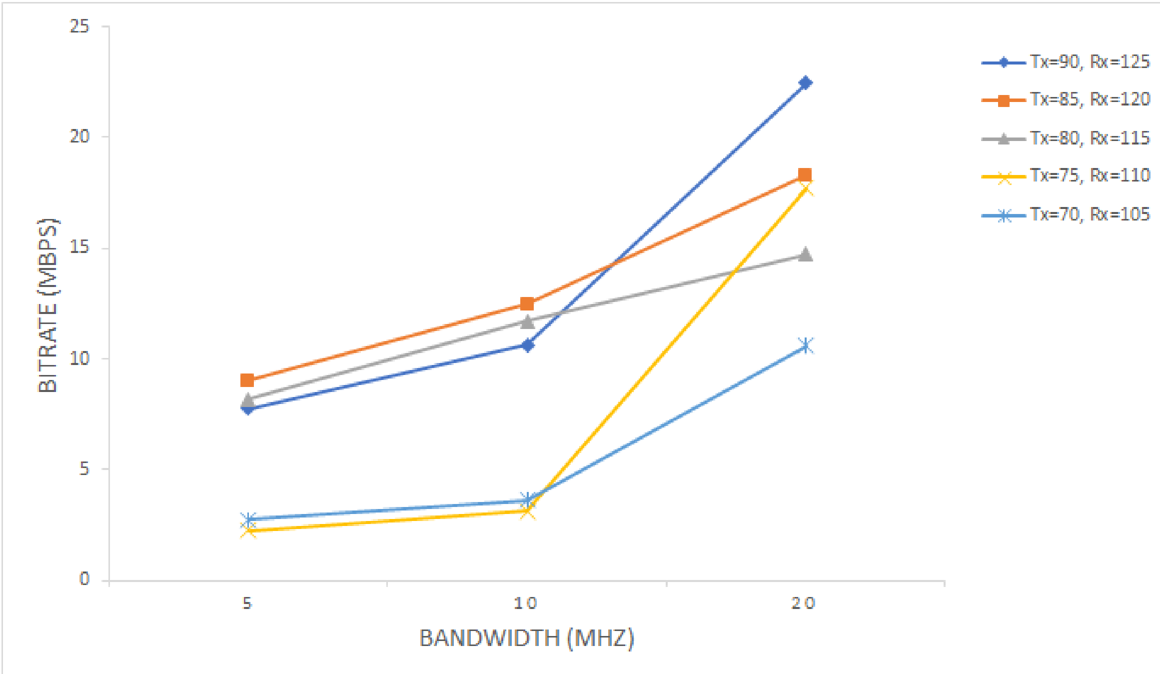


Figure 5.11: Results of Downlink with connection between UE and Self-Backhauling LTE In-Band network

In contrast to Figure 5.8, in Figure 5.11 we obtain results for all the tested values of "Tx" and "Rx" gain. Hence, with a self-backhauling solution we can extend the coverage of the conventional LTE network, even if an in-band solution is used.

Although there is self-interference when in-band solution is used, we obtained better results in downlink bitrate than in uplink bitrate, the reason could be the type of modulation they use, since downlink uses Orthogonal Frequency Division Multiple-Access (OFDMA) and UL uses Single-Carrier Frequency-Division Multiple-Access (SC-FDMA). OFDMA uses multiple subcarriers and this could be reducing the self-interference in contrast to SC-FDMA which uses a single carrier.

5.2.2. Self-Backhauling Out-Band

Figures 5.12 and 5.13 shown the throughput results of the self-backhauling in-band network setup for UL and DL, respectively.

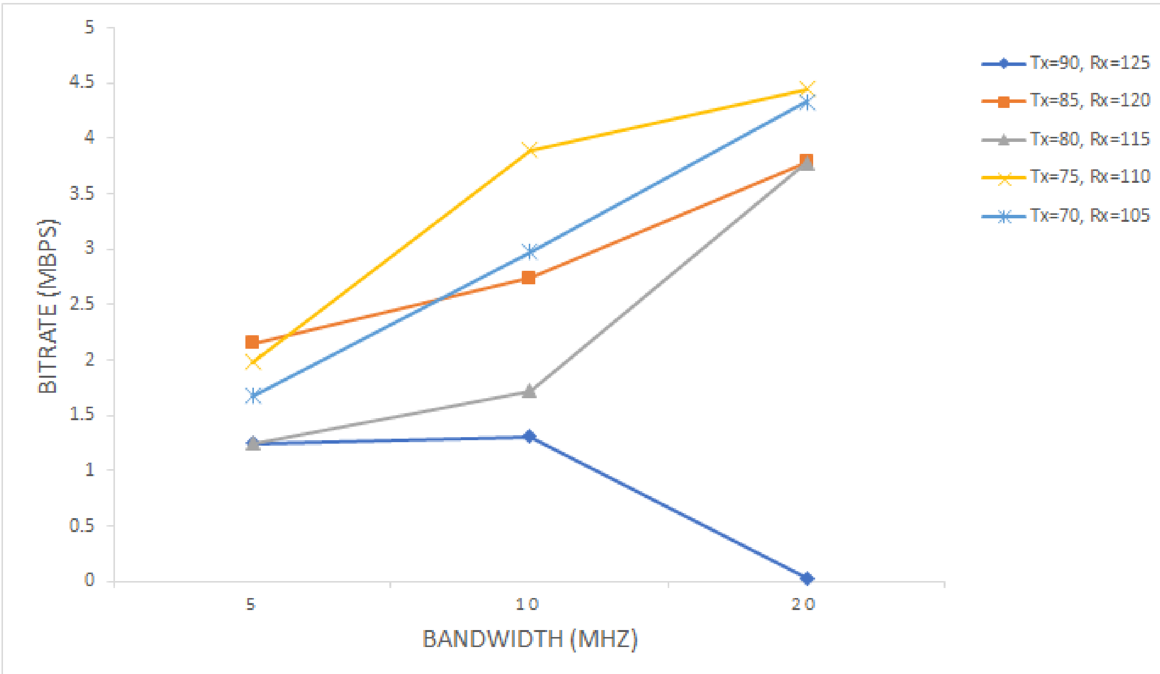


Figure 5.12: Results of Uplink with connection between UE and Self-Backhauling LTE Out-Band network

In contrast to in-band solution (see Figure 5.10), where the access and the backhaul link share the same frequency, in the out-band solution there is no self-interference, so the throughput of the uplink is better as we can see in Figure 5.12.

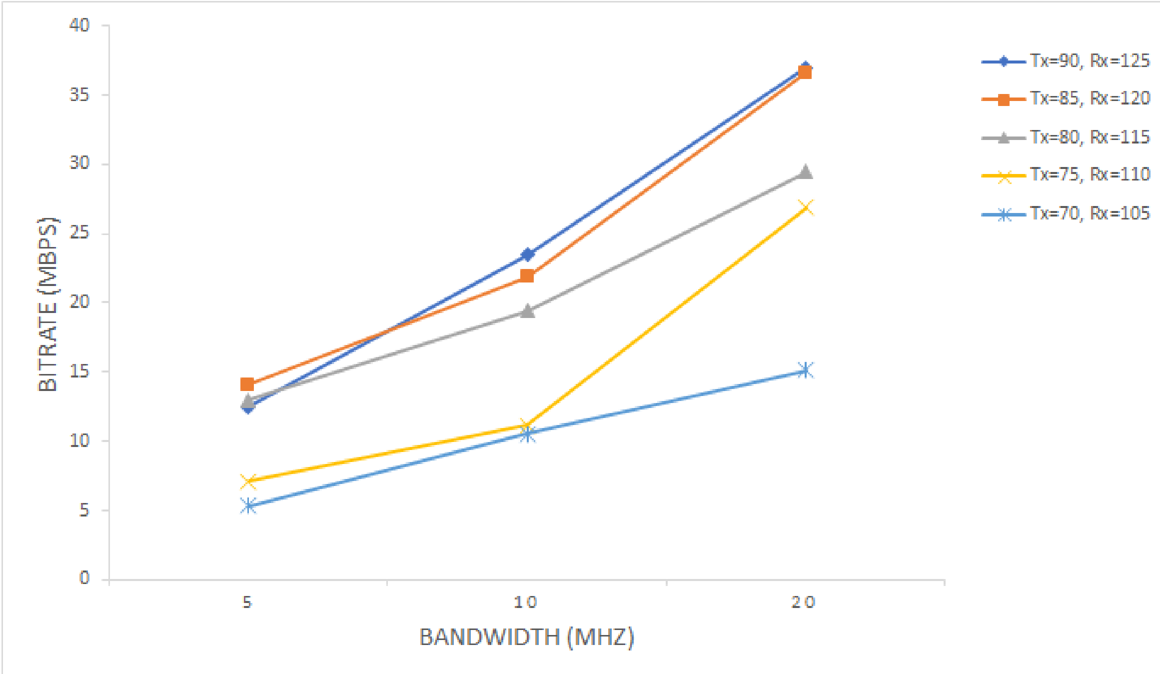


Figure 5.13: Results of Downlink with connection between UE and Self-Backhauling LTE Out-Band network

Although we obtained an enhancement of the conventional LTE network in the downlink when in-band solution was used, using out-band the results are even better because (see Figure 5.13), as explained before, in this setup there is no self-interference. As we have shown, using a self-backhauling eNB help us to increase the radio coverage of the conventional LTE network.

5.2.3. End-to-end Delay and Packet Delivery Ratio (PDR) Results

Table 5.2 shows the range of ping results when using conventional LTE network using both Nexus 5 and Dongle LTE Huawei E3772 as UEs. Table 5.3 shows the range of ping results when using self-backhauling LTE network using Nexus 5 as a UE. In this test, ping sends data with small packet sizes, and six tests were made using in-band and out-band solution, the first two when "Tx" gain is 90 and "Rx" gain is 125, the next two when "Tx" gain is 85 and "Rx" gain is 120, and the last two when "Tx" gain is 80 and "Rx" gain is 115.

UEs	**Delay**	**PDR**
Nexus 5	15ms - 25ms	100%
PC (DONGLE LTE Huawei E3772)	30ms - 50ms	100%

Table 5.2: Range of ping results in conventional LTE network

UEs	**Delay**	**PDR**
Nexus 5	40ms - 65ms	100%

Table 5.3: Range of ping results in Self-Backhauling LTE network

5.2.4. Packet Captures Analyzing the Self-Backhauling Setup

In the following figures, wireshark captures of how the self-backhauling network works are shown. In these pictures it will be shown when a UE is surfing through the Internet.

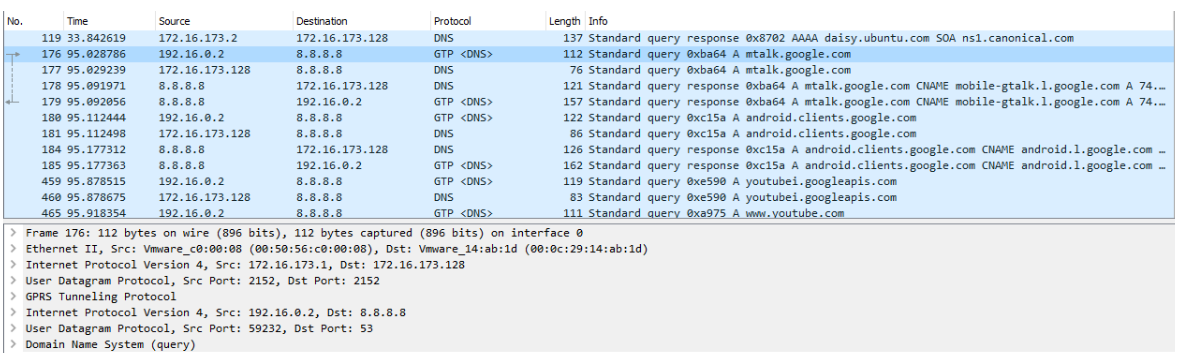
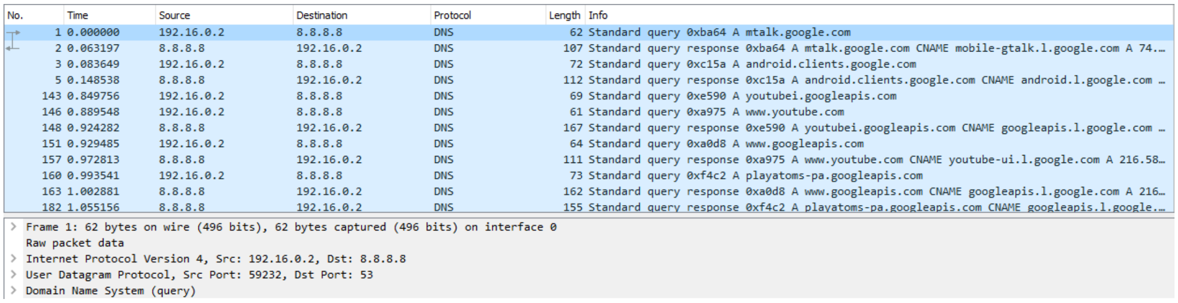


Figure 5.14: Wireshark capture of the packets in vSGW

Figure 5.14 shows the wireshark captures of the vSGW when the B-eNB sends the data received from UE after the GTP encapsulation. As we can see the UE is sending traffic through DNS protocol in 192.16.0.2 (IP of the UE) -> 8.8.8.8 (public DNS of google) which is encapsulated by the B-eNB using GTP encapsulation.

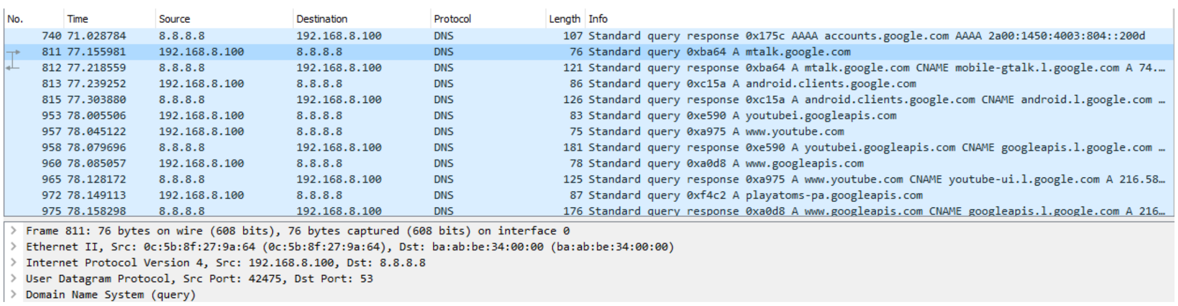


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.16.0.2	8.8.8.8	DNS	62	Standard query 0xba64 A mtalk.google.com
2	0.063197	8.8.8.8	192.16.0.2	DNS	107	Standard query response 0xba64 A mtalk.google.com CNAME mobile-gtalk.l.google.com A 74.…
3	0.083649	192.16.0.2	8.8.8.8	DNS	72	Standard query 0xc15a A android.clients.google.com
5	0.148538	8.8.8.8	192.16.0.2	DNS	112	Standard query response 0xc15a A android.clients.google.com CNAME android.l.google.com …
143	0.849756	192.16.0.2	8.8.8.8	DNS	69	Standard query 0xe590 A youtubei.googleapis.com
146	0.889548	192.16.0.2	8.8.8.8	DNS	61	Standard query 0xa975 A www.youtube.com
148	0.924282	8.8.8.8	192.16.0.2	DNS	167	Standard query response 0xe590 A youtubei.googleapis.com CNAME googleapis.l.google.com …
151	0.929485	192.16.0.2	8.8.8.8	DNS	64	Standard query 0xa0d8 A www.googleapis.com
157	0.972813	8.8.8.8	192.16.0.2	DNS	111	Standard query response 0xa975 A www.youtube.com CNAME youtube-ui.l.google.com A 216.58…
160	0.993541	192.16.0.2	8.8.8.8	DNS	73	Standard query 0xf4c2 A playatoms-pa.googleapis.com
163	1.002881	8.8.8.8	192.16.0.2	DNS	162	Standard query response 0xa0d8 A www.googleapis.com CNAME googleapis.l.google.com A 216…
182	1.055156	8.8.8.8	192.16.0.2	DNS	155	Standard query response 0xf4c2 A playatoms-pa.googleapis.com CNAME googleapis.l.google.…

Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0
Raw packet data
Internet Protocol Version 4, Src: 192.16.0.2, Dst: 8.8.8.8
User Datagram Protocol, Src Port: 59232, Dst Port: 53
Domain Name System (query)

Figure 5.15: Wireshark capture of the packets in vPGW

Figure 5.15 shows the wireshark captures of the vPGW when the vSGW sends the data received from B-eNB after the GTP decapsulation. As seen in the capture, here, the packet is decapsulated and the resulting IP packet source is B-eNB and the destination is the DNS server.



No.	Time	Source	Destination	Protocol	Length	Info
740	71.028784	8.8.8.8	192.168.8.100	DNS	107	Standard query response 0x175c AAAA accounts.google.com AAAA 2a00:1450:4003:804::200d
811	77.155981	192.168.8.100	8.8.8.8	DNS	76	Standard query 0xba64 A mtalk.google.com
812	77.218559	8.8.8.8	192.168.8.100	DNS	121	Standard query response 0xba64 A mtalk.google.com CNAME mobile-gtalk.l.google.com A 74.…
813	77.239252	192.168.8.100	8.8.8.8	DNS	86	Standard query 0xc15a A android.clients.google.com
815	77.303880	8.8.8.8	192.168.8.100	DNS	126	Standard query response 0xc15a A android.clients.google.com CNAME android.l.google.com …
953	78.005506	192.168.8.100	8.8.8.8	DNS	83	Standard query 0xe590 A youtubei.googleapis.com
957	78.045122	192.168.8.100	8.8.8.8	DNS	75	Standard query 0xa975 A www.youtube.com
958	78.079696	8.8.8.8	192.168.8.100	DNS	181	Standard query response 0xe590 A youtubei.googleapis.com CNAME googleapis.l.google.com …
960	78.085057	192.168.8.100	8.8.8.8	DNS	78	Standard query 0xa0d8 A www.googleapis.com
965	78.128172	8.8.8.8	192.168.8.100	DNS	125	Standard query response 0xa975 A www.youtube.com CNAME youtube-ui.l.google.com A 216.58…
972	78.149113	192.168.8.100	8.8.8.8	DNS	87	Standard query 0xf4c2 A playatoms-pa.googleapis.com
975	78.158298	8.8.8.8	192.168.8.100	DNS	176	Standard query response 0xa0d8 A www.googleapis.com CNAME googleapis.l.google.com A 216…

Frame 811: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0
Ethernet II, Src: 0c:5b:8f:27:9a:64 (0c:5b:8f:27:9a:64), Dst: ba:ab:be:34:00:00 (ba:ab:be:34:00:00)
Internet Protocol Version 4, Src: 192.168.8.100, Dst: 8.8.8.8
User Datagram Protocol, Src Port: 42475, Dst Port: 53
Domain Name System (query)

Figure 5.16: Wireshark capture of the packets in B-eNB Dongle before sending to A-eNB

Figure 5.16 shows the wireshark captures of the B-eNB Dongle interface when it sends the data as a an LTE UE attached to the A-eNB. Here, the IP source address is changed by the dongle to 192.168.8.100, i.e., to the IP of interface assigned dongle at B-eNB PC.

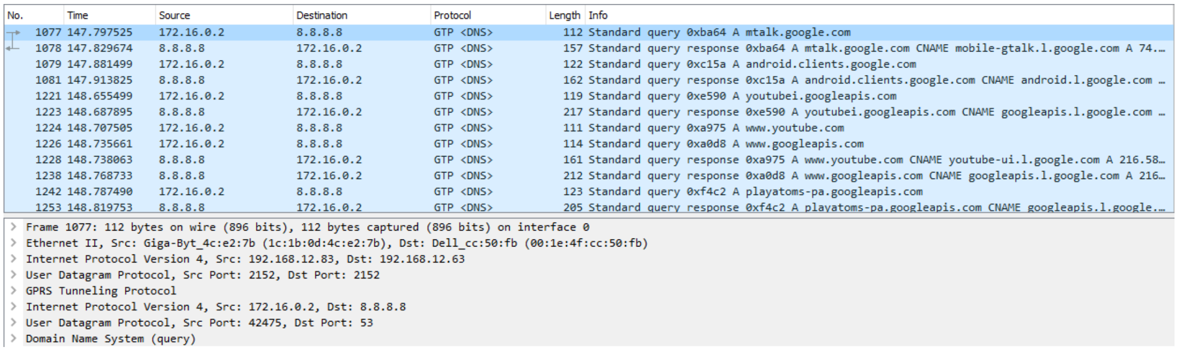


Figure 5.17: Wireshark capture of the packets in A-eNB

Figure 5.17 shows the wireshark captures at A-eNB when it sends the data received from B-eNB after the GTP encapsulation is done. Here we can see that the IP is changed since the EPC gives to the dongle the IP 172.16.0.2, and the data is coming from the B-eNB.

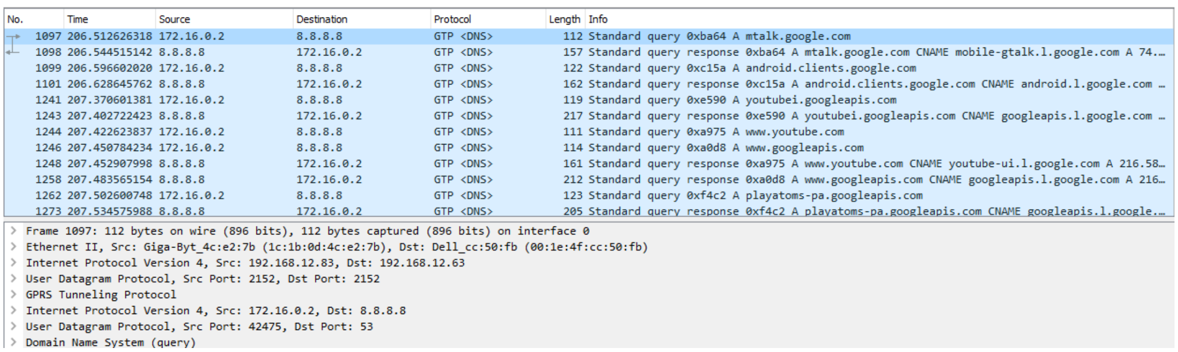
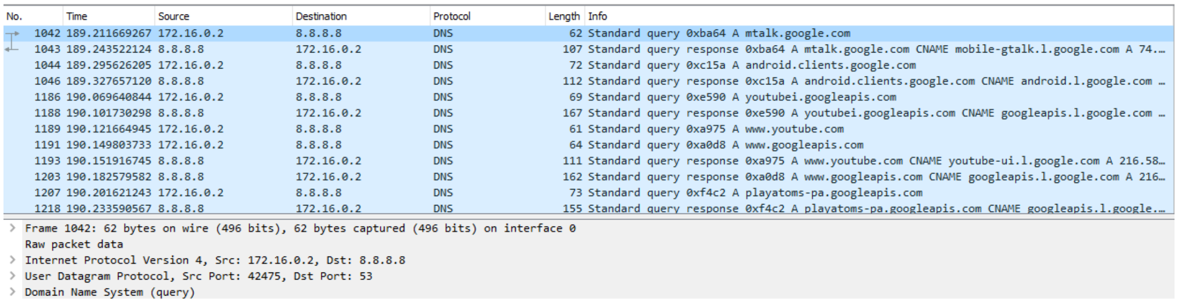


Figure 5.18: Wireshark capture of the packets in SGW

Figure 5.18 shows the wireshark captures of the SGW when the A-eNB sends the data after its GTP encapsulation. As we can see, the headers are the same with the previous capture, since in Figure 5.17 the packets that the A-eNB sends to the SGW after GTP encapsulation are shown, and here the packets received by SGW machine (i.e. the machine hosting EPC) are shown.

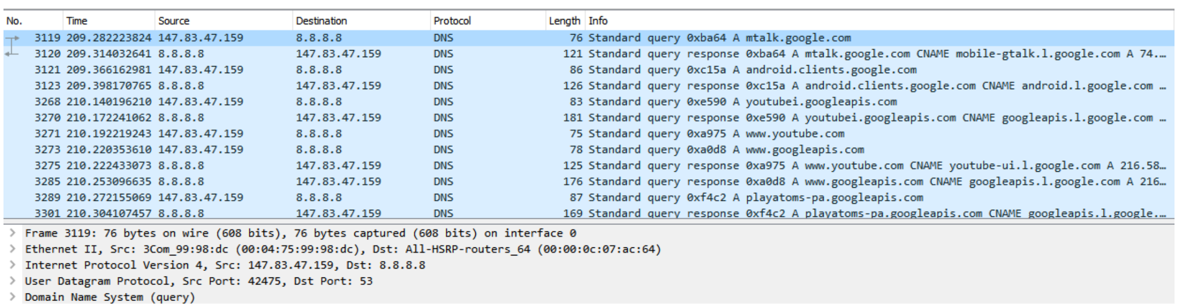


No.	Time	Source	Destination	Protocol	Length	Info
1042	189.211669267	172.16.0.2	8.8.8.8	DNS	62	Standard query 0xba64 A mtalk.google.com
1043	189.243522124	8.8.8.8	172.16.0.2	DNS	107	Standard query response 0xba64 A mtalk.google.com CNAME mobile-gtalk.l.google.com A 74.…
1044	189.295626205	172.16.0.2	8.8.8.8	DNS	72	Standard query 0xc15a A android.clients.google.com
1046	189.327657120	8.8.8.8	172.16.0.2	DNS	112	Standard query response 0xc15a A android.clients.google.com CNAME android.l.google.com …
1186	190.069640844	172.16.0.2	8.8.8.8	DNS	69	Standard query 0xe590 A youtubei.googleapis.com
1188	190.101730298	8.8.8.8	172.16.0.2	DNS	167	Standard query response 0xe590 A youtubei.googleapis.com CNAME googleapis.l.google.com …
1189	190.121664945	172.16.0.2	8.8.8.8	DNS	61	Standard query 0xa975 A www.youtube.com
1191	190.149803733	172.16.0.2	8.8.8.8	DNS	64	Standard query 0xa0d8 A www.googleapis.com
1193	190.151916745	8.8.8.8	172.16.0.2	DNS	111	Standard query response 0xa975 A www.youtube.com CNAME youtube-ui.l.google.com A 216.58…
1203	190.182579582	8.8.8.8	172.16.0.2	DNS	162	Standard query response 0xa0d8 A www.googleapis.com CNAME googleapis.l.google.com A 216…
1207	190.201621243	172.16.0.2	8.8.8.8	DNS	73	Standard query 0xf4c2 A playatoms-pa.googleapis.com
1218	190.233590567	8.8.8.8	172.16.0.2	DNS	155	Standard query response 0xf4c2 A playatoms-pa.googleapis.com CNAME googleapis.l.google.…

Frame 1042: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0
Raw packet data
Internet Protocol Version 4, Src: 172.16.0.2, Dst: 8.8.8.8
User Datagram Protocol, Src Port: 42475, Dst Port: 53
Domain Name System (query)

Figure 5.19: Wireshark capture of the packets in PGW

Figure 5.19 shows the wireshark captures of the PGW when the SGW sends the data received from A-eNB after the GTP decapsulation. As can be seen, there is no tunnel created for S5 interface (SGW-PGW interface) in OAI, possibly because they run in a single (spgw) executable. After the GTP encapsulation, it can be seen that, the IP source of the packet is now the LTE dongle (i.e., B-eNB dongle) and the destination is the DNS server.



No.	Time	Source	Destination	Protocol	Length	Info
3119	209.282223824	147.83.47.159	8.8.8.8	DNS	76	Standard query 0xba64 A mtalk.google.com
3120	209.314032641	8.8.8.8	147.83.47.159	DNS	121	Standard query response 0xba64 A mtalk.google.com CNAME mobile-gtalk.l.google.com A 74.…
3121	209.366162981	147.83.47.159	8.8.8.8	DNS	86	Standard query 0xc15a A android.clients.google.com
3123	209.398170765	8.8.8.8	147.83.47.159	DNS	126	Standard query response 0xc15a A android.clients.google.com CNAME android.l.google.com …
3268	210.140196210	147.83.47.159	8.8.8.8	DNS	83	Standard query 0xe590 A youtubei.googleapis.com
3270	210.172241062	8.8.8.8	147.83.47.159	DNS	181	Standard query response 0xe590 A youtubei.googleapis.com CNAME googleapis.l.google.com …
3271	210.192219243	147.83.47.159	8.8.8.8	DNS	75	Standard query 0xa975 A www.youtube.com
3273	210.220353610	147.83.47.159	8.8.8.8	DNS	78	Standard query 0xa0d8 A www.googleapis.com
3275	210.222433073	8.8.8.8	147.83.47.159	DNS	125	Standard query response 0xa975 A www.youtube.com CNAME youtube-ui.l.google.com A 216.58…
3285	210.253096635	8.8.8.8	147.83.47.159	DNS	176	Standard query response 0xa0d8 A www.googleapis.com CNAME googleapis.l.google.com A 216…
3289	210.272155069	147.83.47.159	8.8.8.8	DNS	87	Standard query 0xf4c2 A playatoms-pa.googleapis.com
3301	210.304107457	8.8.8.8	147.83.47.159	DNS	169	Standard query response 0xf4c2 A playatoms-pa.googleapis.com CNAME googleapis.l.google.…

Frame 3119: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0
Ethernet II, Src: 3Com_99:98:dc (00:04:75:99:98:dc), Dst: All-HSRP-routers_64 (00:00:0c:07:ac:64)
Internet Protocol Version 4, Src: 147.83.47.159, Dst: 8.8.8.8
User Datagram Protocol, Src Port: 42475, Dst Port: 53
Domain Name System (query)

Figure 5.20: Wireshark capture of the packets in SGi

Figure 5.20 shows the wireshark captures of the SGi interface when the PGW sends the data received from SGW to send it to the Internet. As seen in the figure, now a global IP replaces the source address, since this is the "public" IP used by the PGW for SGi. This packet will go to DNS server and the reply will be targeted to this same SGi interface that will be forwarded to the UE.

As we saw in the previous figures, it is shown through physical experiments that the self-backhauling LTE network solution developed in this project works successfully, and the problem of the double GTP encapsulation was solved.

Chapter 6

Conclusions and Future Work

In this thesis, we propose and develop an LTE self-backhauling solution built on an open-source LTE and EPC project (by OpenAirInterface project) and using commodity hardware.

First, a conventional LTE network is implemented, which covers all protocol stack of 3GPP in E-UTRAN and EPC. This is used as a base to setup the self-backhauling network, and also to compare with the final results. For the self-backhauling network, two solutions were presented and evaluated: the first was in-band self-backhauling network, where the self-backhauled eNB (B-eNB) shares the same central frequency and bandwidth as the backhauling eNB (A-eNB) and the second was out-band self-backhauling network, where they are assigned non-overlapping frequency spectrum.

Through physical evaluations and using an off-the-shelf UE, it was shown that a self-backhauling network allow us to extend the radio coverage of the conventional LTE network. Besides using out-band solution is better in performance compared to in-band solution, since in out-band solution the self-interference does not exist, because in contrast to in-band solution, it uses different frequencies for B-eNB and A-eNB.

For future work, in terms of better performance, the code of the A-eNB can be changed in order to distinguish the incoming packets from a normal UE and a B-eNB, for intelligent resource scheduling olution. Moreover, this way the packets coming from B-eNB might not be encapsulated through GTP (assuming static connections). With this solution, it will not be necessary to setup a virtual MME and SPGW as shown in this project, since the A-eNB will solve the problem of the double GTP encapsulation shown. Another solution would be that in the EPC, the SGW code can be updated to be able to remove the double GTP encapsulation.

References

[1] A. Apostolaras et al., "Evolved user equipment for collaborative wireless backhauling in next generation cellular networks" in IEEE SECON, 2015.

[2] A. Lo and I. Niemegeers, "Multi-hop Relay Architecture for 3GPP LTE-Advanced," Proceedings of the 2009 IEEE 9th Malaysia International Conference on Communications, December 2009, pp. 123-127.

[3] A. Mourad, "Self-Backhauling in 5G", Interdigital, 2015.

[4] C. Hoymann, A. Racz, N. Johansson, and Johan Lundsjo, "A Self backhauling Solution for LTE-Advanced", WWRF21-WG4-07, 2008.

[5] C. Hoymman, "LTE-Advanced: Self-Backhauling for cost reduction", Ericsson Research.

[6] EURECOM. [Online]. Available: www.eurecom.fr/en

[7] E. Thirumalai Nambi, M. Rajeev Kumar, "LTE Techniques in 4G Wireless System Using Protocol Stack", Springer, 2011.

[8] E-UTRAN User Guide. [Online]. Available: http://manualzz.com/doc/4142140/eutran-user-guide

[9] F. Khan, Z. Pi, "An introduction to millimeter wave mobile broadband systems", IEEE Commun. Mag., vol. 49, no. 6, pp. 101-107, Jun. 2011.

[10] Git Repository of Open Air Interface Project. [Online]. Available: https://gitlab.eurecom.fr/oai

[11] Long Term Evolution (LTE) Overview. [Online]. Available: http://www.3gpp.org/technologies/keywords-acronyms/98-lte

[12] "Microwave backhaul for LTE and beyond", Nokia, 2016.

[13] N. Nikaein et al., "Openairinterface 4g: an open lte network in a pc,"International Conference on Mobile Computing and Networking, Demo paper, 2014.

[14] OAI Hardware Requirements. [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirSystemRequirements

[15] OpenAirInterface. [Online]. Available: www.openairinterface.org

[16] OpenAirInterface System Emulation (OAISIM). Available: https://gitlab.eurecom.fr/oai/openairinterf

[17] Panasonic, 'R1-082397: Discussion on the various types of relays', www.3gpp.org, 3GPP TSG RAN WG1, meeting 54, Warsaw, Poland, June 2008.

[18] P. Lescuyer, T. Lucidarme, "Evolved Packet System (EPS): The LTE and SAE Evolution of 3G UMTS". John Wiley & Sons, Ltd, 2008.

[19] R. Favraud, N. Nikaein, "Wireless Mesh Backhauling for LTE/LTE-A Networks", MILCOM 2015, pp. 695-700, 2015-Oct.

[20] Specifications Sheet of USRP B200/B210. [Online]. Available: https://www.ettus.com/content/files/b200-b210_spec_sheet.pdf

[21] S. Rangan, T. S. Rappaport, E. Erkip, "Millimeter wave cellular networks: Potentials and challenges", Proc. IEEE, vol. 102, no. 3, pp. 366-385, Mar. 2014.

[22] "SVN repository of the OAI prototype implementation of the proposed architecture." [Online]. Available: https://svn.eurecom.fr/openairsvn/ openair4G/branches/lolamesh

[23] T. Ali-Yahiya, "Understanding LTE and its Performance", Springer, 2011.

[24] The Evolved Packet Core. [Online]. Available: http://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core

[25] “The LTE Network Architecture”, White Paper, Alcatel Lucent, 2009.

[26] The openairinterface platform. [Online]. Available: www.openairinterface.org

[27] The Statistics Portal. Number of mobile phone users worldwide from 2013 to 2019 (in billions). Available: https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/

[28] T.S Rappaport et al., "Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!", IEEE Access, vol. 1, pp. 335-349, 2013.

[29] Tutorials (OpenAirInterface Usage). [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirUsage

[30] V. Raghavan, J. Cezanne, S. Subramanian, A. Sampath, O. Koymen, "Beamforming tradeoffs for initial UE discovery in millimeter-wave MIMO systems", IEEE J. Sel. Topics Signal Process., vol. 10, no. 3, pp. 543-559, Apr. 2016.

[31] Y. Chen, X. Lagrange, "Architecture and Protocols of EPC-LTE with relay", TELECOM BRETAGNE 2013.

[32] 3GPP. HetNet/Small Cells. Available: http://www.3gpp.org/hetnet

[33] 3GPP. LTE Advanced. Release 10. Available: http://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced

[34] 3GPP. TR 36.806 Evolved Universal Terrestrial Radio Access (E-UTRA); Relay architectures for E-UTRA (LTE-Advanced). Available: http://www.qtc.jp/3GPP/Specs/36806-900.pdf

[35] 3G TS 29.060 v3.5.0. “GPRS Tunnelling Protocol Across Gn and Gp interface”, Nokia, 2016.