Investigating the Value and Capabilities of Over the Air Testbeds Through Implementation of an LTE Network and Simple Coexistence Strategy

Prepared by:

Matthew William Lock
LCKMAT002

Department of Electrical Engineering
University of Cape Town

Prepared for:

Dr. Joyce Mwangama

Department of Electrical Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Bachelor of Science degree in Electrical and Computer Engineering.

November 11, 2020

Keywords— LTE, Over-The-Air Testbed, Coexistence, srsLTE, OpenAirInterface

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Signature:  .....

M. W. Lock

Date: 11/11/2020.....

Terms of Reference

Supervisor	Dr Joyce Mwangama
Title	Over the Air Implementation of a Mobile Network
Description	Creating over-the-air testbeds and prototypes of wireless systems have become increasingly more pervasive in the last decade thanks to advances in Software Defined Radio technology. But hardware alone is not the reason. Open source and commercial software have played a large role in making over-the-air testbeds possible. This project implementation of a cellular stack integrated onto an SDR to emulate a mobile base station (NodeB, eNodeB or gNodeB). The aim of this project is to design and implement a software-defined radio access network platform that utilises the OpenAirInteface toolkit with the integration of a controller to emulate the mobile radio network over unoccupied frequencies. The emulation platform should be used to test, evaluate and validate a real mobile network system performance over actual physical devices.
Deliverables	An implementation of a base station whose mobile network control is emulated and transport is achieved over the air. This should also be interworked to show how a real user device can communicate over this network.
Skills/Requirements	Proficient programmer in Python, C, C++ or Java languages. Comfortable with Linuxbased systems, wireless network protocols and network measurements.
GA 4: Investigations, experiments and analysis.	1. An investigation into the tools required to build the over the air mobile network. 2. Once the network is deployed, experimentation into the performance of the network in terms of (i) user equipment attachment procedures (ii) data plane transfer between the user equipment and the mobile network. 3. An investigation into the complexity and viability of using open source testbeds to develop novel techniques such as coexistence strategies for LTE operating in the unlicensed ISM bands.

Acknowledgments

Firstly, I would like to express my gratitude to my supervisor, Dr Joyce Mwangama, for her continued guidance and feedback throughout the duration of this project. Some of the outcomes would certainly not have been achieved if it weren't for the various resources she has provided and commitments she has made. In the same light, I was most humbled by the willingness of the Council for Scientific and Industrial Research (CSIR) to facilitate a collaborative effort between myself and their researchers. As such, my sincerest thankfulness must go my hosts at the CSIR including Chris Burger, Dr Albert Lysko and Mla Vilakazi for their kindness. The opportunity they provided is invaluable and has been a cornerstone for this project. I further extend my gratitude to Professor Andrew Wilkinson and Dr Simon Winberg for making themselves available when I needed advice.

Finally, I must thank my family and friends for their continued support and words of encouragement. My family especially has put this project in the forefront and have provided me with absolutely everything I could need in order to succeed. I cannot express enough how grateful I am and how much of an impact they have been.

Abstract

With the proliferation of mobile devices increasing at a rapid rate, the world is seeing an unprecedented growth in the demand for improved throughput and availability of mobile networks. Consequently, we have seen mobile communications technologies developing at an accelerated rate. With each new generation of mobile networks there as has been a growing level of complexity needed for implementation and an increased need for extensive and rigorous evaluation of system performance, link performance, and algorithmic efficiency under a variety of different realistic deployment scenarios in order to ensure that novel technologies are able to perform consistently in real world deployments. Furthermore, while link-level and system-level simulators have seen an increasing level of complexity and ability to carry out these evaluations, they often abstract or completely cut away parts of the processing chain, falling short of representing true network deployment. Thankfully, improvements in software defined radio technology and open source emulation platforms that fully implement various protocol stacks have enabled relatively low cost research and evaluation of novel technologies and techniques. This project seeks to investigate the values and capabilities of such open source emulation platforms through consolidation of the available platforms and implementation of an LTE network with true over-the-air radio links realised through USRP flavoured radio defined software. The capabilities assessed include the scalability, computational efficiency, and the ease of further development for enhancement of the platform. This is evaluated through rigorous testing of the different network configurations, generation of different IP traffic profiles for both emulated and commercial off the shelf user equipment, as well as the implementation of a simple coexistence strategy to enable LTE networks to coexist with co-located technologies in the unlicensed ISM bands.

Contents

List of Figures

List of Tables

List of Listings

Nomenclature

3GPP	3rd Generation Partnership Project
ACK	Acknowledgement
AIPN	All-IP Network
AN	Access Network
AP	Access Point
AUC	Authentication Centre
BS	Base Station
BSC	Base Station Controller
BTS	Base Transceiver Station
C-RAN	Cloud RAN
C-RNTI	Temporary Cell Radio Network Identifier
CA	Carrier Aggregation
CCA	Clear Channel Assessment
CN	Core Network
COT	Channel Occupation Time
COTS	Commercial Off-The-Shelf
CP	Cyclic Prefix
CS	Circuit Switched / Carrier Sensing
CSAT	Carrier Sense Adaptive Transmission
CSMA/CA	Carrier-Sense Multiple Access With Collision Avoidance
E-UTRA	Evolved UTRAN
ED	Energy Detection
EDGE	Enhanced GPRS
EIR	Equipment Identity Register
eNodeB	Evolved NodeB
EPC	Evolved Packet Core
FDMA	Frequency-Division Multiple Access
GERAN	GSM EDGE Radio Access Network
GGSN	Gateway GPRS Support Node
GPP	General Purpose Processor
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HARQ	Hybrid Automatic Repeat Request
HLR	Home Location Register
HSPA	High Speed Packet Access

HSS	The Home Subscriber Server
ICMP	Internet Control Message Protocol
IMEI	International Mobile Equipment Identity
IP	Internet Protocol
ITU	International Telecommunication Union
LAN	Local Area Network
LBT	Listen Before Talk
LTE	Long-Term Evolution
LTE-A	Long-Term Evolution Advanced
MAC	Media Access Control Layer
MCC	Mobile Country Code
ME	Mobile Equipment
MIB	Master Information Block
MIMO	Multiple Input Multiple Output
MME	Mobility Management Entity
MNC	Mobile Network code
MSC	Mobile Switching Centre
NAS	Non-Access Stratum
NR	New Radio
OAI	OpenAirInterface
OFDMA	Orthogonal Frequency Division Multiple Access
OFDMA	Single Carrier FDMA
P-GW	PDN Gateway
PCI	Peripheral Component Interconnect
PCRF	Policy and Charging Rules Function
PDCP	Packet Data Convergence Protocol
PDN	Packet Data Network
PDU	Cloud RAN
PHY	Physical Layer
PLCP	Physical Layer Convergence Protocol
PRB	Physical Resource Block
PS	Packet Switched
PSS	Primary Synchronisation Signal
QoS	Quality of Service
RAT	Radio Access Technology
RLC	Resource Link Control
RN	Relay Node
RNC	Radio Network Controller
RNS	Radio Network Subsystem
RRC	Radio Resource Control
RRM	Radio Resource Management

S-GW	Serving Gateway
S1AP	S1 Application Protocol
SAE	System Architecture Evolution
SCell	Secondary Cell
SCTP	Stream Control Transmission Protocol
SDR	Software Defined Radio
SDU	Service Data Unit
SGSN	Serving GPRS Support Node
SSS	Secondary Synchronisation Signal
TAC	Tracking Area Code
TB	Transport Block
TDMA	Time-Division Multiple Access
TTI	Transmission Time Interval
UE	User Equipment
UHD	USRP Hardware Driver
UMTS	Universal Mobile Telecommunications Service
UTRAN	UMTS Terrestrial Radio Access Network
VLR	Visitor Location Register
VNI	Virtual Network Interface
VoIP	Voice Over Internet Protocol
VoLTE	Voice Over LTE

Chapter 1

Introduction

1.1 Background to the study

With modern day smartphones playing the role of all-in-one devices offering both exceptional mobile computational ability and reasonably fast and reliable internet connections, it is no wonder that the world has seen such a paradigmatic shift in the applications and users of mobile networks. From 1990 to 2011, the global number of mobile subscribers grew from 12.4 million to an estimated 6 billion, representing around 87% of the global population [1] and generating roughly 51.5% of global website traffic [2]. With the strength of such a rapid upward trend not likely to cease anytime soon, both researchers and industry alike remain committed to investigating new mobile communications technologies and approaches.

One of the newer and most promising technologies offering not only to cater for the exponentially growing number of users and absurd data throughput demands, but also to address the concern and focus placed on energy consumption and carbon footprints, is that of fifth generation (5G) mobile networks. This generation aims to increase mobile data volume per area over predecessors by a factor of 1000, increase the number of devices connected to the network by a factor of 10-100, reduce the end-to-end latency by a factor of 5, and increase typical user data rates by a factor of 10-100 [3]. New approaches needed for the adoption of this technology includes cloud based radio access networks (RANs), the application of software defined network (SDN) principles, exploiting new and unused portions of spectrum, as well as the use of massive multiple-input multiple-output (MIMO) and full-duplex communications. With cellular providers only having started incremental deployment of 5G since 2019 [4], the technology is still considered to be in its infancy. As such, extensive and rigorous evaluation of system performance, link performance, and algorithmic efficiency under a variety of different realistic deployment scenarios have become increasingly important [5].

While a more appropriate manner of assessing the expected real world performance may lend itself to experimentation with commercial equipment, this severely restricts configuration capabilities. Even so far as restricting the implementation of novel ideas due to constraints posed by the commercial hardware vendors[was 1]. Furthermore, due to the high costs of developing and constructing prototype customised systems, it has become increasingly necessary to develop experimental simulation platforms which closely comply with the appropriate standards and protocols, at the same time taking into account different scenarios and stochastic factors in order to produce meaningful results [5]. While link-level and system-level network simulators provide useful analytical tools and have significantly evolved in past decades with regards to complexity and capability [6], they fall short of capturing and reflecting the nuances of real-time operations in radio frequency (RF) environments.

A strong candidate for circumventing the above-mentioned issues are full stack emulation platforms (known as testbeds) that fully implement all layers of the protocol stack in a real time operating environment, compatible with software-defined radio (SDR) front-ends to enable validation through highly realistic network scenarios and RF conditions. While commercial implementations of these emulators do exist, their closed source nature often present large barriers to entry for researchers. This has led to the development of open source modular testbeds such as OpenAirInterface(OAI) and srsLTE. Despite lagging behind their commercial counterparts and only having fully adopted 4G/LTE standards, they have drawn wide adoption for experimental and commercial purposes due to their extremely valuable open source nature. Given the practical requirement for backward compatibility between successive technologies, this comes as no surprise as it is rational to assume that novel 5G technologies will evolve as an extension to current LTE standards.

1.2 Objectives of this study

1.2.1 Problems to be investigated

For the purposes of this project, an investigation and implementation of the various open source emulation platforms will be done in order to realise a fully working over-the-air LTE testbed. The implemented testbed will be assessed for its functional capabilities and potential value to academic institutions and educational contexts. Furthermore, this project will evaluate the convenieve of using such platforms to validate novel techniques through the implementation of a simple proof of concept coexistence strategy for LTE signals in the unlicensed ISM bands. As such, the objectives of this project are to present:

- A literature review detailing the tools and theory needed to implement an over-the-air testbed.
- A fully functional over-the-air LTE testbed encompassing the full end-to-end LTE stack.
- A primitive coexistence strategy implemented the same open source LTE framework.

These deliverables will aid in answering the main questions posed in this project. Namely these are questions related to whether over-the-air testbeds are deployable in educational or academic setting and what value they hold, as well as questions related to their performance and ease of enhancement.

1.2.2 Purpose of the study

The main purpose of this study is to provide an extension of the work done by Khwezi Majola in 2017 in implementing a virtualised LTE network [7]. Investigation and implementation of an open source full stack LTE testbed not only provides value to the research done by Dr Joyce Mwangama in developing future mobile technologies, but will also act to lay the foundation for future over-the-air testbed emulation for educational purposes within the faculty of Electrical Engineering at the University of Cape Town. Furthermore, these platforms could provide value and opportunity to under resourced regions of South Africa through inexpensive small scale LTE deployments.

1.3 Scope and Limitations

Due to the rapid speed at which mobile communications technologies are developed, as well as the growing levels of complexity associated with each new generation of technology, it was important that this project follows a well defined scope. While implementation of a 5G testbed would have been ideal, the technology is still in its infancy and there are simply no open source 5G testbeds implemented. LTE on the other hand is quite stable, with various options for open source implementation.

Due to the limited time frame for completing the project, and the unpredictable nature of component delivery from international vendors, deciding to venture beyond the project's predefined scope would have resulted in insufficient time for production of any meaningful results. With the subject matter of mobile networks being a rather extensive and intricate field of knowledge with which I am not immediately accustomed, a sufficient amount of time for research, implementation and subsequent further investigations investigation had to be allocated. Further limitations included the availability of only one Ettus USRP. Consequent to this, I travelled from Cape Town to Pretoria where I collaborated with the Council for Scientific and Industrial Research (CSIR) and was graciously given the opportunity to use one of their own USRPs, as well as additional radio equipment. Testing at the CSIR took place within a limited timeframe from 26^{th} - 30^{th} October 2020.

Due to the limitations outlined above, the following scope was adopted:

- Implementation of a over-the-air tesbted will be limited to the LTE protocol stack, with investigations into the feasibility of deploying a 5G network using the same hardware and emulation platform remaining purely theoretical; based on work done in the Literature Review.
- Network configuration is limited to a single architecture whereby one GPP is responsible for hosting the RAN while the other is responsible for hosting the LTE core network. Absolutely no attempt will be made for multi-machine deployment of components within the core network.
- Only three scenarios scenarios will be designed for and tested. (1) A single cell single emulated user scenario where one attached UE will be connected to the network. (2) A single cell single COTS user scenario where one COTS UE will be attached to the network.(3) A single cell multi-user scenario where multiple COTS UEs will be attached to the network. Keep in mind that a single cell will comprise of only one BS (eNB and an interfaced SDR).

1.4 Plan of development

Chapter 1 presents an introduction of the subject matter and highlights the relevance, scope and objectives of this project. Chapter 2 provides a body of knowledge regarding mobile networks, tested emulation platforms and the challenges of coexistence that has been compiled through consolidation of the existing literature. Chapter 3 details the design methodology followed for implementation and experimentation of the testbed and coexistence strategy, while full details of the implementation are given in Chapter 4. Chapter 5 presents both qualitative and quantitative results following experimentation, while Chapter 6 discusses these in relation to the literature. Finally Chapter 7 and 8 present various conclusions based on these results and offer various recommendations for new or improved future implementations.

Chapter 2

Literature Review

2.1 Simplified Evolution of Mobile Networks

The natural progression of mobile telecommunications networks showcases a myriad of astounding and highly complex scientific and technological innovations. Current commercially available fourth generation (4G) mobile networks based on orthogonal frequency-division multiple access (OFDMA) technologies and Internet Protocol (IP) networks are hardly comparable to the original analog and circuit switched designs of the first generation mobile networks. Thus, this section seeks to provide a simplified overview of the evolution of mobile networks from 1G to next generational 5G technology, so far as to highlight the advancements made in mobile technology and architectural design that is prerequisite to the deployment of an LTE network.

2.1.1 First Generation Mobile Networks (1G)

First generation (1G) mobile networks were first made commercially available in Japan by Nippon Telegraph and Telephone (NTT) in 1979 [8]. These first-generation technologies made use of integrated architectures for their mobile base stations as they contained all radio and baseband signal processing. The circuit switched [9] architectural design and the adoptions of early frequency division multiple access (FDMA) [10] wireless standards schemes meant that limitations of 1G networks included poor spectral efficiency, as large gaps of spectrum were required between users, and support only being extended to one user per channel [11] as shown in Figure 2.1. These limitations often resulted in poor voice quality without much security, easily droppable call connections, unreliable handoffs, and maximum data throughput rates of 2.4 Kbps [12]. Ultimately, this created a scenario where poor scalability of networks [11] left operators with limited capacity for the rapidly increasing number of users.

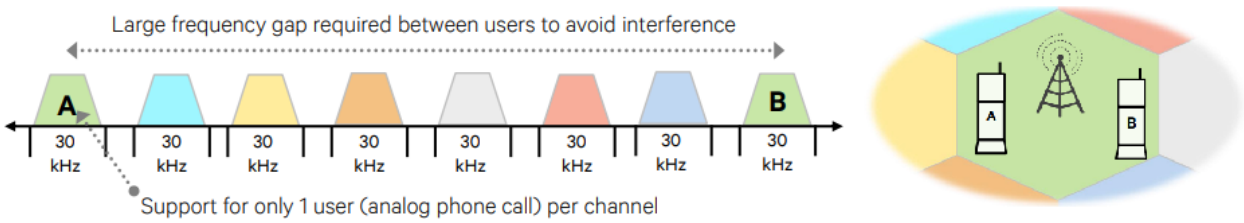


Figure 2.1: Showcase of FDMA technology utilised by 1G mobile networks.
Source: Adapted from [11]

2.1.2 Second Generation Mobile Networks (2G)

The adoption of the second generation (2G) of mobile networks marked a transition from analog to digital modulation schemes for both control signalling and data plane traffic [8], providing a collection of advantages and new features to mobile telecommunications technologies. These included advanced source and channel encoding, compressed data streams for transmission of voice data, and a high degree of resistance against interference and channel fading. Another important feature enabled by digital communications was that of providing data services over the mobile-communications networks [13]. The most pertinent data services introduced included the Short Messaging Service (SMS) and circuit-switched data services that enabled simple email communications.

Just as important as the introduction of digital modulation was the adoption of time division multiple access (TDMA) technologies, as shown in Figure 2.2, which enabled 2G networks to support more than one user per channel. This was achieved by allocating timeslots to each user for their specified frequency, allowing up to three users to simultaneously transmit voice data over the same channel. An enhancement made to 2G was the General Packet Radio Service (GPRS) [14] that offered an increased theoretical maximum transfer rate of 40 Kbps through IP packet transfers with external networks such as the internet. A further addition of 2G known as the Enhanced Data Rates for GSM Evolution (EDGE) [14] offered a greatly increased transfer rate with theoretical maximum transfer rate of 384 Kbps [10]. While the introduction of TDMA certainly improved throughout rates, network scalability, and spectral efficiency, TDMA still required large frequency gaps between allocated channels in order to reduce inter channel interference [11]. This left a substantial margin for possible improvement in future generations of mobile networks.

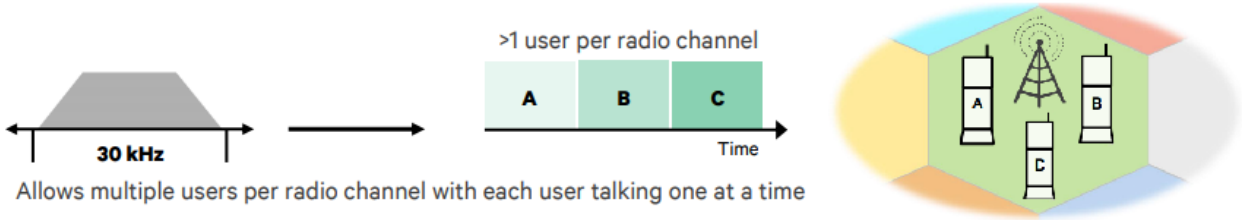


Figure 2.2: Showcase of TDMA technology utilised by 2G mobile networks.
Source: Adapted from [11]

i. Global System for Mobile Communications (GSM)

One of the most widely deployed standards of 2G networks was that of the Global System for Mobile Communications (GSM). Developed in 1982 by a grouping of several European countries originally known as Groupe Spécial Mobile [13], GSM was deployed globally in around 200 countries and at its peak managed to hold over 90% of the 2G market share [8]. One of the major advantages of GSM was the standardisation of interfaces and components within the GSM network. This allowed mobile network operators to purchase and successfully deploy different components within the network from a variety of different vendors [7]. Furthermore, GSM standards allowed for the distribution of intelligence

throughout the network, as opposed to integrated architecture found in 1G technologies, reducing the load placed on mobile base stations and enabling more efficient baseband signal processing.

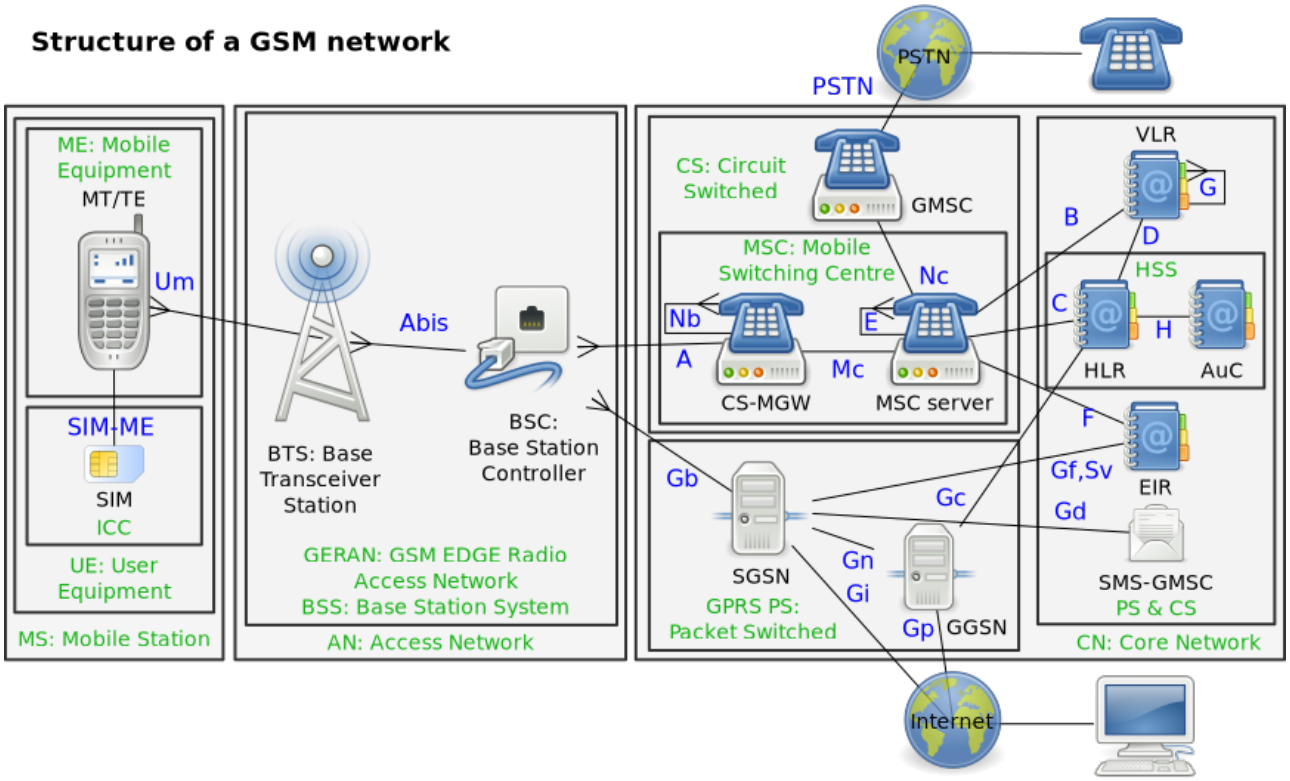


Figure 2.3: High-level overview of SM network architecture.
Source: [15]

As highlighted in Figure 2.3, the GSM network can be decomposed into several functional entities and subnetworks such as the Access Network (AN), Core Network (CN) as well as the the GPRS Switching Center and Packet Switched networks. Through this architecture, subscribers' Mobile Equipment (ME) are offered access through a radio link provided by the GSM EDGE Radio Access Network (GERAN), comprised of the Base Transceiver Station (BTS) and the Base Station Controller (BSC). Here the BTS is responsible for baseband signal processing of radio transmissions and reception, while the BSC is responsible for managing the radio-link and radio resources (such as radio-channel setup, frequency hopping, handovers [16]) between the BTS and ME for one or more BTSs. Further responsibilities of the BSC include managing traffic aggregation and the implementation of mobility management services across various centrally controlled BSSs.

Once a subscriber has established a radio-link connection with the Base Station System (BSS), user traffic is routed to the CN through the Mobile Switching Center (MSC) which implements various switching functionalities and subscriber management services such as registration, authentication, location updating, handovers between different BSSs and call routing [16]. These user mobility management and roaming services are more precisely handled by the CN through a collection of databases including the Home Location Register (HLR), Visitor Location Register (VLR), Equipment Identity Register(EIR), and the Authentication Centre (AUC). This is as the MSC itself does not store any data pertaining to MEs. Instead, the HLR is responsible for storing persisting subscription

information and some location information for all the subscribers registered with a particular network operator, enabling charging services and call routing. On the other hand, the VLR is responsible for storing information about users that are currently reside in the service area of a particular MSC, enabling roaming services between various different centrally controlled MSCs. Finally, connectivity between ME and fixed telephone users is also facilitated through the MSC by means of the Public Switched Telephone Network (PTSN). This further provides global connectivity between MSCs, thus providing global connectivity between mobile users [7].

2.1.3 Third Generation Mobile Networks (3G)

Naturally, as successor to the 2G mobile networks, standardisation of IMT-2000 by the International Telecommunications Union (ITU) was termed the third generation (3G) of mobile systems [17]. These specifications sought to increase network capacity through heightened spectral efficiency and implement a vast array of new services including video calls, wide-area wireless voice telephone, broadband wireless information as well as High Speed Packet Access (HSPA) data transmission with transfer rates of up to 14.4 Mbps and 5,8 Mbps in the downlink and uplink directions [17].

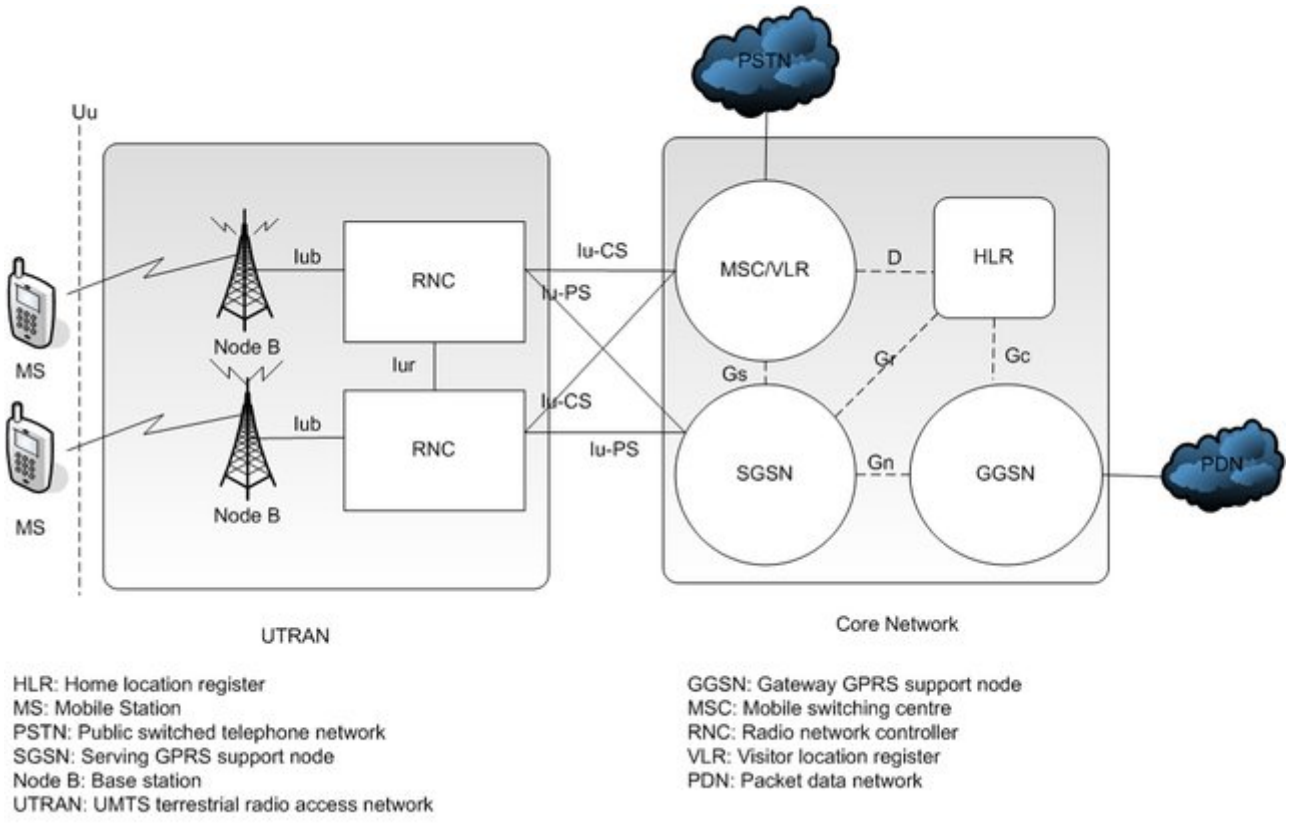


Figure 2.4: High-level overview of UMTS network architecture.
Source: Adapted from [18]

i. Universal Mobile Telecommunication System (UMTS)

Developed by the 3rd Generation Partnership Project (3GPP) as an evolution to GSM networks and the GPRS, the Universal Mobile Telecommunication System (UMTS) was widely adopted throughout Europe as a steady implementation 3G mobile networks as far back as 2000 [17]. Main components comprising the UMTS network include a packet-switched CN based on IP, as well as the UMTS RAN (UTRAN). In order to maintain backwards compatibility with older cellular devices, many operators maintained the legacy GPRS RAN in parallel to their UTRAN implementations. The overall UMTS network architecture inherited a number of logical network elements from previous GSM networks, with the purpose of these elements remaining similar but with some stark difference in implementation [19]. This is evident in Figure 2.4 where dashed and solid lines represent signalling and data links respectively.

UMTS RAN

The architectural design of the UTRAN showcased in Figure 2.4 remains inspired by the previous generation GERAN implemented in GSM networks. As such, the UTRAN is similarly compromised of a number of Radio Network Subsystems (RNSs) connected to each other through the Iur interface. Each RNS is further decomposed into several Base Stations, (BSs) known as NodeBs, and connected to a Radio Network Controller (RNC) through the Iub interface. At a high level, the RNS functionality remains equivalent to that of the BSS implemented in GSM by providing and managing the air interface for communication with user equipment (previously known as MS). RNC functionality is split between Radio Resource Management (RRM) functions such as handover control, power control, admission control, packet scheduling, and code management [20], as well as mobility management functions.

UMTS Core Network

As before, the UMTS CN remained inspired by the previous generation GSM CN. As such, the UMTS core architecture was implemented as a migration from GSM with the various subnetworks and subsystems, as well as a number of added elements needed to realise further functionality demanded by UMTS, with the CN remaining on both the packed switched (PS) and circuit switched service (CS) domains. In the CS domain, UMTS networks are able to remain connected to fixed telephone users via an interface with PTSN through the MSC which essentially remained the same functional unit implemented in GSM [21]. In the PS domain, UMTS connects to external packet data networks (PDNs) through the serving GPRS support nodes (SGSN) and gateway GPRS support nodes GGSN [18]. Although both of these entities provide session management functions, the GGSN is solely responsible for interfacing with external networks such as the internet, while the SGSN remains responsible for the delivery of data packets to appropriate BSs within its geographical service area through packet routing, as well as mobility management, logical link management, authentication, and billing amongst others [18, 21].

Other elements such as the home location register (HLR), equipment identity register (EIR) and authentication centre (AuC) are shared between PS and SW service domains [21]. Once again, these entities follow the same functional role as those implemented in the GSM network. In more detail, these entities are responsible for containing all the administrative information about each subscriber along with their last known location, performing network attachment authentication, as well as storing authentication keys contained within MS USIMs.

2.1.4 Fourth Generation Mobile Networks (4G)

The fourth generation (4G) of mobile networks was standardised as the Long Term Evolution (LTE) network by the 3GPP in release 8 of their specifications [22]. Subject to the demand for an exponential increase in mobile telecommunications capacity, a great deal of focus was placed on improving the overall architectural design of the LTE network. This resulted in the adoption of a new RAN developed by the Systems Aspects group known as System Architecture Evolution (SAE), a simplified flat architectural design with an all-IP Network (AIPN), separation of control plane and user plane traffic, and breakthrough technological developments including OFDMA. The resulting architecture was less hierarchical in nature and could be implemented with fewer nodes than previous generations. Furthermore, standardisation of the architecture and interfaces allowed for simpler deployments where components could be purchases from different vendors and allowed for entities of the network to full inter-work with related wireless technologies such as WCDMA, WiMAX, and WLAN. As the successor of UMTS, this new generation of mobile networks saw exponential increases in the targeted peak instantaneous downlink and uplink rates as they were set at 100 Mbps and 50 Mbps respectively for 20 MHz of allocated spectrum. This translates to a spectral efficiency of 5 bps/Hz for downlink and 2.5 bps/Hz for uplink. This was achieved through the adoption of OFDMA which sought to convert the wide-band frequency selective channel into a set of many flat fading sub-channels [23] and greatly increase the processing efficiency of receivers, while only requiring a reasonable level of complexity for implementation. This combined with the fact that OFDMA enabled frequency domain scheduling for increased spectral efficiency meant that numerous other benefits were spawned, including radically reduced latency and significantly less network congestion.

Further developments where made to LTE with the standardisation of LTE-Advanced (LTE-A) through release 12 by the 3GPP [24]. These developments saw the introduction of new functionality such as Carrier Aggregation (CA) [25], multi-antenna techniques for multiple input multiple output (MIMO) support [26], and support for Relay Nodes (RN) [27]. Consequent to this, LTE-A supported an improved spectral efficiency of 16 bps/Hz and 30 bps/Hz for uplink an downlink connections respectively, leading to theoretical peak instantaneous uplink and downlink rates of 1.5 Gbps and 3 Gbps using MIMO techniques [28].

i. Network Architecture

Following our understanding of where 4G/LTE networks fit into the perpetual development of mobile telecommunications technology, it is only fitting to explore the inner workings of LTE operations in more depth. This portion of the literature review provides a high-level overview of the architectural design of LTE networks, and will serve to provide an indication of the component implementation needed in order to realise a fully operational LTE stack. Following this, further details regarding LTE specifications and lower layer implementation of the LTE stack, such as the physical (PHY) and media access control (MAC) levels, are discussed in Section 2.2. Using the information presented in these complimentary sections, one should develop a holistic understanding of the end-to-end LTE stack.

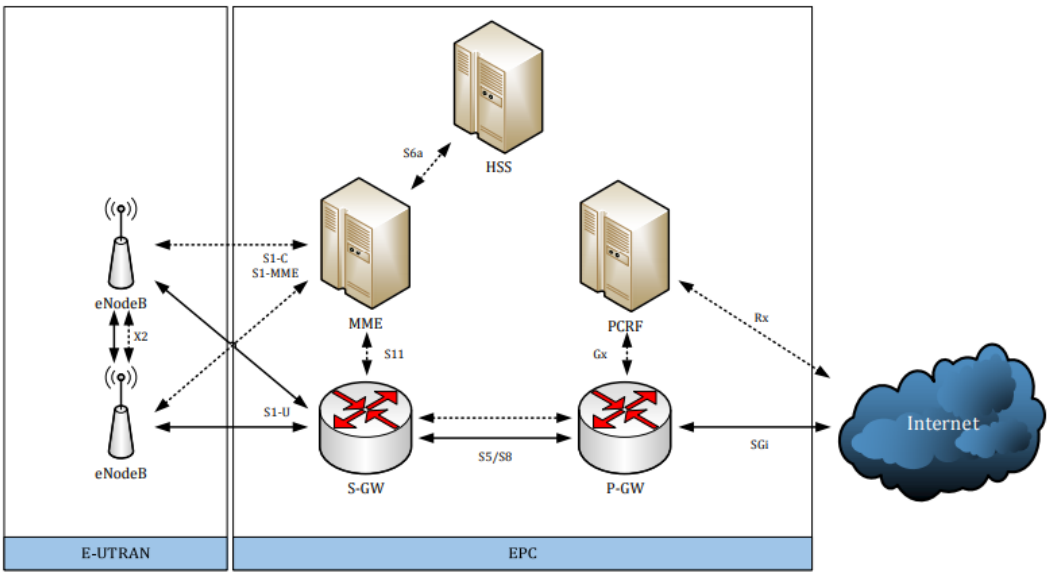


Figure 2.5: High-level overview of 4G radio access network architecture.
Source: [29]

The LTE network architecture is composed of three primary systems; user equipment (UE), the CN entity, more specifically known as the Evolved Packet Core (EPC) in LTE, and the RAN entity known a Evolved Universal Terrestrial Radio Access Network (E-UTRAN). The combination of EPC and E-UTRAN culminate in the evolved packet system (EPS) [29], showcased in Figure 2.5, representing the overall LTE network. Clearly evident from this is the simplification of the network architecture which has been referred to as a "flat" architectural design [29, 30]. While this has removed many logical entities such as RNCs from previously bloated architectures, this has resulted in the design of an evolved NodeB (eNodeB) which is significantly more complex and requires more computational power. In this configuration, the eNodeB is responsible for resource scheduling, as well as transmission and reception of the LTE signal, while the EPC is responsible for for handling UE attachment and establishing communication paths from UE to the internet [29]. For the purpose of implementation, both EPC and E-UTRAN are described in more detail below.

Evolved Packet Core (EPC)

Although the EPC has been designed to provide legacy support for 2G and 3G networks via SGSN nodes present in previous generations [7], the EPC implementation stands in stark contrast to its GSM/UMTS predecessors as it only supports the PS service domain. Furthermore, all nodes within the EPC are split between user plane and control plane functionality [29].

The main logical nodes of the EPC include:

- Home Subscriber Service (HSS)
- Mobility Management Entity (MME)
- Serving Gateway (S-GW)
- PDN Gateway (P-GW)
- Policy Control and Charging Rules Function (PCRF)

The HSS ia a central database responsible for maintaining subscriber information and performing user authentication. In order for authentication procedures to successfully complete, an exchange of information between UE and EPC must occure whereby XOR or MILENAGE algorithms are used to verify that information stored on the universal subscriber identity module (USIM) is consisten with the information stored on the HSS.

The MME is the main control plane node responsible for managing access of UE to the EPC through the E-UTRAN. Duties of the MME include [31] :

- Bearer management.
- Connection management.
- Security functionality for the E-UTRAN.
- Management of intra-system handover.
- Processing communication between UE and CN through Non-Access Stratum (NAS) protocols.

The P-GW is a user plane node primarily serving as a gateway for interfacing the EPC with external networks such as the internet. Following attachment with the network, each UE is assgined at least one P-GW which is done through dynamic or static IP address allocation for by the P-GW. Further responsibilites of the P-GW include QoS enhancement [31] according to the rules from the PCRF and data rate throughput enforcement. On the other hand, the S-GW is data plane node responsible for forwarding packets from the EPC to the eNodeB, acting as a gateway between the two. Further responsibilities of the S-GW include holding bearer information when UEs are in an idle state and temporarily buffering downlink data while the MME pages the UE to re-establish bearer paths [31]. While the S-GW and P-GW are specified as seperate logical nodes, they are often combined into a single node known as the SP-GW or SAE gateaway [29, 30].

Finally, the PCRF is a control plane node responsible for policy control, enforcing QoS and controlling subscriber charging functionalities according to the appropriate subscription profile and rules [32].

Evolved Universal Terrestrial Radio Access Network (E-UTRAN)

Unlike previous generations, the E-UTRAN is characterised by a network of BSs comprising only of single component. Namely, the Evolved NodeB (eNodeB). As such there is no centralised control unit responsible for handling connection with the EPC [33]; there exists a direct connection to the EPC via the S1 interface. As such, intelligence is said to have been distributed amongst the LTE BSs with control functionality being absorbed by the eNodeB on top of its baseband signal processing responsibilities. This has resulted in higher development costs and increased complexity of the eNodeB, but has also enabled faster network setup times for cellular providers [33].

As previously mentioned, the eNodeB is responsible for resource scheduling, both transmission and reception of LTE signals, as well as processing handover of UE to neighbouring eNodeBs. The latter is achieved through the optional X2 interface interconnecting eNodeBs, also responsible for the transference of load and interference information [32] [33].

2.1.5 Fifth Generation Mobile Networks (5G)

While current LTE/4G deployments are considered quite stable, there continues to be an exponential increase in the demand for network availability and performance driven by a growing user base, the proliferation of high bandwidth applications such as video streaming and cloud computing services, as well as the emergence of internet of thing (IoT) devices. In order to deal with this, the fifth generation of mobile networks (5G) sees a complete paradigm shift over previous mobile telecommunications technologies.

This paradigm shift realised through the 5G specification requirements [34] listed below.

1. A 10 to 100x throughput improvement over 4G and 4.5G networks
2. Significantly reduced latency of 1 ms.
3. A 1000x improvement in bandwidth per unit area.
4. A 100x increase in the number of connected devices per unit area over 4G LTE
5. A 90% reduction in network energy usage.

In order to achieve this goal, 5G dictates the need for serious bandwidth requirements. The solution to this requirement has been the implementation of a high frequency transmission that has become known as millimeter-wave [35]. While this offers extremely high bandwidth capabilities, several major disadvantages of millimeter-wave include propagation path loss due to higher carrier frequency, reduced scattering which reduces the available diversity, and critically weaker non line of sight paths [35]. Meaning that there will need to be a densification of cells as new 4G BSs, known as New Radio (NR) [36], are no longer able to serve larger geographical areas. This poses a challenge in terms of cost and complexity, requiring innovative solutions such as Cloud RAN (C-RAN) [37]. Despite the delay of Release 17 of these standards [38] and the challenges presented by 5G, the world has already seen 114 commercial 5G deployments [39].

2.2 Long Term Evolution (LTE) Specifications

Beyond the high level overview discussed in the LTE Network Architecture Section of this report, functionality with the LTE network is distributed throughout various levels and layers of the LTE protocol stack. This section presents a shallow overview of implemented transmission, as well as the various layers within the stack, aiming to construct a sufficient base of knowledge encompassing the most appropriate layers of the E-UTRAN stack needed for realisation of an LTE testbed, as well as implementation of a basic coexistence strategy.

2.2.1 LTE Protcol Stack

The E-UTRAN protocol stack (referred to from here on out as the LTE protocol stack) illustrated in Figure 2.6 is split into three main layers, further deconstructed into four main levels:

- Packet Data Convergence Protocol (PDCP) Level
- Resource Link Control (RLC) Level
- Multiple Access Control (MAC) Level
- Physical (PHY) Level

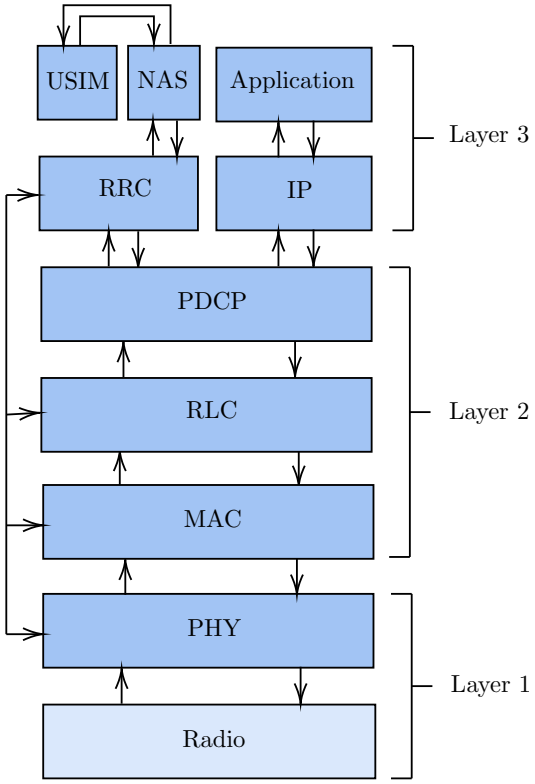


Figure 2.6: High-level overview of LTE protocol stack.
Source: Adapted from [29]

i. PHY Level

The PHY level is the lowest level of the LTE protocol stack and is ultimately responsible for ensuring the transfer of information from the MAC transport channels known as transport blocks (TBs) over the air interface between eNodeB and UE [40]. This is done through mapping of the TBs to the resource grid to and from the MAC level [29]. Other responsibilities of the PHY level include link adaption (AMC), power control, cell search for initialisation and handover purposes, as well as making network quality measurements for the RRC level [40, 32].

ii. MAC Level

The MAC layer takes on the task of multiplexing and demultiplexing between logical channels and transport channels. More explicitly, this entails multiplexing of unencapsulated MAC service data units (SDU) from different logical channels onto TBs, and demultiplexing of unencapsulated MAC SDUs from different logical channels onto TBs [40]. Furthermore, the MAC level implements error correction through the Hybrid Automatic Repeat Request (HARQ) process, manages priority handling between UEs by means of dynamic scheudling, reports buffer information to the eNodeB, as is also responsible for managing UE random access control [29, 40, 32]. An example tasks carried out by the NAS include the transmission of USIM information to the MME during attachment [29].

iii. RLC

The RLC is ultimately responsible for the transfer of upper layer PDUs, with the main responsibility of the RLC being packet segmentation and reassembly. There are three different RLC modes which will dictate the procedures followed for each function. These are namely the transport, unacknowledged and acknowledged modes [29]. To simplify, some other examples of the functionality of this level include the transfer of upper layer PDUs and error correction through ARQ (only applicable in the acknowledged mode) [41].

iv. PDCP

The PDCP level resides on both the user and control planes and offers various services to the upper levels [42]. These services include but are not limited to header compression and decompression, ciphering and deciphering and transfer of control plane data [29, 42].

v. NAS

The NAS layer is a functional layer at the highest level in the protocol stack and is primarily used for managing the establishment and continual communication with UE while it is moving. More descriptively, it is the protocol responsible for passing message between UE and the EPC. As such, core functionality of the NAS includes call control management, identity management, mobility management and session management [43].

2.2.2 Transmission Scheme and Frame Structure

The 3GPP LTE specifications define how data is transmitted, processed and controlled through all the layers of the LTE stack [was 44]. These methods were designed to achieve much greater efficiency compared to the older generations of mobile networks and are able to offer high data rates, low latency, quality of service (QoS) guarantees, and fairness scheduling. Since the specifications are quite extensive, specifications consolidated and described in this section shall be limited to those relevant for the understanding and implementation of an over-the-air testbed, as well as coexistence with Wi-Fi technologies.

i. Transmission Scheme

As mentioned in earlier sections, LTE makes use of the OFDMA transmission scheme for downlink transmission. In essence, OFDMA works by splitting the available bandwidth into a set of equally spaced subcarriers [29] onto which complex data symbols are mapped. In the case of LTE, subcarriers are placed 15 kHz apart. Unlike previous transmission schemes where guard bands were needed to prevent intersymbol interference, the OFDMA subcarrier are overlapping. The major difference being that subcarriers are orthogonal to one another as shown in Figure 2.7. This can be seen by the fact that there are no other subcarrier signals present at the peak of any particular subcarrier signal. Consequently, this makes for extremely efficient use of spectrum and for low complexity signal processing chains for UE [29]. This is quite attractive as it lowers the price needed to develop high data rate UEs.

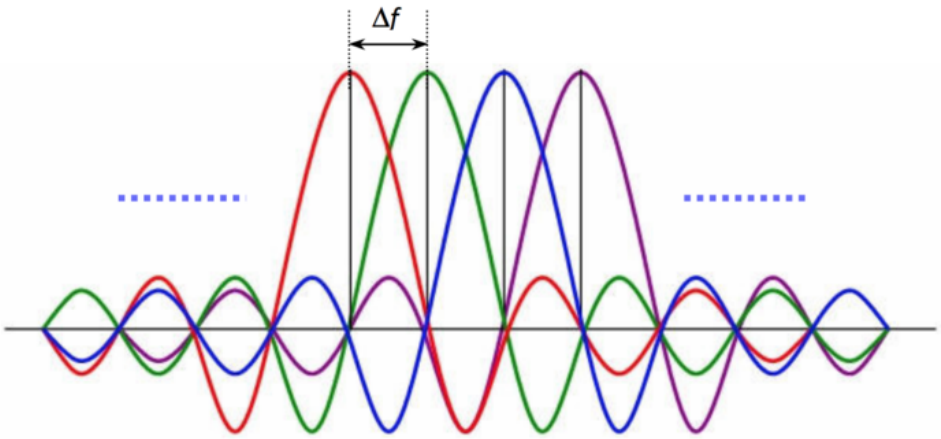


Figure 2.7: OFDMA transmission scheme subcarriers.
Source: [44]

On the other hand, Single Carrier FDMA (SC-FDMA) is used for uplink transmissions. In this case, instead of one complex symbol being mapped onto each subcarrier, symbols are mapped onto multiple subcarriers with information being multiplexed in the time domain instead. This in turn reducess the peak-to-average power ratio and preservers the battery life for UEs [29].

ii. Frame Structure

The LTE downlink transmission implements the OFDMA transmission scheme based on both FDMA and TDMA principles. As such, physical resources in LTE can be represented on a resource grid in both the time and frequency domains. At the least granular level, the LTE transmission in composed of frames [27]. In the time domain, frames constitute a 10 ms transmission, further decomposed into 10 subframes, each with a duration of 1ms. This is illustrated in Figure 2.8. Following this, each subframe is further subdivided into two slots of duration 0.5 ms. Contained within each slot are either six or seven OFDM symbols depending on the selected cyclic prefix (CP) length, with the extended CP length resulting in only 6 ODFM symbols present in a frame.

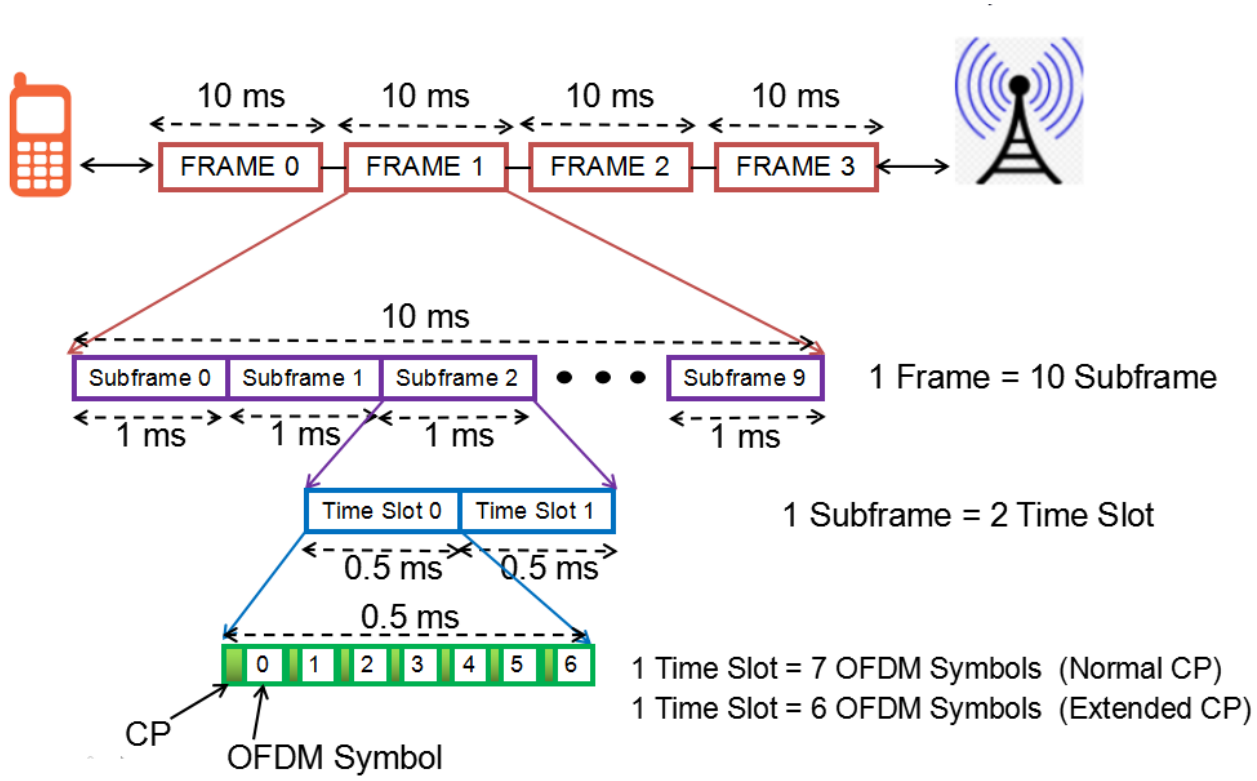


Figure 2.8: Breakdown of LTE frame structure.
Source: [45]

In the frequency domain, spectrum is composed of physical resource blocks (PRBs). Each resource block amounts to 180 kHz, constituting 12 subcarriers with a 15 kHz spacing. Thus the smallest element of transmission that can be allocated to UE is known as a resource block (RB), comprising the time duration of a slot and the bandwidth of a PRB. For the sake of analysis, RBs can be further subdivided, with the smallest representable element described as a resource element (RE). These are defined as single OFDM symbols modulated onto a subcarrier. Using this knowledge, one is able to combine the time and frequency domain concepts into an overall frame structure known as the LTE resource grid. An illustration of this is provided by Figure 2.9 whereby the lowest number of PRBs possible in an LTE network is represented. Taking into account the required guard bands, this amounts to a total LTE signal bandwidth of 1.4 MHz.

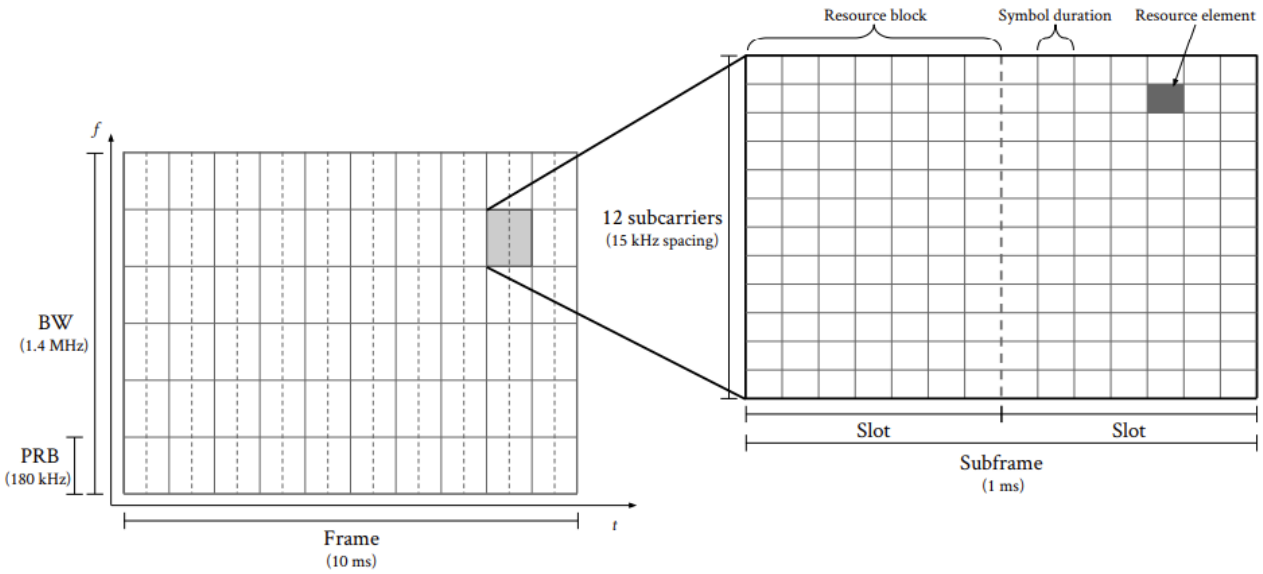


Figure 2.9: LTE resource grid illustrating the physical representation of resources in both the time and frequency domains with six available PRBs.
Source: [29]

As mentioned earlier in the analysis of the LTE Protcol Stack, TBs are passed from the MAC layer to the PHY layer where they are mapped onto the resource grid. The process of scheduling, whereby RBs are allocated to individual UE is performed by the eNodeB on a subframe level. Consequent to this, the assignment of resources is dynamic with a resolution of 1 ms. Additionally, TBs are delivered to the PHY layer once per transmission time interval (TTI), where a TTI corresponds to the duration of one subframe [29].

2.3 Testbed Platforms

Following the massively successful 3G mobile networks and the standardisation of the LTE/4G and LTE-A protocols by the 3GPP, testing and validation has played an essential role in evaluating expected real world performance of mobile system deployments. Due to the high cost of developing and constructing real systems, it has become necessary to develop experimental simulation platforms which closely comply with the appropriate standards and protocols. At the same time, these simulators need to take into account different scenarios and stochastic factors that are expected to impact system performance in order to produce meaningful results [5]. While the flat network architecture adopted by LTE reduces architectural complexity and moves away from previous networks' hierarchical structure [7, 46], this has lead to an increased complexity of components within the network, ultimately creating new challenges with regards to performance testing and simulation.

2.3.1 Link Level Simulations

Link-level simulations based on tools such as MATLAB, Labview, Ptlomey II and Simulink place their focus on a single link in the wireless system at the PHY level of the protocol stack, often purely assessing the performance of transmission between user EU and BS. This ultimately simplifies and cuts away the upper layers of the protocol stack [5], meaning that link-level simulation results may prove different to practical application as they cannot accurately reflect the real-world system performance when multiple links are present in the system.

One of the more prominent link-level simulators is the MATLAB based LTE physical layer simulator developed by the Vienna University of Technology [23]. The platform carries an open source license and is available free of charge for academic and non-commercial applications. While developed mostly in MATLAB, computationally expensive functions such as channel decoding are implemented in ANSI-C as MEX functions. The highly flexible nature of the simulator allows researches to fully implement and investigate novel algorithms and processing techniques in the LTE PHY layer with the advantage of providing real time operations and baseband signal processing for over-the-air testbeds. While this enables the system to act with a fairly high level of realism to investigate LTE link performance due to RF impairment [23], the simulator falls short of representing a real world deployment of the full LTE stack.

2.3.2 System Level Simulations

System-level simulation on the other hand is a common approach taken in which either the entire protocol stack or partial higher level chains of the protocol stack are implemented. These simulations are often analytical in nature [6], based on discrete-time events, rendering them incapable of real-time operation with radio front-ends, and are implemented using tools such as MATLAB and ns-3 [47]. Others simulators present themselves as packet-orientated network simulators which either model or abstract parts of the network stack. Examples of these include LTE-Sim [48].

In the case of higher level partial chain simulations, system-level simulations mainly focuses on the higher layers of the protocol stack such as the MAC layer, transport layer, TCP/IP layer, and application layer. Simulations here are meant to test scenarios with multiple UEs and BSs present in the network. Unfortunately, due to the large number of links present in such a simulation, most system-level simulations implement a simple abstracted PHY level [5]. While this works to reduce the time and resources needed for system-level simulation, abstraction of the PHY level cannot accurately represent real world conditions and scenarios or accurately assess transmission algorithms or coding and modulation schemes [5]. This may lead to a discrepancy the between expected and measured results of a system once deployed.

2.3.3 Emulation Platforms

One potential solution to the problems faced are highly configurable, scalable, and accurate LTE testbed emulation platforms [6]. Unlike system-level and link-level simulators, LTE testbed emulation platforms implement all layers of the protocol stack [5] in a real time environment and manner that respects frame timing constraints and the 3GPP specifications. Not only are they able to represent all layers accurately, but they are able to implement radio and core networks of the stack on both standardised commercial off-the-shelf (COTS) equipment such as software defined radio (SDR) modules, everyday cellular devices, as well as general-purpose processors (GPP). Thus allowing researchers and industry to test and validate highly novel technologies and algorithms in simulated environments closer to real world scenarios, producing more meaningful test results without incurring the extensive costs associated with developing real systems. This essentially allows for quick, repeatable and portable research of wireless communication systems from a system-level, while overcoming the abstract defects of link-level simulations [5].

While these testbeds play a vital role in the development of novel communications protocols [46, 6, 37, 49], such as those needed for 5G, as well as the development and evaluation of novel coding and modulation schemes [23, 50, 51], these platforms present potential beyond merely being an experimental tool for evaluating new mobile network standards and techniques. The Department of Communications Engineering [52] showed that testbeds can be used to give students a more well-rounded understanding of mobile telecommunications by encouraging them to design and implement their own mobile networks using open source tools. Another group of researchers [53] showed that by using LTE emulation testbeds, multi-thread parallel processing techniques could be developed to improve computational efficiency to allow for future potential large-scale testing of mobile networks. Finally, it was also shown that these testbeds could be used to simulate DDOS attacks within an LTE network, enabling researchers to develop new protections against such attacks [54].

All in all, there are many available testbeds platforms developed by industry and research groups alike including those developed by Virginia Tech University [55] and Ericsson [56]. Many of these test beds deploy custom routines developed in LabView, MATLAB or C++ and are not necessarily open source or fully compliant with 3GPP standards. While testbeds like GNU Radio are open source [6], testbeds like OpenAirInterface (OAI) developed by EURECOM and srsLTE developed by Software Defined Systems provide fully open source platforms that implement all elements of the LTE system architecture [6]. Another close competitor to these full testbed emulation platforms is that of Amarisoft's LTE 100, a commercial closed source platform. Available open source testbed platforms are more thoroughly explored in below.

i. openLTE

In its current version, openLTE [57] is an open source implementation of the 3GPP LTE specifications featuring an eNodeB and a simple integrated EPC supporting MME and HSS functionalities. The platform is well documented, easy to understand and modify, and is compatible with the Ettus B2x0 USRP SDR family. Unfortunately, the platform does not provide a full LTE stack and when compared to alternative solutions the platform is relatively incomplete. This is as several entities such as the UE are either still under development or simply not offered.

ii. OpenAirInterface (OAI)

OAI is an open source LTE testbed platform developed by the Mobile Communications Department at EURECOM [58] that fully emulates the entire LTE stack from the physical layer to the network layer and is designed to operate on GPPs with an SDR frontend architecture. At the time of writing, OAI provides a standard-compliant implementation of a subset of LTE Release 10 for the UE, eNodeB, MME, HSS, S-GW and P-GW components on standard Linux-based computing equipment (Intel x86 PC/ARM architectures) [59] and can be freely distributed by the Alliance under the terms stipulated in the OSA license model.

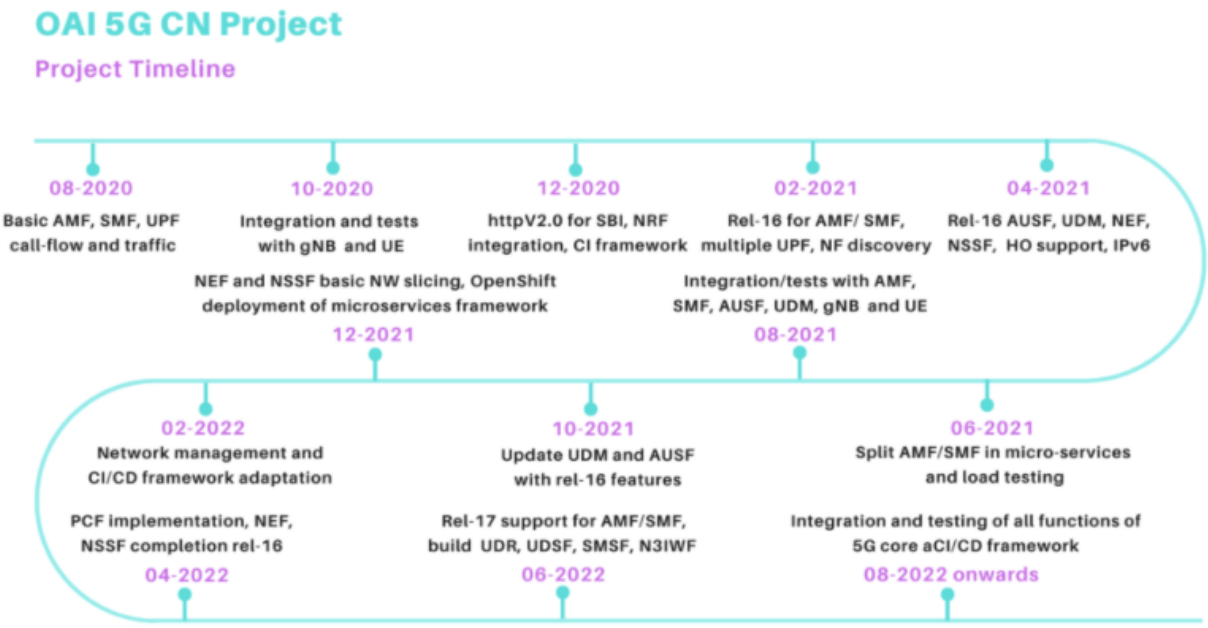


Figure 2.10: OpenAirInterface 5G CN roadmap.
Source: [60]

OAI's primary future objective is to provide an open-source implementation following on from the 3GPP Release 13 standardisation to provide a path towards 5G mobile testbeds which are freely available for experimentation on commodity laboratory equipment. This objective has been further realised by their recent commitment to providing a full 5G software stack by the end of 2020 [61] with the full roadmap for the 5G compliant CN illustrated in Figure 2.10.

The architectural design of OAI is illustrated in Figure 2.11 where it can be seen that each major component within the LTE stack has been realised as an individual entity. OAI has been designed such that nodes in the stack can be deployed within a single GPP or throughout a distributed processing environment, enabling large-scale scalability. Each node possesses a standard compliant interface that is capable of connection to real or faux traffic generation applications.

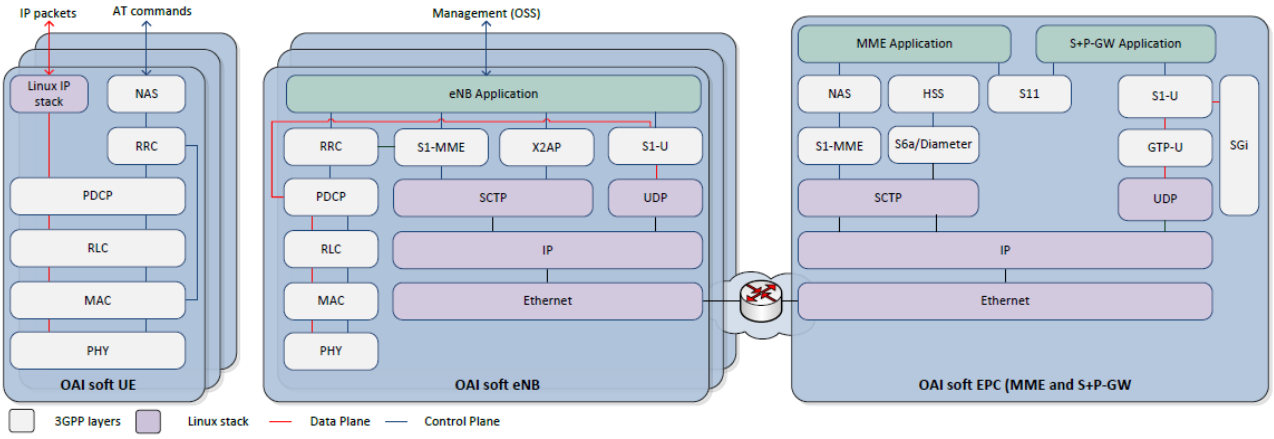


Figure 2.11: OpenAirInterface system architecture.
Source: [59]

The UE and eNodeB entities of OAI realise their transceiver functionality through SDR front-ends, while the GPP hosts remain responsible for baseband signal and network processing. Standard RF laboratory equipment supported by OAI includes:

- National Instruments hardware
- Ettus USRPs
- PXIe platforms
- Custom RF hardware provided by EURECOM

Beyond offering a true radio link connection to the LTE stack, OAI also offers an emulated radio link capability whereby the behaviour of wireless technologies in a real-world settings are simulated whilst respecting the temporal frame timing of the air-interface [6]. This is argued to be a unique feature of OAI as it allows for "a seamless transition between real experimentation and repeatable, scalable emulation" within a real-world execution environment [6]. For these purposes, two different emulation modes are realized. The first is a PHY level abstraction mode which simulates error events over the air-interface, while the second is a full PHY layer mode that involves the production of physical LTE signals over an emulated channel in real time. These channels implement 3GPP models comprising of path loss, shadow fading and small scale fading which interact with the mobility generator to perform different channel realisations. Both are meant to provide a means of testing without requiring an SDR front end and has been used to employ fully virtualised LTE networks [7]. To this end the full PHY implementation is more detailed and computationally expensive to run compared to its abstracted counterpart. This is primarily used for control testing prior to deployment with a radio front end and enables realistic validation and performance evaluation from both the link-level and system-level perspectives.

When placed directly against alternative testbed solutions, OAI is comparatively more complete in its implementation as it provides a rich software development environment with a myriad of control and measuring tools, message and time analysis tools, low-level log features, traffic generation tools, profiling tools, and soft scope [6]. Moreover, OAI supports a range of deployment scenarios including:

1. OAI UE ↔ OAI eNB ↔ OAI EPC
2. OAI UE ↔ OAI eNB ↔ Commercial EPC
3. OAI UE ↔ Commercial eNB ↔ OAI EPC
4. OAI UE ↔ Commercial eNB ↔ Commercial EPC
5. Commercial UE ↔ Commercial eNB ↔ OAI EPC
6. Commercial UE ↔ OAI eNB ↔ Commercial EPC
7. Commercial UE ↔ OAI eNB ↔ OAI EPC

There are unfortunately some distinct disadvantages to OAI. While the source code itself is relatively well documented, its complex and ill-defined repository structure makes it difficult to comprehend and customise [49]. Furthermore, personal experience eminating from this project regarding quiet mailing lists and poor support from developers is echoed within the literature [62]. This is compounded by a recent change in the OAI licensing and repository structure [63] which leaves installation and usage guides for the current CN implementation incomplete and obsolete.

iii. Software Defined Systems LTE (srsLTE)

The srsLTE platform, developed by Software Radio Systems [64], provides a full stack LTE testbed platform designed to be a high-performance LTE library for SDR applications. The library is available under both commercial and open-source licenses [62] with the highly modular nature and minimal need for inter-module or external dependencies making the platform perfect for researchers who want to easily customise, improve or completely replace core components without affecting the rest of the code [13]. The implementation is compliant with the 3GPP LTE Release 15 and supports various FDD and TDD network configurations for tested bandwidths of 1.4, 3, 5 and 10 and 20 MHz.

The codebase itself, illustrated in Figure 2.12 is well structured and is provided with up to date and continuously refined guides and documentation. All core components of the LTE stack including a lightweight EPC, eNodeB, and commercially verified emualted UE (tested against commercial LTE networks in Telefonica Spain, Three.ie and Eircom in Ireland) are implemented.

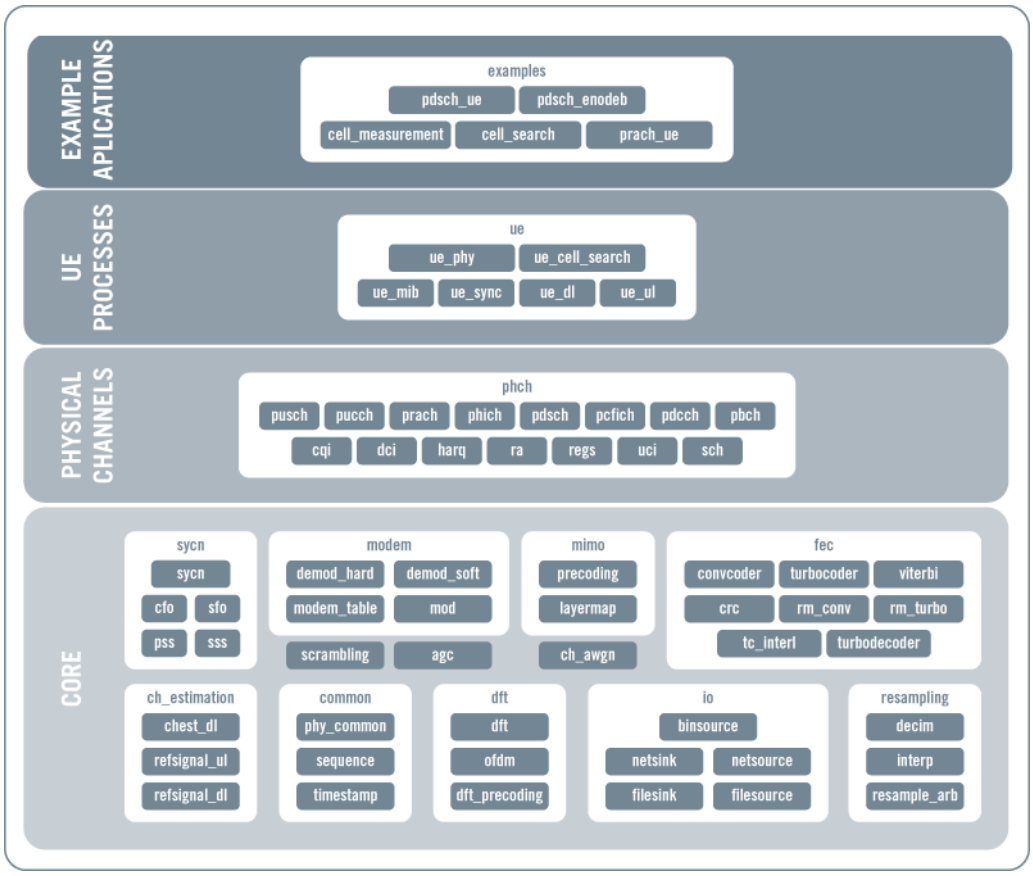


Figure 2.12: srsLTE application architecture.
Source: [64]

Figure 2.12 above shows the implementation structure of components within the srsLTE library. Each level of implementation and its constituents are discussed below using information adapted from [65]:

- Core: Core modules represent the main building blocks of the LTE PHY level. Included in this level are implementations of the convolutional coders and decoders, modulators and demodulators, synchronization techniques, channel estimators and reference signal generators, as well as OFDMA and SC-FDMA processing units.
- Physical Channels: Each module implemented here uses core building blocks to implement signal processing chains required to convert between raw data bits and samples ready for the digital conversion, with dedicated modules implemented for uplink and downlink channel processing.
- UE Processes: UE processes are implemented using physical channel physical channel modules for establishing uplink and downlink processing procedures.
- Example Applications: Provided by srsLTE are a number of example applications showcasing how to use the library. The most pertinent of which are the cell_search and pdsh_enodeb applications. The pdsh_enodeb application is dicussed in subsequent sections for the purpose of testing coexistence strategies.

Similarly to OAI, srsLTE is targeted to run on GPP hosts. Therefore, srsLTE deploys a number of tactics to maximuse computational efficiency. This includes extensive use of look-up tables (LUTs) for CRC and encoders, pre-generation of all scrambeling sequences, reference signals, and some Physical Uplink Control Channel (PUCCH) signals where memory access is more efficient that computation, pre-computation of interleaves, as well as data and instructional parallelism to make efficient use of CPU cores. More specifically, parallelism techniques are used extensively in the turbo decoder, channel estimation and equalizer, as well as the demodulator parts of the receiver as these represent one of the most complex links in the processing chain. While a parallel computing solution was experimentally developed for OAI's downlink processing chain [53], it remains unclear whether this technique was adopted into the core OAI codebase.

Furthermore, personal experience working with the platform proved the existence of an active mailing list as well extremely helpful and timeous support from developers. Going forward, srsLTE will be the preffered method of implementation for this project.

2.4 IEEE 802.11 (WiFi) Specifications

One of the most widely adopted protocols for wireless internet connectivity has been that of the IEEE 802.11 specification. While the first release of the specification can be traced as far back as 1997 [21], current releases of the 802.11 specification remain the basis on which WiFi devices are built. Through this, we are able to connected mobile devices such as laptops, cellphones, and even toasters to our local area network without requiring a physical connection. A high level overview of the IEEE 802.11 specifications are given. For the purposes of this project, only specifications relevant to the idea of wireless communications coexistence will be explored.

Similarly to other wireless communications technologies, WiFi makes use of OFDM digital modulation schemes that divide the spectrum into multiple OFMA subcarrier [66] with a total bandwidth spanning a multiple of 20 Mhz. In more detail, WiFi adopts a hierarchical channel bonding scheme whereby 20 MHz subchannels are combined in a non-overlapping manner. Unlike other wireless communications systems such as LTE, WiFi protocols have been designed with the issue of coexistence in the forefront. This is as Wifi is intended to operate in an unlicensed shared spectrum. Consequently, there is no guarantee that WiFi devices will operate in optimal RF conditions without interference from neighbouring devices. As a solution, WiFi specification mandate the implementation of discontinuous transmissions schemes in order to avoid causing continual interference with co-located networks. These include contention-based protocols known as Carrier Sensing Multiple Access with Collision Avoidance (CSMA/CA) [66] to estimate channel conditions and availability before every transmission. This protocol dictates that WiFi nodes must to listen to the shared medium to determine if another another node is busy transmitting. This is more clearly known as Clear Channel Assessment (CCA).

2.4.1 Clear Channel Assessment (CCA)

Two of the major functions utilised in order to realise CCA is that of Carrier Sense (CS) and Energy Detection (ED). Both are used as means of detecting the presence of transmissions of co-located devices. Should either of these two detection algorithms report that the channel is busy, the node must postpone its transmission and wait for a free DCF Inter-Frame Space (DIFS) or Arbitration Inter-Frame Spacing (AIFS) if QoS is enabled, plus a random back off time to avoid packet collisions to ensure collision avoidance when multiple nodes have sensed the medium to be idle and also wish to transmit [67]. Following a successful transmission, the WiFi node waits for an ACK message.

Carrier Sense (CS)

More formally, CS is described as the ability of the WiFi receiver to listen to the medium and to detect and decode incoming WiFi preamble. Through this, WiFi nodes are able to detect the presence of other WiFi devices transmitting in the same spectral space. Following the detection of WiFi preamble, the power level of the transmission is calculated. Should this signal have a power greater than or equal to -82 dBm [66], then CCA reports the channel as busy for the timeslot that is indicated in the frame's Physical Layer Convergence Protocol (PLCP) header length field.

Energy Detection (ED)

Should the CS functionality fail and an incoming transmission not be decoded, ED is used as a subsequent line of defence to determine the appropriate response measures for the WiFi node. Quite simply, ED is used to determine the energy level in the operating channel based on non-WiFi signals or corrupted WiFi signals that are sensed on the same operating band [67]. If the detected energy level is greater than or equal to -62 dBm [66], CCA reports the channel as busy. Importantly, ED must estimate the energy level of the relevant channel every time slot as the length of time for which the medium will be busy cannot be decoded.

2.5 Coexistence in the Unlicensed ISM Bands

Currently, huge increases in the demands for data throughput and mobile network network availability is driving many operators to improve spectral efficiency and increase spectral capacity. Studies suggest that 1.2 – 1.7 GHz of additional spectrum is required to meet the growing mobile data traffic [6]. While seeking the acquisition of additional licensed spectrum would seem like an appropriate approach for mobile operators as it guarantees predictable QoS and control over the network, this has proved quite challenging as the simple shortage of available spectrum has led to an exorbitant cost. In South Africa, the latest round of bidding for spectrum held by the Independent Communications Authority of South Africa (ICASA) [68] indicated they would make 406 MHz of spectrum available for the provision of mobile telecommunications in South Africa in the 700MHz, 800MHz, 2,600MHz, and 3,500MHz frequencies, with a reserved lot price of R 527 million for 10 MHz [69]. As the price of spectrum is forecast to increase as the availability of licensed spectrum becomes increasingly scarce, operators will have to find alternative measures of increasing their spectral capacity and efficiency.

One potential solution that spectrum scarcity has driven mobile network operators to consider is utilising the unlicensed ISM bands in conjunction with licensed bands to offload limited and expensive licensed spectrum. This would significantly improve coverage and improve spectral efficiency [66], and would serve to enhance the experience offered to users of mobile networks by offering higher data rates in dense network deployments. This would also create numerous challenges as transmission of LTE in these bands would negatively affect existing technologies operating in the space unless appropriate coexistence strategies are developed and deployed. This is evident from LTE and WiFi protocols presented in the Long Term Evolution (LTE) Specifications and IEEE 802.11 (WiFi) Specifications Sections of this Report as LTE transmissions where designed as a scheduled transmission with sole use of the available spectrum, while WiFi technologies have been designed with tolerance for co-located transmissions. Consequently, the eNodeB node will work to schedule transmissions regardless, and somewhat unaware, of the presence of co-located WiFi networks. Should WiFi and LTE transmissions operate in the same spectrum, the LTE network would act to monopolise wireless resources, resulting in starvation of the WiFi network. This is confirmed through a study by Abinadar et al [70] which found that the average throughput for LTE networks coexisting Wi-Fi networks remained rather similar to that of LTE networks operating in the licensed spectrum in a non-coexistence scenario, and that the average Wi-Fi throughput suffered from degradation leading to a reduction in throughput. Concerns have already been raised by Federal Communications Commission (FCC) [71] and the WiFi Alliance [72]. Despite this, Companies such as Qualcomm and Huawei are already working on developing mechanisms for their LTE and LTE-A technologies to be transferred onto the unlicensed ISM bands [73] with a concept known as LTE-Unlicensed (LTE-U). Unfortunately, no clear consensus has been reached on the impacts all of these radio access technologies (RATs) may have on one another [62].

As such, experimentation and extensive evaluation of coexistence strategies in realistic scenarios is essential in ensuring fair coexistence strategies that are in line with appropriate regulation. Up until this point, Studies on coexistence of different RATs have been largely simulation based and fall short of accurately representing the performance of these coexistence strategies in real world environments [62]. Full stack LTE platforms such srsLTE and OAI now enable reconfigurable broadcast frequencies and schemes. Consequently, these now serve as a vital experimental platforms to both analyse the effects of coexistence and to evaluate novel coexistence strategies.

2.5.1 LTE-Unlicened (LTE-U)

The LTE-U specification was conceptualised and designed by the LTE-U Forum [74] comprised of key members such as Qualcomm, Verizon, Ericsson, and Samsung, and was designed to address the idea of coexistence while closely following specifications released by 3GPP. Key specifications published by the forum include the minimum performance for coexistence when operating in unlicensed spectrum. Consequently, LTE-U is expected to be the first coexistence technology implemented directly into the lower layers of the LTE stack [75].

Fundamentally, LTE -U is design as a complementary or supporting data pipeline in small cell deployments where demands for user plane data is great [76]. Meaning that LTE-U deployments are intended for use as a secondary cell (SCell) within the LTE carrier aggregation framework.
At it's core, LTE-U uses a duty cycled version of the LTE waveform as illustrated in Figure 2.13 which would serve to provide WiFi nodes with opportunities for transmission. This is combined with a host of algorithms to improve coexistence between LTE-U networks and Wi-Fi networks. The LTE-U access point (AP) listens actively to Wi-Fi and other LTE-U transmissions to estimate the network usage patterns. Active reception of Wi-Fi transmission implies that it can interpret channel type (primary/secondary), packet type, packet length, etc. This information is used to evaluate channel activity, and consequently for dynamic channel selection and adaptive duty cycling. This adaptive duty cycling is realised through a real-time algorithm called carrier sense adaptive transmission (CSAT) and allows for the duty cycle to be modified by changing the TON and TOFF values. The resolution of duty cycling is equivalent to 1ms LTE subframe boundaries.

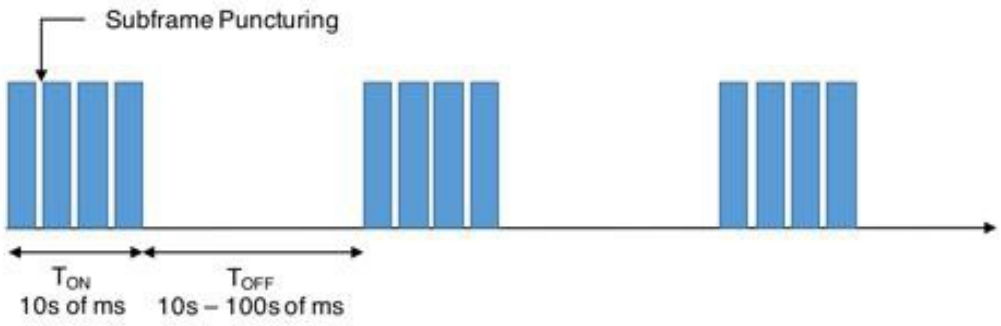


Figure 2.13: Duty cycled LTE waveform used by LTE-U.
Source: [77]

Importantly LTE-U has been designed to adhere to applicable regulatory requirements for unlicensed band operation, such as limits on transmit power spectral density and channel bandwidth occupancy. Unfortunately, as LTE-U does not employ any listen before talk (LBT) procedure required in more stricly controlled countries, LTE-U is targeted primarily at countries such such as USA, Korea, China, and India where LBT is not required for unlicensed channel access. Importantly, one of the design requirements for LTE-U is that it must achieve fair coexistence with other nodes of either the same or a different technology.

2.5.2 LTE-Licensed Assisted Access (LTE-LAA)

LTE-LAA is a standards based approach to deployment of LTE in the unlicensed spectrum developed by the 3GPP and specified as a part of LTE Release 13 [78]. Similarly to LTE-U, LTE-LAA is deployed as a SCell for downlink communications wherein initial implementations use licensed carriers aggregated with unlicensed carriers, enhancing downlink throughput through opportunistic offloading.

Key difference between LTE-U is that LAA is designed for worldwide operations and hence includes LBT procedures in order to be compliant with strict regulatory requirements governing power spectral density and maximum channel occupation time (COT) (which for example is 4 ms in Japan. This is shorter than an LTE frame transmission of 10 ms) and issues surrounding continuous transmission in the unlicensed spectrum. Furthermore, LAA has been designed with sufficient configurability in mind to enable operations in different regions where regulations are quite distinct from one another [77]. New functionalities implemented by LTE-LAA over LTE-U include clear channel assessment based on LBT procedures, discontinuous transmission with limited maximum transmission duration, and dynamic frequency selection for radar. These functionalities have major impacts on the LTE stack ranging from physical channel design, CSI estimation and reporting, HARQ operations and radio resource management (RRM), hence making LTE-LAA a much more complex solution when compared to LTE-U.

Chapter 3

Methodology

The process of designing and deploying a full LTE stack that allows for over the air transmission, successful attachment with COTS UE, and meaningful user plane data transfer culminates in a rather complicated developmental path. The requirements for an intimate understanding of all LTE components and procedures is further compounded when considering the enhancements that need to be made to the LTE stack for fair and successful coexistence with co-located radio access technologies (RATs). The objective of this section is to provide an outline of the methodological processes adopted for determining the appropriate system requirements, designing the system and network architecture, as well as performing validation and performance testing of the system.

3.1 Outline of Methodological Process

This section is dedicated to providing an overview of the design process followed from initial design to systems testing and verification, as well as to providing the appropriate background needed to rationalise the selected design choices.

3.1.1 Background and Scope

As mentioned in earlier sections, the scope of this project is an extension of previous work done by Khwezi Majola [7] and encompasses the design and implementation of an over-the-air LTE network in a university laboratory environment. The network must be realized through open source LTE testbeds to comprise a full LTE stack including at least one EPC, one eNodeB, one or more virtualized UEs operating through RF frontends, as well as one or more COTS UEs. As such, these testbeds must employ USRP flavoured SDR frontends for real-time LTE signal transmission and reception for eNodeB and UE components of the LTE network. The network should support a distributed computing paradigm whereby components of the LTE stack are implemented in different physical or virtual machines. This ensures that the system load is fairly distributed to ensure optimal performance of the network, as well emulate a realistic network deployment by real-world network operators. Furthermore, components provided by the open source testbeds should ideally be compatible with a variety of other open source platforms and commercial vendors to emulate reconfigurability of real-world LTE deployments. Once the network has been deployed, a variety of experiments must be run to validate and evaluate the performance of the network. An overview of all prerequisite knowledge for deployment of an LTE network and the available open source testbeds has been provided in the Literature Review chapter of this report.

Consequent to the deployment of the network in a laboratory environment where one cannot assume the availability of full RF shielding, the scope of this project has been extended to investigate the implications and ability of open source platforms to investigate possible coexistence strategies of LTE transmissions in the unlicensed ISM bands with co-located RATs such as IEEE 802.11 Wi-Fi technologies. This serves to ensure that radio transmissions in uncontrolled environments are fully compliant with the appropriate regulations surrounding transmissions in the licensed spectrum, as well as to showcase the reconfigurable and adaptable nature of open source LTE testbeds. This works to prove that the selected testbed allows researchers to perform realistic RF validation and performance testing for novel technologies at a relatively low monetary and time cost for development.

3.1.2 Methodological Process

The design process employed for this project follows that of the V-Model [79]. While the V-Model's intended purpose is to represent a commercial system's development lifecycle and is more prominent in software engineering projects, the model has been selected to provide a framework for the development process of this project. This is as the model provides clear and intentional relationships between different phases of the system planning, development and validation. This is an important feature for this project as each component of the LTE stack and phase of the project needs to be fully validated and evaluated. Hence, the V-Model process followed for design and validation of this project is represented by Figure 3.1 and is explored in detail in subsequent sections of this chapter. The left side of this diagram represents the decomposition of requirements and specifications for the system in an increasingly granular manner, working from the overall concept of operations down to the subsystem design. A direct relationship between these requirements and specifications and the overall validation of the system is indicated by means of dotted arrows. This clearly highlights how each level of the system is validated and how performance is evaluated at an increasingly granular level, working up from subsystem testing until validation and performance testing of the full LTE stack.

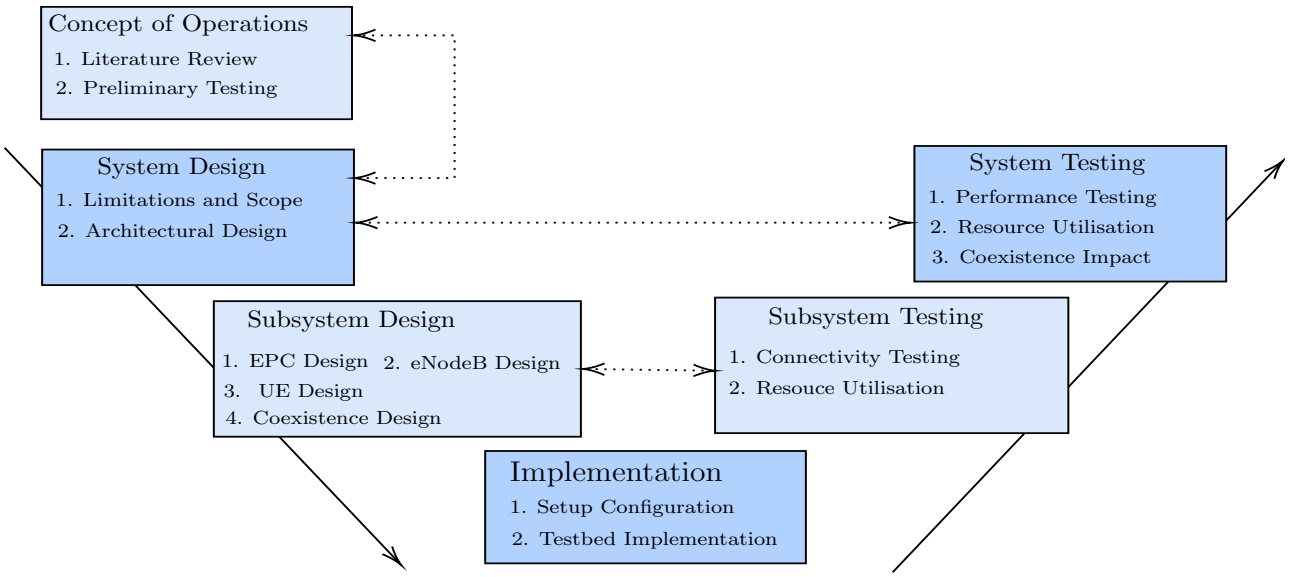


Figure 3.1: V-diagram showcasing the project progression phases.
Source: Adapted from [7]

3.2 Concept of Operation

In the development lifecycle of a product, the Concept of Operations phase is intended as a dedicated time for communication with the customer to ascertain the user needs of the system, thoroughly describing the operating environment and developing a well-rounded understanding of the system requirements and specifications. In the context of this project the Concept of Operation phase was used to perform the necessary research for understanding the progression of mobile telecommunications technologies, specifications and implementation of 4G/LTE network architectures, lower level MAC and PHY processing chains, various LTE testbed platform offerings, specifications of IEEE 802.11 Wi-Fi technology relevant to coexistence, as well as various LTE enhancements and algorithms relevant to coexistence in the unlicensed ISM bands. Information acquired from this research was compiled into the knowledge base presented in the Literature Review section of this report and was used to develop a holistic understanding of the entire LTE stack and the challenges presented for coexistence scenarios.

3.3 Architectural Requirements and Design

Specifications released by the 3GGP for the LTE network architecture are well defined and fully encompass the LTE stack. As explored in the literature, these specifications mandate the existence of two main components at the highest level of the LTE architecture. Namely, these components are the RAN and EPC. As such, the overall architectural design for this project must meet the LTE architectural specifications mandated by the 3GPP, shown previously in Figure 2.5. Furthermore, the implemented testbeds must meet a number of functional requirements illustrated in Table 3.1 including over-the-air capabilities to enable COTS UE attachment.

Functional Requirement Number	**Functional Requirement**
FR-001	The mobile network must fully implement the entire LTE protocol stack including both the RAN and core network components.
FR-002	The mobile network must be implemented using fully open source LTE testbed platforms.
FR-003	The mobile network must allow easy reconfigurable operation.
FR-004	Components of the network must be support deployment across physically separate host machines.
FR-005	The network must allow for successful attachment from COTS UE.
FR-006	The network must allow a sufficient level of adaptability in order to accommodate the development of LTE coexistence strategies.

Table 3.1: Functional requirements of the system.

Unfortunately, there are several design limitations to consider relating to project requirements, limited development time, hardware availability, as well as the associated licensing schemes of the open source frameworks. Prior to the final design of the network, a considerable amount of time was spent planning for and researching the limitations of the available platforms for implementation in order to reduce the risk of failure. The findings of this research are presented in the Testbed Platforms Section of the Literature Review. Due to these limitations, implementation of components within the network are limited to:

- Fully open source frameworks which place no restriction on non-commercial use.
- Components that are fully compliant with at least a subsection of 3GPP Release 8 [22].
- Components that are capable of running in a distributed deployment environment.
- RAN implementations that are compatible with the USRP B210 and USRP 2994 SDR devices.

Keeping the aforementioned design limitations in mind, the final architectural design adopted for this project is to remain somewhat static as the srsLTE platform will primarily be used for implementing components comprising the LTE stack, with a secondary implementation of the OAI platform. In slightly more detail, srsLTE will be used exclusively to implement the EPC and will also serve as the primary implementation for the eNodeB and emulated UE components. Following this, there will be secondary implementation of an OAI based eNodeB and emulated UE. Implementation of EUs will also include COTS UE. Figure 3.2 provides a high level overview of this overall architectural design while full details regarding the design and implementation of particular subsystems are presented in subsequent sections.

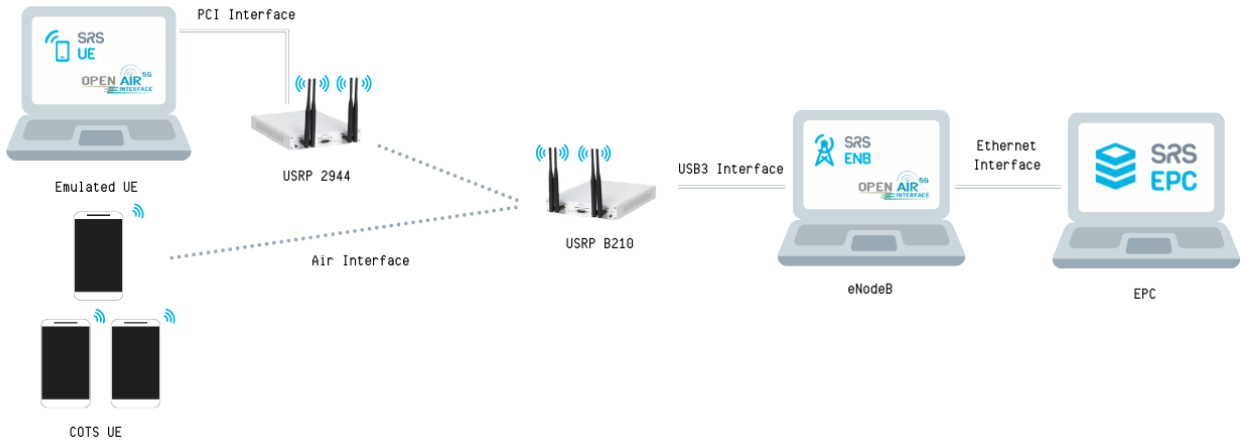


Figure 3.2: Distributed LTE network architecture.

As shown, the architectural design is indeed in line with previously listed functional requirements. This is as components are deployed in distributed manner using open source platforms and realising an over-the-air radio link. While virtualised components of the network such as the EPC and eNodeB could be co-deployed in the same machine through either separate virtualised environments or even the same local environment, a distributed deployment was selected for this project to more closely emulate a real world deployment scenario. As such, the network is configured to run the emulated UE, EnodeB, and EPC on native Linux installations on physically separate machines. Consequent to this,

connection between the eNodeB and EPC components is realised through an Ethernet connection on a local area network (LAN). This is somewhat representative of a real world deployment where fibre optic cables are used as a backbone to support IP user plane and control plane traffic between physically seperated eNodeBs and EPC. Communication between the emulated UE and eNodeB is facilitated through the USRP 2944 (graciously provided by the CSIR [80]) and USRP B210 SDRs deployed as radio front ends for the eNodeB and UE components respectively.

3.4 Subsystem Requirements and Design

This section details the requirements and design for the various components found within the LTE architecture. In the interests of reducing the length of the report, details of design for these components will be limited to the srsLTE platform as this has been chosen for primary implementation.

3.4.1 Core Network (EPC)

Requirements for the CN include full compliance with at least a subset of 3GPP Release 8. As such the EPC internal architecture should be split into the standard specified components and should support eNodeB attachment through the S1 interface. Consequently, the virtualised core network implemented is that of the SRS EPC provided by Software Radio Systems and their srsLTE platform [81]. The SRS EPC is designed to be a lightweight implementation of the complete LTE CN and runs as a single binary (opposed to the separately configurable and executable binaries implemented by OAI [59]) that implements the key EPC components such as the HSS, MME, S-GW and P-GW. Figure 3.3 depicts the high-level architectural design of the SRS EPC that is fully compliant with 3GPP LTE specifications. The various standard compliant subsystems and connecting interfaces are apparent in the figure.

While the SRS EPC has been selected for implementation of the EPC for it's easily re-configurable and implementable nature, there remain some limitations of the SRS EPC that will limit the scope of testing for this deployment. This includes the fact that VoLTE (Voice over LTE) functionality has not been implemented by srsLTE, meaning that mobile voice functionality would have to be carried out exclusively through external Voice over Internet Protocol (VoIP) services.

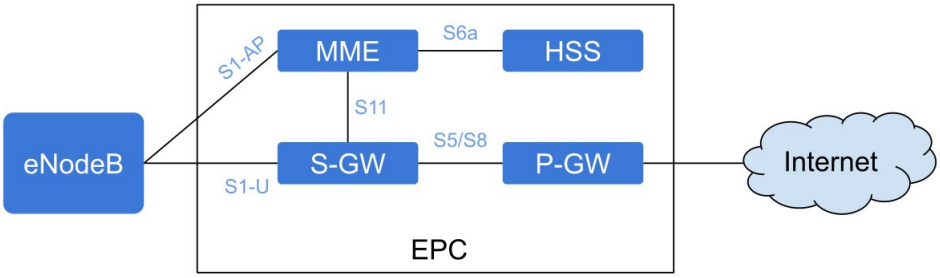


Figure 3.3: SRS EPC core network architecture.
Source: [82]

i. Home Subscriber Service(HSS)

The HSS entity of the SRS EPC implements a simple CSV based database that stores information such as the user's Mobile Station Equipment Identity (IMEI) number, authentication key, authentication algorithm specification of either XOR and MILENAGE authentication, settings for determining dynamic or statically allocated IPs for UE, as well as the user assigned usage limits. By consolidating this database, the CN carries out services such as authentication and authorisation of UE attachment.

ii. Mobility Management Entity (MME)

The MME entity of the SRS EPC acts as the main control element of the EPC and provides support for 3GPP standard compliant Non-access Stratum (NAS) and S1 Application Protocol (S1AP) protocols to provide control plane communication between EPC, eNodeBs, and UEs. It is responsible for handling user mobility, processing of user attachment and control messages, allocation of temporary identities to UEs, and paging UEs that are in idle mode. Other features included in the SRS EPC MME entity at the NAS level include [82]:

- Attachment procedures, detachment procedures, and service request procedures.
- NAS Security Mode Command, identity request/response, and authentication.
- Support for the setup of integrity protection (EIA1 and EIA2) and ciphering (EEA0, EEA1 and EEA2).

Features included at the A1AP level include:

- S1-MME Setup/Tear-down.
- Transport of NAS messages.
- Context setup/release procedures.
- Paging procedures.

iii. SPGW

The SPGW entity of the SRS EPC combines the functionality of the S-GW and P-GW components present in LTE networks. More specifically, functionality implementation relating to the S-GW include acting as the main dataplane gateway for users, providing a mobility anchor and gateway for UEs, essentially acting as an IP router and helping to establish GTP sessions between the eNodeB and P-GW. Functionality relating to the P-GW entity includes acting as a point of contact for the EPC with external networks while working to enforce QoS parameters for subscriber sessions. Overall, the SP-GW entity of the SRS EPC provides support for user and control plane communication between the EPC, eNodeB and UE through the S1-U and SGi interfaces.
Features included in the SPGW include [82]:

- SGi interface exposed as a virtual network interface (TUN device).
- Transport of NAS messages.
- SGi to S1-U forwarding using standard compliant GTP-U protocols.
- Support of GTP-C procedures to setup/teardown GTP-U tunnels.
- Support for Downlink Data Notification procedures.

3.4.2 Radio Access Network (RAN)

As explored in preceding chapters, the main component of the LTE RAN is that of the eNodeB which offers subscribers access to a mobile network through radio links. For this project, it is required that a 3GPP compliant eNodeB is implemented. This eNodeB must support communication with the LTE CN through the S1 interface and must support USRP flavoured SDR front-ends to realise a true radio link connection with emulated and COTS UE. As such, the primary implementation of the eNodeB has been realized through a virtualised SRS eNodeB component, while a secondary virtualised OAI eNodeB is implemented in order to evaluate the modular capabilities of the open source tools.

i. eNodeB

As with the SRS EPC, the SRS eNodeB has been designed to run as a 3GPP compliant eNodeB capable of real-time operation through a single binary. Figure 3.4 illustrates the standard compliant internal architecture of the SRS eNodeB entity where all the necessary lower and higher level components have been correctly implemented and stacked. This corresponds with the eNodeB LTE stack described in the 2 Chapter of this report. The manner in which the eNodeB interfaces with the EPC is highlighted in Figure 3.3 whereby an Ethernet connection is made between the EPC and eNodeB components. Control plane signalling occurs over this connection through the S1AP [83] interface using the Stream Control Transmission Protocol (SCTP) [84].

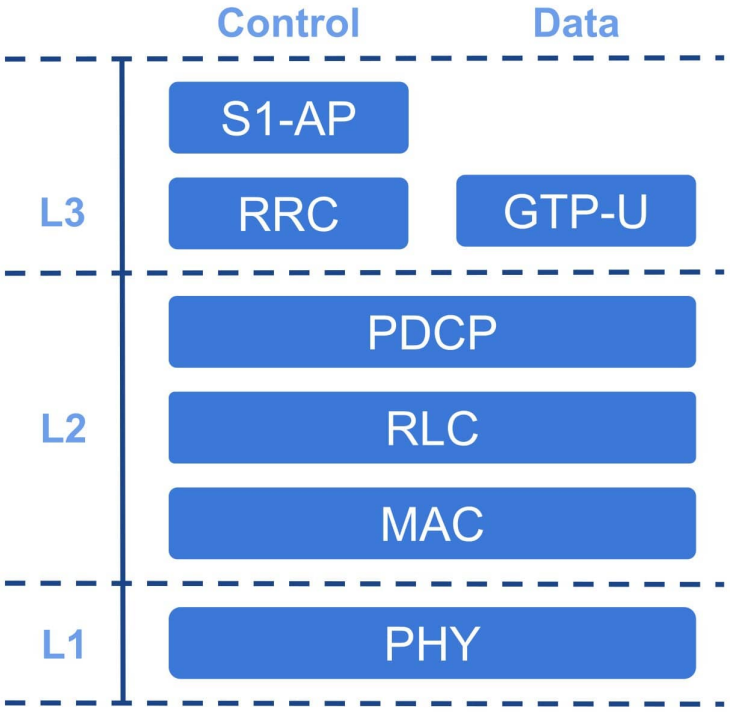


Figure 3.4: SRS eNodeB architecture.
Source: [82]

3.4.3 User Equipment (UE)

As stated, multiple implementations of the UE to be used in this project includes virtualised UE as well as physical COTS UE. Beyond selecting LTE compatible COTS UE, there is not much else that can be designed for or specified. Hence this subsection purely focuses on the emulated UE specifications and design.

i. Emualted UE

As with the eNodeB and EPC, the OAI and SRS UE selected for use as virtualised UE are fully compliant with at least a subsection of the 3GPP Release 8 [22]. This will act to ensure that testing carried out with virtualised UE present results that are as close as possible to real world deployment scenarios with commercial equipment.

ii. SRS UE

Figure 3.5 illustrates the internal protocol stack implemented in the SRS UE, while Figure 3.2 highlights how the UE realises radio link communications with the network. As can be seen in Figure 3.2, the UE's communication with the network is established through the USRP 2944 radio front-end which is connected to the SRS UE through a high performance Peripheral Component Interconnect (PCI) interface. The Radio Resource Control (RRC) layer apparent in Figure 3.5 makes use of the srsLTE API and USRP driver hardware (UHD) [85] for accessing the USRP radio head. As mentioned earlier, baseband signal processing and implementation of the LTE stack remains the responsibility of the SRS UE implemented on a GPP host device. Processed baseband signals and the appropriate metadata is delivered in turn to the USRP SDR through the UHD and srsLTE functions by the RRC whereafter the USRP heterodynes the signal up to the selected carrier frequency.

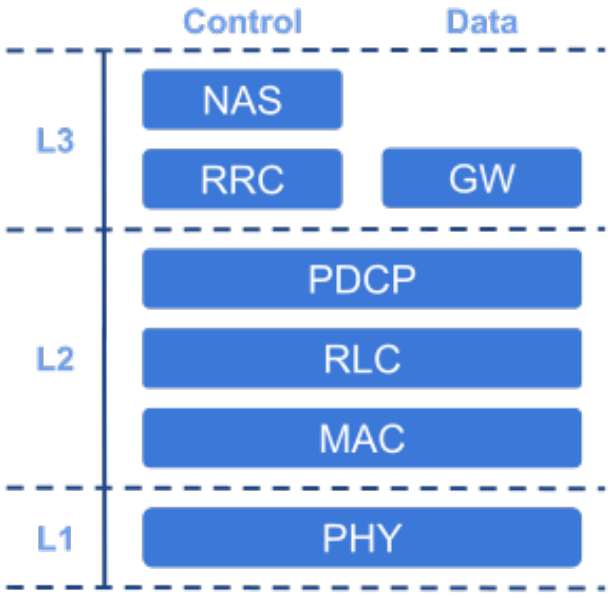


Figure 3.5: SRS UE architecture.
Source: [82]

Once the the UE has successfully attached to the network, a virtual network interface (VNI) is established as a Linux "TUN device". This acts as a tunneling interface that is capable of relaying IP packets between the UE and EPC, enabling applications running on the UE to access the UE gateway and make use of the LTE connection [29].

3.4.4 Coexistence Strategy

The coexistence strategy to be designed must be implemented as either a part of the LTE testbed or as a seperate application built on top of the srsLTE platform. The strategy must employ a duty cycled based discontinuous transmission scheme with a period of 150 ms. Due to the short time-frame for implementation, this algorithm need not include any channel sensing functionality or random back off procedures. Further requirements for the coexistence testbed include the ability to control the duty cycle, the number of PRBs, the transmission power level, and the transmission LTE transmission frequency.

3.5 Subsystem Testing

Subsystem level testing takes place in two phases. The first phase aims to verify correct implementation and functioning of the various subsystems. Following this, baseline resource utilisation experiments are carried out to assess the computational demands of the subsystems.

3.5.1 Connectivity

Connectivity testing at a subsystem level is the first step in ensuring that all the different subsystems are operating as expected and that the network configuration and implementation has been successful. The process of testing connectivity differs slightly between each subsystem but generally involves monitoring the log recordings of each subsystem to make sure that the subsystems are capable of inter-system communication and that the appropriate setup procedures have successfully occurred. Connectivity is then further validated by means of the Linux ping command illustrated in Listing 3.1. This procedure works by sending five consecutive Internet Control Message Protocol (ICMP) echo requests packages to the specific IP address of one subsystem from another. Once the pinged subsystem responds with the ICMP echo replies, we are able to determine the status of connection between the subsystems and record the communication RTT between them. Beyond simple observations of successful intersystem communications, further verification methods include monitoring the network interfaces through packet capturing software such as Wireshark [86]. This is used to verify that the packet sequences correctly match the procedures highlighted in the system logs. Connectivity testing is carried out between each subsystem, with the procedure for each subsystem explored below.

```
$ ping entity_ip_address -c 5
```

Listing 3.1: Linux ping command.

i. CN ↔ eNodeB

Before initiating the startup procedure for the EPC and eNodeB, the first step in ensuring both components are correctly configured is testing whether or not the entities reside within the same network and are visible to one another. This is achieved through the previously described procedure for pinging whereby we will assess connectivity from EPC to eNodeB according to the network configuration described in the Implementation Chapter.

Once all appropriate echo replies have been received, we can proceed with initialisation of the EPC and eNodeB according to the process described in the Implementation Chapter. Following this, monitoring of the console level log output of the components, as well as monitoring of the IP traffic captured by Wireshark is done to assess whether or not successful attachment has occurred. Should the processes complete successfully, we expect to see the appropriate S1 setup requests and response messages output to the console on the EPC side, while the appropriate initialisation request, S1 setup requests and S1 setup response should be observed in Wireshark on both entities.

ii. eNodeB ↔ UE

Once the EPC and eNodeB have been determined to successfully interconnect, the next step in ensuring correct configuration of the network is assessing whether or not UE is capable of attachment to the eNodeB. Keeping in mind that network attachment and authentication procedures are carried out by the EPC, this connectivity test will purely serve as an assessment of whether or not the LTE network is visible to the UE and whether or not the eNodeB is capable of receiving random access requests from the UE.

After the eNodeB has successfully attached to the EPC and is transmitting an LTE signal over a UE supported LTE band, we are able to test visibility of the network by performing a network operator search on the UE. The process for this is provided for various UE types in the Implementation Chapter. Connection between the two devices is considered successful if the network appears visible to the UE, while we observe a random access request in the console level log output of the eNodeB.

iii. CN ↔ UE

The final step in ensuring overall connectivity in the network is assessing the attachment procedure carried out by the EPC after a random access request from the UE has been received by the eNodeB. By monitoring the console level log output at the EPC, we are able to observe that the correct attachment requests have been received and whether or not these requests have been approved or rejected. Should the HSS database and USIM information of the UE have been correctly configured according to the Implementation Chapter, we should observe the EPC's approval of the attachment request and the allocation of a UE IP address on the **172.16.0.x** subnet. Following this, we further assess the UE's attachment by means of pinging the SG-W's IP address of **172.16.0.1** from the UE or pinging the UE assigned IP address from the EPC.

3.5.2 Baseline Resource Utilisation

The second phase of subsystem level testing involves assessing the baseline resource utilisation of the crucial network entities. These entites are namely the eNodeB and EPC as they represent the main load bearing components in the network. For the purpose of this project, resource utilisation statistics will not be measured for the emulated or COTS UE. This is as the emulated UE is run on high performance GPPs provided by the CSIR and would not produce meanigful results in the context of this project. Additionally, resource measurements on COTS UE would prove to be both unnecessarily difficult and pointless due to their dedicated hardware for signal processing. Furthermore, measuring ulitisation for UEs would not provide any useful insight into the performance and capabilities of the network itself.

Resources to be measured for the EPC and eNodeB entities include the random-access memory (RAM) and computational resource (CPU) consumption. Measurements of the consumption of these resources will be done on the respective host machines through the GNOME Resource Monitor. This utility is somewhat limited and constrains testing to a duration of one minute. The benefit offered over alternatives such as the Linux **top** command is that we are able to assess the resource utilisation of individual cores and hyper-threads, as well as SWAP utilisation. In order to maintain meaningful results, all other applications on the host machines are closed and testing only begins after a steady state has been observed. The two scenarios tested will be that of EPC and eNodeB initialisation, whereafter resource utilisation will be observed during the attachment procedure of a single COTS UE.

3.6 System Testing

Once all of the implemented subsystems have been determined to work as expected and baseline performance experiments have been conducted, system level testing can commence. This level of testing seeks to assess the operations and performance of the testbed over the entire LTE stack during emulated UE and COTS UE deployment scenarios. This section details the different test setups and test procedures carried out at the system level to determine throughput capabilities, resource utilisation, scalability, and ease of reconfiguration for coexistence testing.

3.6.1 Throughput Performance

The first step in assessing the performance of the network is to measure data transfer performance between EPC and UE for a variety of different deployment scenarios, network parameters, and data transfer protocols. The main objectives of this performance analysis is to determine the capabilities of open source LTE testbed to achieve the performance goals set by the 3GPP for LTE and to compare the performance against completely virtualised deployments of LTE networks [7]. By generating data through different transfer protocols such as TCP and UDP, we are able to simulate realistic loads being placed on the network for a variety of use cases. More specifically, through TCP transfer rates we are able to asses the networks capacity to handle large amounts of reliable data transfer needed for

transferring files, video streaming services, and textual communications services. UDP on the other hand can be used to asses the networks capacity to handle large loads of high priority and loss tolerant data transfer needed for services such as video calling. Through the assessment of data transfer rates and packet loss statistics, we are able to determine the predicted quality of service for the network. All in all, this will provide valuable insight into the feasibility of over-the-air and virtual deployments for mobile networks development, research, and educational purposes.

In order to asses the throughput capabilities of the network, the **iperf** network performance tool is used to generate both TCP and UDP data streams for a variety of network configurations and parameters. Data is generated in both the uplink direction, whereby the UE device acts as the client and the EPC acts as the server, and downlink direction, whereby the UE device acts as the server and the EPC acts as the client. The **iperf** tool will be used to record packet transfer rates at a specified reporting interval. Furthermore, the transferred packets are captured using Wireshark [86] on the appropriate Ethernet and tunnel interfaces for the duration of traffic generation experiments. By filtering this data according to IP source and destination, we can verify the traffic generation profile.

In the context of this project, the UE is always initialised as a server for the purpose of TCP testing. The command used to initialise the server is shown in Listing 3.2.

```
$ iperf -s -i 1
```

Listing 3.2: TCP server initiation command.

Once the TCP server has been initialised, the EPC is set to run as a client working in reverse mode for one minute according to Listing 3.3 with a reporting interval of one second. In this sense, the EPC first acts as a TCP client for one minute, facilitating data transfer in the downlink direction, whereafter the client and server roles are switched, facilitating data transfer in uplink direction for an additional minute. This is advantageous as it enables bidirectional TCP data transfer statistics to be captured through a single test with all results appearing as CSV formatted output at the EPC console.

```
$ iperf -c ue_ip_address -t 60 -i 1 -y C -r
```

Listing 3.3: TCP client initiation command.

The commands shown in Listings 3.2 and 3.3 have been wrapped into the **tcp_server_init.sh** and **tcp_client_init.sh** shell scripts detailed in the Iperf Flags and Usage Appendix for ease of use during testing. Included in this appendix is a full list and explanation of the various **iperf** flags.

The procedure for generating UDP traffic in the uplink and downlink direction is similar to that of the aforementioned TCP procedure, with slight differences including the need to set the targeted bandwidth and the fact that meaningful output information can only be captured at the server side. The primary reason for this is due to the nature of UDP transfers whereby the client makes no attempt to measure network or achievable throughput conditions. As such, all useful statistics produced by

the UDP transfer will appear on the server side.

```
$ iperf -s -u -i 1 -y C
```

Listing 3.4: UDP server initiation command.

Once the UDP server has been initialised on the appropriate entity according to Listing 3.4, the UDP client can be initiated according to Listing 3.5 on the entity at the opposite end of the LTE stack. This will produce meaningful information such as the achieved bit transfer rate, jitter, packet loss, and the number of out of order packets received at the server over a period of one minute. Following this, the roles of the client and server will switch whereby the same information is generated on what was initially the client over a period of one minute for a dataflow in the opposite direction.

```
$ iperf -c ue_ip_address -u -t 60 -i 1 -y C -r -b specified_bandwidth
```

Listing 3.5: UDP client initiation command.

As before, the commands shown in Listing 3.4 and 3.5 have been wrapped into the shells scripts **udp_server_init.sh** and **udp_client_init.sh** detailed in the Iperf Flags and Usage Appendix.

i. Emulated UE Scenario

The first deployment scenario to be evaluated is that of the full srsLTE stack from virtualised EPC to virtualised UE. Full implementation details of this deployment scenario are described in the Implementation Chapter of this report while an overfiew of the LTE stack is shown in Figure 3.6.

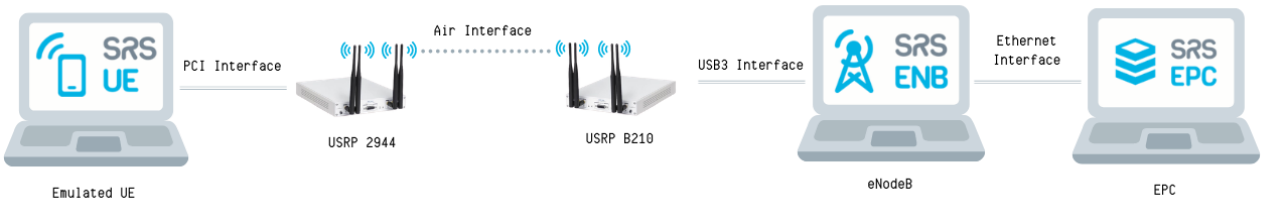


Figure 3.6: Full srsLTE network stack.

UDP and TCP traffic will be generated for this deployment scenario according to the previously described methodology for traffic generation. This traffic will be generated in both the uplink and downlink directions for 25, 50, and 100 resource blocks. The targeted UDP traffic bandwidth set for each resource block configuration is detailed in Table 3.2.

Number of Resource Blocks	Baseband Signal Bandwidth	UDP Uplink Traffic Targeted Bandwidth	UDP Downlink Traffic Targeted Bandwidth
25	5 MHz	5 Mbps	20 Mbps
50	10 MHz	10 Mbps	40 Mbps
100	20 MHz	20 Mbps	80 Mbps

Table 3.2: UDP traffic generation configurations.

ii. COTS UE Scenario

The second deployment scenario to be evaluated is that of a partial srsLTE stack, with network attachments from a single COTS UE as illustrated in Figure 3.7. This scenario serves to validate the capability of the network to handle data requests from commercial equipment, thus confirming the successful deployment of a working LTE network. This performance evaluation will further serve as a means of evaluating the performance as well as comparing the capability of the network with a variety of different COTS UE as listed in Table 3.3.

COTS UE Model	Supported LTE Bands
Samsung S10	1, 2, 3, 4, 5, 7, 8, 12, 13, 17, 18, 19, 20, 25, 26, 28, 32, 38, 39, 40, 41, 66
Samsung S7	1, 2, 3, 4, 5, 7, 8, 12, 13, 17, 18, 19, 20, 25, 26, 28, 38, 39, 40, 41
Samsung J5	1, 3, 5, 7, 8, 20
iPhone 6	1, 2, 3, 4, 5, 7, 8, 12, 13, 17, 18, 19, 20, 25, 26, 27, 28, 29, 30, 38, 39, 40, 41

Table 3.3: List of COTS UE specifications.

Keeping in mind that our objective is to evaluate the compatibility of the testbed with commercial equipment, as well as assess the performance capabilities of the network, performance evaluation with COTS UE will exclusively take place in the downlink direction. This will ensure that scheduling and resource management functionalities of the network are sufficiently stressed and that inadvertent assessment of the UE's capabilities instead of the network's is avoided. To this end, both UDP and TCP traffic will be generated in the downlink direction using the previously described methods for traffic generation. Furthermore, these tests will only be carried out with a network configuration of 100 resource blocks in order to evaluate the absolute maximum transfer rate capability of the network.

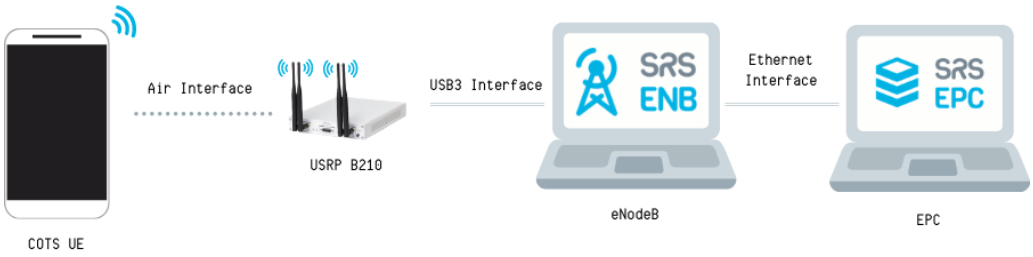


Figure 3.7: COTS UE network stack.

iii. Multiple COTS UE Scenario

The final deployment scenario considered for throughput performance testing is the multiple UE scenario illustrated in Figure 3.8. This aim of this performance test is to maximise the load placed on the network through traffic generation from multiple COTS UE. Due to hardware availability limitations, the number of devices to be attached to the network and tested for simultaneous traffic generation will be limited to the devices listed in Table 3.3. As with the previous singular UE testing, TCP and UDP traffic generation will take place exclusively in the downlink direction with 100 resource blocks available to the network.

Methodology for traffic generation will follow the previously described method, while the procedural order of operations for UE attachment to the network and simultaneous traffic generation is shown below. Steps two to five will take place in one minute intervals, whereafter all devices will be simultaneously receiving data in the downlink direction. Simultaneously reception of data will take place for an additional minute before the experiment is stopped. This process will ensure incremental testing of the network from one UE up to four, and will provide sufficient time between the addition of each UE for throughput rates to stabilise.

Order of Operations for Multiple UE Experiment:

1. Attach all COTS UE to the LTE network.
2. Initiate continual TCP/UDP traffic generation for Samsung S10
3. Initiate continual TCP/UDP traffic generation for Samsung S7
4. Initiate continual TCP/UDP traffic generation for Samsung J5
5. Initiate continual TCP/UDP traffic generation for iPhone 6

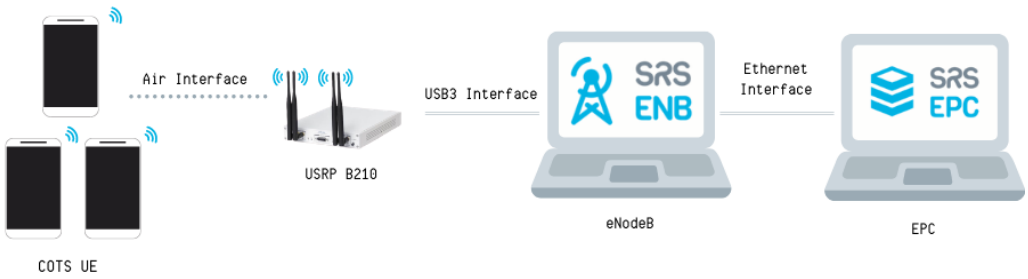


Figure 3.8: Multiple COTS UE network stack.

3.6.2 Resource Utilisation Performance

The next step in assessing the performance of the network is to measure the resource usage of each of the subsystems when placed under load. As before, resource utilisation assessment entails measuring the RAM and CPU consumption and will be done for the EPC and eNodeB components. Resource consumption measurement will follow the methods previously described in Subsection 3.5.2 with the exception that the test will be run from steady state with all network entities initialised and one COTS UE attached, where after a UDP downlink transfer will begin with a targeted bandwidth of 80 Mbps for 30 seconds.

These measurements will help to gain valuable insight into the overall computational efficiency of the system and can be used to assess the networks limitations and scalability, as well determine the capabilities of the deployment compared to completely virtualised deployments.

3.6.3 Coexistence Performance

Performance of the coexistence strategy will be evaluated through the measured impact on co-located WiFi networks. Through this, the viability and fairness of the coexistence strategy can be assessed. Further qualitative findings relating to development time and effort can be made about the ease of further enhancement and development of the open source testbeds to implement such strategies.

The process of evaluating the impact on co-located WiFi networks begins with configuration of a local WiFi network. Following this, two WiFi capable devices are connected to the network and placed at approximately 1 m increments from the WiFi node. The carrier frequency is then identified and used to transmit the periodic LTE signal for varying duty cycles and transmission powers. During each test, UDP traffic with a targeted bandwidth of 80 Mbps will be generated between the two devices using iperf for 60 seconds. The impedence of traffic flow caused by the LTE signal will be measured as a drop in throughput between the two devices.

The duty cycles to be tested range from 0%, as to mimic an absence of the LTE transmission, to 100% to simulate a continuous LTE transmission. Duty cycles will be incremented in steps of 25%. Unfortunately, a spectrum analyser was not available at the time of testing. Hence, the output transmission power cannot accurately be set. Ideally, three different power levels would be used; power levels exceeding the maximum tolerance causing a disassociation of the devices from the network, power levels below the threshold for disassociation but still considered high, and power levels that are not expected to cause interference. Instead, an attempt will be made to achieve these levels though three different transmission gains set through the srsLTE platform; 80, 90, and 100.

Chapter 4

Implementation

This chapter seeks to provide a detailed overview of all the tools and applications used, as well as the process followed in order to realise a fully working over-the-air LTE testbed. Included herein is a description of the tools used for USIM programming, RF spectrum analysis and system performance analysis, details of the physical configuration, as well as the steps required to initialise the various appropriate srsLTE and OAI components. An in depth analysis of the srsLTE and OAI platforms will not be presented here as they are already provided in the Literature Review Chapter of this report.

4.1 Tools and Applications

The various tools and applications used throughout implementation of the over-the-air test bed are discussed in this section. While most of these tools are not directly used to implement the testbed, they play a significant role in both the configuration and performance testing phases. Tools and applications discussed include networking tools, RF spectrum analysis hardware, and SDR frontend hardware directly integrated into the testbed for radio-link communications.

4.1.1 Networking Tools

Various networking tools were used throughout both implementation and results capturing processes. Each of these tools are mentioned and briefly described below, where it is assumed that each tool is used within the Linux environment unless otherwise stated.

i. net-tools

The net-tools package is a Linux tool used for controlling the network subsystem of the Linux kernel and includes functions such as arp, ifconfig, netstat, rarp, nameif and route. For the purposes of this project, the net-tools ifconfig command was used to extract details of the network interface of each device. More explicitly, this was used to correctly configure the network and bind the MME interface of both EPC and eNodeB to the correct address. The net tools package can be insalled using the commands provided in Listing 4.1.

```
$ sudo apt-get update -y
$ sudo apt-get install -y net-tools
```

Listing 4.1: Net-tools installation commands.

ii. iperf

The iperf package is a tool commonly used for active network performance analysis through the generation of standard compliant IP traffic. The iperf tool supports both UDP and TCP transfers and offers both client and server implementations. In a standard use case, traffic is generated by the client and sent to the server. For the purpose of this project, the iperf tool was used to generate traffic in both the uplink and downlink directions for evaluation of throughput capabilities of the LTE testbed. Installation commands for the iperf package are shown in Listing 4.2 while Appendix A offers additional details such as usage examples.

```
$ sudo apt-get update -y
$ sudo apt-get install -y iperf
```

Listing 4.2: Iperf installation commands.

iii. Wireshark

Wireshark [86] is a network protocol analysing tool used for packet capturing. For the purposes of this project, Wireshark was used to monitor traffic flow through the network interface of various components within the testbed. Captured packet data was used through both the implementation and testing phases to determine correct functioning of the network. Installation commands for the Linux flavoured Wireshark application are given in Listing 4.3.

```
$ sudo apt update -y
$ sudo apt install wireshark -y
# select "yes" when prompted
$ sudo usermod -aG wireshark $(whoami)
$ sudo reboot
```

Listing 4.3: Wireshark installation commands.

iv. Wifi Analyser

Wifi Analyser [87] is an Android application used to extract information from currently connected or surrounding WiFi networks on mobile devices. As such, the application was used to determine the carrier frequency of a WiFi network during coexistence testing. This was essential in ensuring full channel overlap between the WiFi and LTE networks.

4.1.2 RF Hardware

Details regarding RF equipment used for LTE transmission and spectrum analysis are provided here.

i. Spectrum Analyser

For the purposes of selecting the best possible downlink frequencies for the LTE transmission, a Rohde and Schwarz FSH4 spectrum analyzer [88], provided by the CSIR, was used to measure the power levels of transmission across various LTE bands.

ii. Software Defined Radios (SDRs)

SDR are essentially reprogrammable radios where physical layer functionality is altered through software repurposing as opposed to hardware reconfiguration. In the context of this project, USRP flavoured SDR devices are used to realise the radio functionality of the eNodeB and emulated UE nodes. The SDR modules implemented in this project, showcased in Figure 4.1, are the Ettus USRP B210 [89] and National Instruments (NI) USRP 2944 [90] provided by the CSIR.

(a) Ettus USRP B210 **(b)** NI USRP 2944

Figure 4.1: USRP SDRs used to implement radio frontends for eNodeB and emulated UE.

While produced by two different suppliers, both SDRs fall under the same USRP device family. Control of the SDRs is realised through their GPP hosts using the USRP Hardware Driver (UHD) software [85]. Consequently, both SDRs are fully compatible and ready for out of the box integration with srsLTE and OAI. Interfacing with the USRP B210 and USRP 2944 is realised though USB3 and the PCI interfaces respectively. Functionality of the two SDRs remain almost entirely identical, with the main differences including their technical specifications and price as showcased in Table 4.1. The USRP is almost triple the real-time bandwidth capacity and is able to produce signals all the way down to 10 MHz. These improved specifications come at an increased price of $10,000, as opposed to the USRP B210's price of roughly $2,500.

SDR	Number of Channels	Maximum Output Power (Pout)	Frequency Range	Frequency Accuracy	Maximum Instantaneous Real-Time Bandwidth
Ettus USRP B210	2	Typically 10 dBm	70 MHz to 6 GHz	± 2 ppm	56 MHz
NI USRP 2944	2	17 dBm to 20 dBm	10 MHz to 6 GHz	± 2.5 ppm	160 MHz

Table 4.1: Comparison of USRP B210 and USRP 2944 technical specifications.

4.1.3 USIM Programming

In order to attach COTS UE to the LTE network, a series of authentication procedures must occur following a random access request whereby information stored within the USIM of the subscriber's UE is matched against subscriber information stored within the HSS. Consequently, programmable USIMs have been used in this project to program specific subscriber information onto four programmable OpenCells USIMs, whereafter the subscriber information is provisioned on the HSS database. Subscriber information is presented in subsequent sections. These USIMs are then inserted into COTS UE (with which they are fully compatible) and used to attach to an appropriate LTE network.

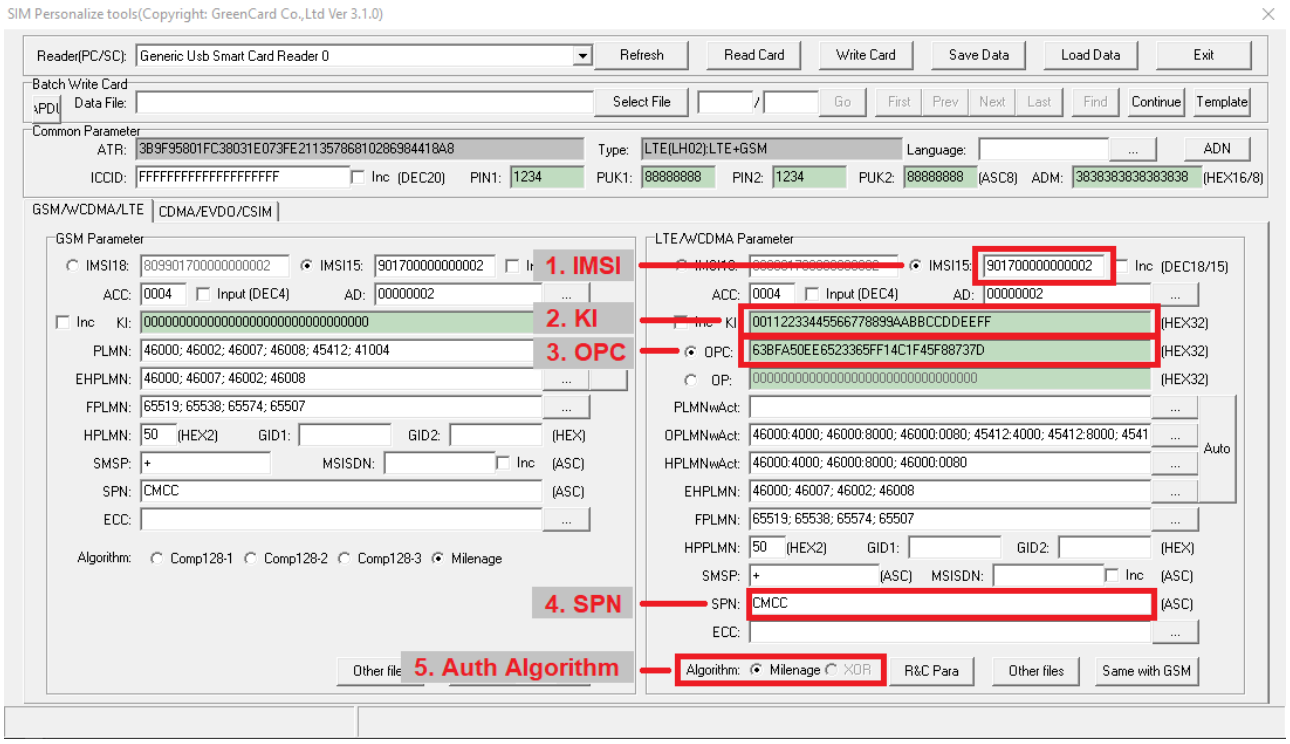


Figure 4.2: Screenshot showing GRSIMWrite application used to program USIMs.

The software and tools used to program these USIMs is that of the GRSIMWrite Application, showcased in Figure 4.2, and the Rocketek Smart Card Reader [91]. The software was provided when purchasing the OpenCells USIMs. As such, the software is a closed source implementation and is quite hard to come by. Advantages offered by GRSIMWrite over alternatives such as PySim include the ability to successfully program the USIM without the correct ADM key.

As shown in Figure 4.2, five main parameters are programmed onto the USIM using GRSIMWrite.

1. International Mobile Subscriber Identity (IMSI)
2. Authentication Key (KI)
3. Derived Operator Code (PC)
4. Service Provider Name (SPN)
5. Authentication Algorithm

4.2 System Setup

This section of the report seeks to detail the procedure followed when implementing the LTE testbed. Included in this is the environment information (i.e host machine specifications), physical testbed setup, as well as the correct procedure for configuring and running various components of the testbed.

4.2.1 Host Machines

The open source LTE testbed platforms have been designed to run in Linux based environments on x64 CPU architectures. As such, each node in the LTE stack is run on a physically separate host machine within a native Linux installation. Due to constraints in the available GPP devices, three differently spec'd host machines have been used. The EPC and eNodeB nodes have been allocated to hosts according to the amount of computational power expected for operation, while the emulated UE is hosted on a desktop machine made available by the CSIR. Specifications of the host machines are listed in Table 4.2.

LTE Node	Machine Description	OS	Central Processing Unit	Graphics Processing Unit	Memory
EPC	Intel® NUC NUC7CJYH	Ubuntu 18.04 LTS	Intel® Celeron® J4005 @ 2.7 GHz x 2	Intel® UHD Graphics 600	4 GiB @ 2400 MHz
eNodeB	Asus FX550V Laptop	Ubuntu 18.04 LTS	Intel® Core® i9-9900 @ 3.10GHz x 8	GeForce GTX 950M 2GiB	8 GiB @ 2133 MHz
Emulated UE	Dell Desktop Machine	Ubuntu 18.04 LTS	Intel® Core® i9-9900 @ 3.10GHz x 8	Intel® UHD Graphics 630	64 GiB @ 2666MHz

Table 4.2: Host machine specifications.

4.2.2 Testbed Setup

i. Emulated UE

The physical testbed setup for the emulated UE scenario is shown in Figure 4.3. As is visible, two different antenna types are used for the LTE signal transmission. More explicitly, log-periodic antennas are used for the eNodeB while omnidirectional antennas are used for the UE. This specific combination of antennas is used for reasons other than the fact that they were the only antennas on hand. The specific configuration of antennas is selected to more closely mimic a real world LTE deployment where eNodeBs contain high power RF equipment and UE transmission capabilities are relatively lacking. This is realised through the fact that the directional nature of the log-periodic antennas will result in higher transmission power. Furthermore, the antennas have been placed 1.5 m apart, with the USRP B210 and USRP 2944 connected to the eNodeB and emulated UE.

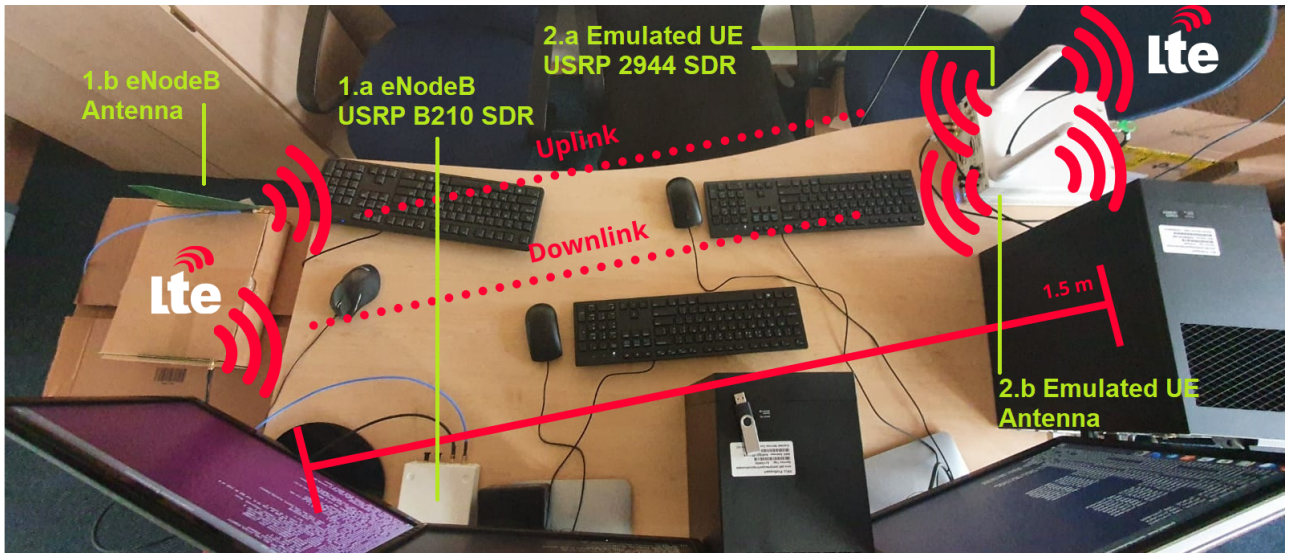


Figure 4.3: Labelled image showing physical testbed setup.

ii. COTS UE

The physical testbed setup for the COTS UE scenario remains similar to the previous setup for the emulated UE. The main differences between the two include the use of log-periodic antenna for eNodeB transmissions over the omnidirectional antennas, and removal of the emulated UE entity. Instead, COTS UEs are placed as needed in the location previously occupied by the emulated UE.

4.3 Testbed Implementation Procedure

The following section will give a detailed account of the procedure taken when implementing various components of the LTE network. Implementation of both OAI and srsLTE based components will be given for the eNodeB and emulated UE, while the EPC will remain solely implemented using srsLTE. It is assumed from this point onward that srsLTE and OAI platforms have been correctly installed and that the host system environment has been correctly configured according to the detailed instructions provided in Appendix B, as well as the Low Latency Linux Kernel required for OAI.

4.3.1 EPC Initialisation

i. Configuration

Prior to initialising the SRS EPC, a number of configuration parameters need to be set. The location of the configuration file needed for amending is made apparent during the EPC initialisation, but is generally found at the `$HOME/srslte/.config/srslte/epc.conf` file location. In order to correctly configure the EPC with the host's network, the correct network configuration needs to be set. More explicitly, the `mme_bind_addr` and `gtpu_bind_addr` need to be set to the network interface address of the host device. This can be unidentified using the `ifconfig` command. These paramters ensure that the MME successfully binds to the interface and correctly listen for S1-MME connections.

The interface address of the host machine used in the context of this project has been identified as 146.64.216.186. Other configuration parameters that should be set include the Tracking Area Code (TAC), Mobile Country Code (MCC) and Mobile Network Code(MNC). The combination of MCC and MNC produce a unique identification number for the network and are needed to ensure successful attachment of the eNodeB. The TAC, MCC and MNC selected for use in the project are 17,655 and 25 respectively. This matches the MCC value allocated to South Africa and represent a vacant MNC not in use by mobile operators. A full list of the parameters set is shown in Listing 4.4.

```
mme_bind_addr = 146.64.216.186
gtpu_bind_addr = 146.64.216.186
tac = 0x0017
mcc = 665
mnc = 25
```

Listing 4.4: EPC configuration parameters set during prior to initialisation.

ii. User Provisioning

Following configuration of the EPC, users are provisioned onto the HSS database by means of editing the csv file found at the $HOME/srslte/.config/srslte/user_db.csv file location. Subscriber information is stored here in the following format:

```
(ueName),(algo),(imsi),(K),(OP/OPc_type),(OP/OPc_value),(AMF),(SQN),(QCI)
    ,(IP_alloc)
```

For the purposes of this project, the following parameters are set as standard values for each of the provisioned users:

- Authentication algorithm (algo) = MILENAGE (mil)
- Authentication key (k) = 00112233445566778899aabbccddeeff
- OP/OPc_type = opc
- OP/OPc_value = 63bfa50ee6523365ff14c1f45f88737d
- AMF = 9000
- SQN = 000000000000
- QCI = 9
- IP_alloc = dynamic

Unique parameters provisioned for each of the four users include the ueName and imsi fields. These are illustrated below with in the format "ueName,imsi".

- ue1,901700000000001
- ue2,901700000000002
- ue3,901700000000050
- ue4,901700000000051

An example of a full user provisioning is provided in Listing 4.5.

```
ue1,mil,901700000000001,00112233445566778899aabbccddeeff,opc,63
    bfa50ee6523365ff14c1f45f88737d,9000,0000000009ef,9,dynamic
```

Listing 4.5: Example showing user provisioning in the HSS database.

iii. EPC Initialisation

Following configuration and user provisioning, the EPC is ready to be initialised. This can be done through the commands presented in Listing 4.6. This should result in successful initialisation of the HSS, MME, and SP-GW components. The EPC is now ready to receive S1 setup requests.

```
$ cd ~/srsLTE/build/srsepc/src
$ sudo ./srsepc
```

Listing 4.6: Commands for EPC initialisation.

4.3.2 eNodeB Initialisation

i. Configuration

Following initialisation of the EPC, the eNodeB should be made ready for attachment. This is done through configuration of the eNodeB network interface, as well as correctly setting the MCC and MNC parameters. The procedure for this is slightly different between the SRS and OAI implementations of the eNodeB. While the SRS eNodeB is simply able to make use of the configuration file located at `$HOME/srslte/.config/srslte/enb.conf`, OAI requires the configuration file to be specified at run-time.

Parameters that need to be set in the SRS eNodeB include the mcc, mnc, mme_addr, gtp_bind_addr and s1c_bind_addrparameters. Here the MCC and MNC values should match those previously set for the EPC, while the MME address should match the network interface previously identified for the EPC. Once again, the GTP and S1 addresses need to be specified as the eNodeB'd host network interface address to ensure that the eNodeB is successfully able to bind to the host device's network. This address can be found using the `ifconfig` command and has been identified as `146.64.19.227` in this project. A detailed list of the SRS eNodeB configuration parameters is given in Listing 4.7

```
mcc = 655
mnc = 25
mme_addr = 146.64.216.186
gtp_bind_addr = 192.168.1.62
s1c_bind_addr = 192.168.1.62
```

Listing 4.7: Configuration parameters set prior to srsLTE eNodeB initialisation.

Similar parameters need to be set for the OAI eNodeB. An example configuration file that can be used is found at the $HOME/openairinterface5g/ci-scripts/conf_files/enb.band7.tm1.25PRB.usrpb210.conf file location. An aditional paremeter that needs to be set by the OAI eNodeB is the eNodeB interface name. Once again this can be found using the ifconfig command. A full list of the parameters set in this project is found in Listing 4.8.

```
...
tracking_area_code = 17;
plmn_list = ( { mcc = 655; mnc = 25; mnc_length = 2; } );
..


...
mme_ip_address       = ( { ipv4       = "146.64.216.186";
                           ipv6       = "192:168:30::17";
                           active     = "yes";
                           preference = "ipv4";
                         }
                       );
...


...
    NETWORK_INTERFACES :
    {
        ENB_INTERFACE_NAME_FOR_S1_MME            = "eno1";
        ENB_IPV4_ADDRESS_FOR_S1_MME              = "192.168.1.62";
        ENB_INTERFACE_NAME_FOR_S1U               = "eno1";
        ENB_IPV4_ADDRESS_FOR_S1U                 = "192.168.1.62";
        ENB_PORT_FOR_S1U                         = 2152; # Spec 2152
...
```

Listing 4.8: Configuration parameters set prior to OAI eNodeB initialisation.

ii. Initialisation

Following successful configuration of the eNodeB components, it is posible to initialise and attach the eNodeB to the SRS EPC. While theoretically both eNodeBs should be able to simualtenously connect to the EPC (provided they have been configured with unique eNodeB IDs) through the X2 interface, for the purpose of this project only a single eNodeB will be attached at any given point in time. As the primary LTE platform selected for this project, it is assumed from this point forward that the implemented eNodeB is that of the srsLTE testbed platform unless otherwise stated.

Initialisation of the SRS eNodeB can be done using the commands provided in Listing 4.9. As is seen, the flags rf.dl_earfcn and enb.n_prb are set at run-time. These flags are responsible for setting the downlink LTE frequency as well as the number of PRBs (i.e the LTE signal bandwidth). These are set according to the configuration scenario described in the Methodology Chapter.

```
$ cd srsLTE/build/srsenb/src
$ sudo ./srsenb --rf.dl_earfcn=2750 --enb.n_prb=100
```

Listing 4.9: Commands for SRS eNodeB initialisation.

Initialisation of the OAI eNodeB can be done using the commands provided in Listing 4.10. Unlike the SRS eNodeB, the downlink frequency and number of RBs are configured through the configuration file specified by the -O flag. These were not changed for the purposes of this project. Further run-time parameters found in Listing 4.10 are responsible for setting the RRC innactivity level thresholds and for logging to the ENB.log file.

```
$ cd ~/openairinterface5g/cmake_targets
$ sudo -E ./lte_build_oai/build/lte-softmodem -O ~/openairinterface5g/ci-
    scripts/conf_files/enb.band7.tm1.25PRB.usrpb210.conf --eNBs.[0].
    rrc_inactivity_threshold 0 2>&1 | tee ENB.log
```

Listing 4.10: Commands for OAI eNodeB initialisation.

4.3.3 UE Initialisation

Once the EPC and eNodeB have been successfully initialised, the network should be ready to accept attachment requests from UE. This section provides some detail on how to attach both SRS and OAI emulated UE, as well as provide details for attaching COTS UE.

i. Emulated UE

Attaching the SRS UE is a simple procedure. Both the Access Point Name (APN) and USIM information have been correctly preconfigured (provided one has not removed any of the preprovisioned users from the HSS database) during the srsLTE installation. The only parameter that needs configuring is that of the downlink frequency. This can be configured through the rf.dl_earfcn flag at run time as illustrated in Listing 4.11. Following initialisation, the SRS UE will automatically begin a cell search around the specified frequency and attempt to attach to nearby LTE networks.

```
$ cd srsLTE/build/srsue/src
$ sudo ./srsue --rf.dl_earfcn=2750
```

Listing 4.11: Commands for SRS UE initialisation.

Similarly, the downlink frequency of the OAI UE can be set at run-time using the -C flag as shown in Listing 4.12. Unlike the SRS UE, the details of the virtual USIM have not been preprovisioned by the SRS EPC. As OAI does not provide an easy way of reconfiguring the USIM parameters, the default USIM parameters of the OAI UE will have to be provisioned onto the SRS EPC using the methods discussed in earlier sections. These parameters will be printed to the console during the installation process of the OAI UE. Once the OAI UE has been initialised, it will automatically start a cell search to identify and attempt attachment with nearby networks.

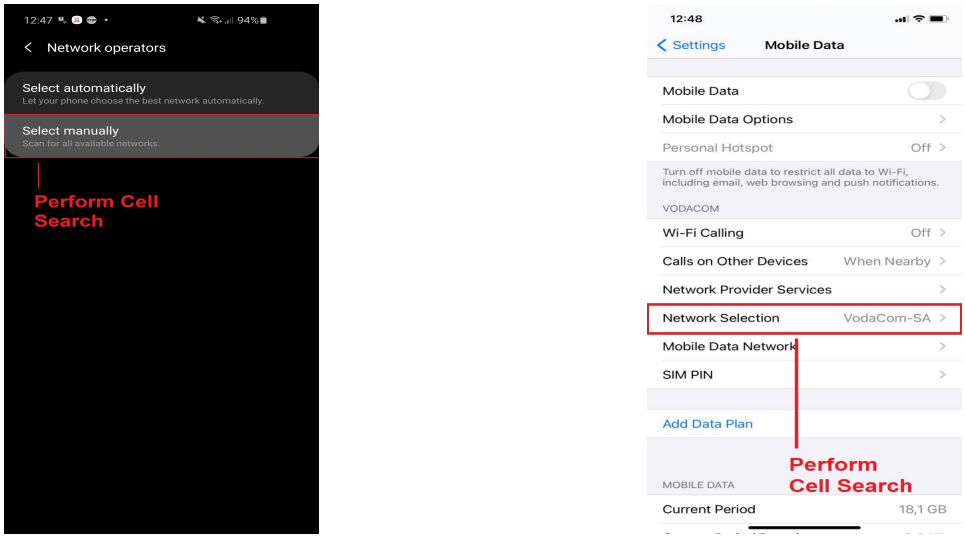
```
$ ~/openairinterface5g/cmake_targets/lte_build_oai/build
$ sudo ./lte-uesoftmodem -C 2620000000 -25 --ue-scan-carrier 2>&1 | tee
  UE.log
```

Listing 4.12: Commands for OAI UE initialisation.

ii. COTS UE

The first step in attaching COTS UE to the network is ensuring that the physical USIMs have been programmed correctly and that the subscriber information has been successfully provisioned on the HSS database. Both USIM programming and subscriber provisioning have been discussed in previous sections. Following this, the physical USIM can be inserted into the COTS UE.

Following the insertion of the physical USIM, the next step in ensuring UE attachment is the correct setup of the APN. This is necessary as the APN is used to identify the P-GW through which packet routing must take place. APN settings can be configured on both Android and iOS devices according to the guide provided in Appendix C. Cell searches can then be carried out through the options highlighted in Figure 4.4. Prior to initial attachment, the network should appear as "66525". Following attachment, the network automatically renames to either "Software Radio Systems LTE" or "CMCC".



(a) Android UE **(b)** iOS UE

Figure 4.4: Screenshots showing options for cell search on COTS UE.

4.4 Coexistence Strategy

4.4.1 Physical Setup

The physical setup used for testing of the coexistence strategy was done according to the setup described in Methodology Chapter. As such, use was made of a local network and router to propagate a WiFi signal. This was done at a time where no users were present to ensure consistent results. WiFi capable devices were then placed at 1m increments from the WiFi node and connected to the network. These devices were namely a Samsung S10 and the Intel NUC used throughout the LTE testbed deployment. The log-periodic previously used was again placed another meter from the Intel NUC and connected to the USRP b210. The USRP was interfaced with the previously used Asus laptop which acted as the GPP host.

4.4.2 Algorithm

Implementation of the rudimentary coexistence algorithm is achieved through modification of the example code provided in the `examples/pdsch_enb.c` location and is similar to the one presented by Gomez-Miguelezis [13]. In this implementation, transmission gain is set to zero during transmission off time while the duty cycle is set through the `-d` flag at run-time. Modifications excluding the additional flag functionality are shown in Listing 4.13. It is worth noting that the accuracy of the clock cannot be held to an extremely high standard but is sufficient for the needs of this project.

Figure 4.5 illustrates the different flags used when running the enhanced code. Through these flags the desired duty cycle, transmission gain, transmission frequency and number of PRBs are selected for testing according to the Methodology. Consequently, the frequency selected must match that of the local WiFi network configured. Using the WiFi Analyser Android application, the local WiFi network's centre frequency was determined to be 2642 MHz. The application was also used to determine that there were no other WiFi networks present in the space that would potentially affect the outcome of the results.

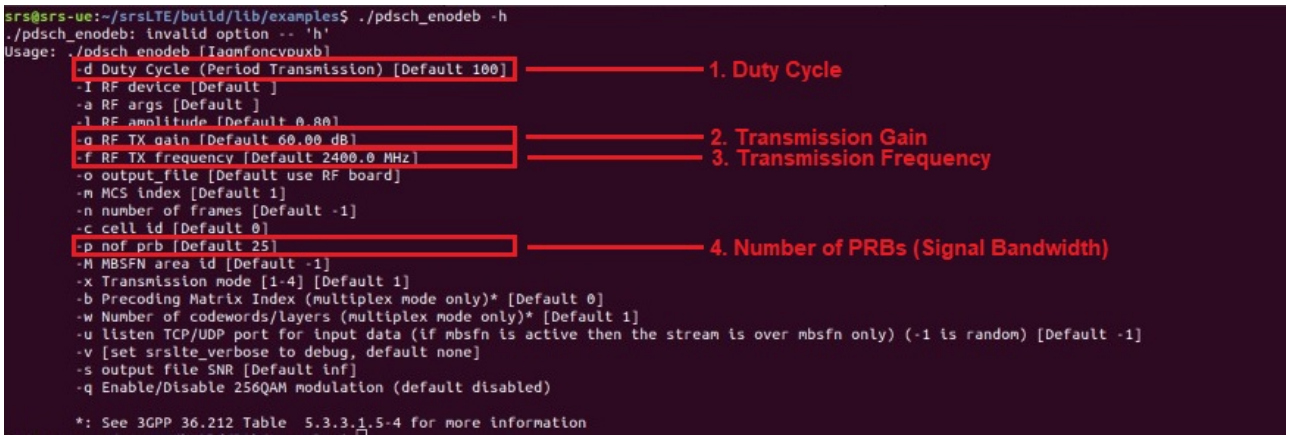


Figure 4.5: Screenshot showing run-time flags for program to test coexistence.

```
#include <time.h>
#define duty_cycle_default 100
clock_t start, end;
double cpu_time_used;
...
float duty_cycle =duty_cycle_default; # later set by run-time flag
...

// Reset duty cycled transmission
if (cpu_time_used > 150){
  srslte_rf_set_tx_gain(&radio, rf_gain); # Set gain to originally
     specified value
  start = clock();                        # Restart the clock
}
// Set transmission gain to null during off period
if (cpu_time_used > 150 * (duty_cycle/100) && cpu_time_used<150){
  srslte_rf_set_tx_gain(&radio, 0);       # Set gain to zero
}

// Send baseband signal to USRP for transmission
srslte_rf_send_multi(&radio, (void**)output_buffer, sf_n_samples, true,
   start_of_burst, false);

// Update the clock
end = clock();
cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC *1000;
```

Listing 4.13: Modifications made to the pdsch_end.c example code to realise a duty cycle based LTE transmission scheme.

Chapter 5

Results

5.1 Subsystem Testing

As described in Section 3.5 of the Methodology Chapter, subsystem level testing took place in two phases. The first phase aimed to verify the correct implementation of the LTE testbed while the second phase was used to record baseline resource utilisation statistics of the EPC and eNodeB components. The results of these phases are presented below.

5.1.1 Connectivity

Following initialisation of the SRS EPC, the various subcomponents described in Subsection 3.4.1 of the Methodology Chapter should start in a particular order as described below. During initialisation of the various components, the MME and SPGW entities should bind to the network address as specified by the **mme_bind_addr** and **gtpu_bind_addr** parameters in the EPC configuration file. The initialisation procedure is monitored through console level log outputs and is shown in Figure 5.1.

EPC Initialisation Procedure:

1. HSS initialisation.
2. MME initialisation.
3. SP-GW initialisation.

```
---  Software Radio Systems EPC  ---

Reading configuration file /home/srslte/.config/srslte/epc.conf...
HSS Initialized.
MME S11 Initialized
MME GTP-C Initialized
MME Initialized. MCC: 0xf655, MNC: 0xff25
SPGW GTP-U Initialized.
SPGW S11 Initialized.
```

Figure 5.1: Screenshot of Console Output Showing Successful SRS EPC Initialisation

The lack of error messages present in Figure 5.1 suggests that the SRS EPC initialisation has completed successfully for the HSS, MME and SPGW components, while outputs showing MCC and MNC values of 655 and 25 correspond with the configuration parameters set in the Implementation Chapter. Furthermore, we can observe the successful initialisation of the GPRS Tunneling Protocol (seen as GTP-C) [92]. Thus, the EPC is working as expected and should be ready for eNodeB attachment.

i. CN ↔ eNodeB

As mentioned in Subsection 3.5.1 of the Methodology Chapter, the first step in ensuring successful inter-system connection between EPC and eNodeB is making sure that both components reside within the same network. This was achieved through the Linux ping command whereby 5 ICMP echo requests were sent from eNodeB to EPC. Some of the packets captured are shown in Figure 5.2 where it can be seen that the SRS EPC has successfully replied to the eNodeB. This confirms that both components reside within the same network and are ready for attachment.

No.	Time	Source	Destination	Protocol	Length	Info
97	5.075480298	146.64.19.227	146.64.216.186	ICMP	98	Echo (ping) request id=0x3452, seq=1/256, ttl=63 (reply in 98)
98	5.075553020	146.64.216.186	146.64.19.227	ICMP	98	Echo (ping) reply id=0x3452, seq=1/256, ttl=64 (request in 97)
179	6.094307633	146.64.19.227	146.64.216.186	ICMP	98	Echo (ping) request id=0x3452, seq=2/512, ttl=63 (reply in 180)
180	6.094380000	146.64.216.186	146.64.19.227	ICMP	98	Echo (ping) reply id=0x3452, seq=2/512, ttl=64 (request in 179)
189	7.118305715	146.64.19.227	146.64.216.186	ICMP	98	Echo (ping) request id=0x3452, seq=3/768, ttl=63 (reply in 190)
190	7.118373731	146.64.216.186	146.64.19.227	ICMP	98	Echo (ping) reply id=0x3452, seq=3/768, ttl=64 (request in 189)
217	8.142314999	146.64.19.227	146.64.216.186	ICMP	98	Echo (ping) request id=0x3452, seq=4/1024, ttl=63 (reply in 218)
218	8.142384145	146.64.216.186	146.64.19.227	ICMP	98	Echo (ping) reply id=0x3452, seq=4/1024, ttl=64 (request in 217)

Figure 5.2: Wireshark packet capture showing SRS EPC and SRS eNodeB components residing on the same local area network.

After initialisation of the SRS eNodeB, attachment to the EPC should occur by means of an S1 setup request. This request is observed by monitoring console level log output of the EPC, as well as monitoring the packets captured at the network interface of either component. Figure 5.3 shows that the S1 setup request has successfully been delivered to the SRS EPC while the eNB id, MCC and MNC values correctly correspond with the parameters configured in the Implementation Chapter. The S1 setup response indicates that the eNodeB attachment has been successful. This is corroborated by the packets captured on the eNodeB network interface, highlighted in Figure 5.4, which shows the successful S1 setup procedure following an initialisation message from eNodeB to EPC.

```
Received S1 Setup Request.
S1 Setup Request - eNB Name: srsenb01, eNB id: 0x19b
S1 Setup Request - MCC:655, MNC:25, PLMN: 5698898
S1 Setup Request - TAC 0, B-PLMN 0
S1 Setup Request - Paging DRX v128
Sending S1 Setup Response
```

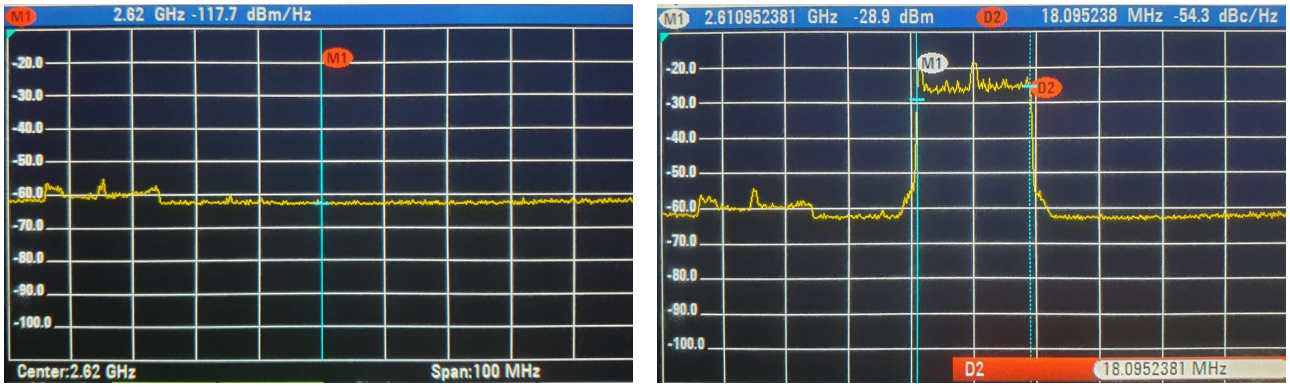
Figure 5.3: Screenshot of console output showing successful S1 setup request and response.

No.	Time	Source	Destination	Protocol	Length	Info
585	22.140358272	146.64.19.227	146.64.216.186	SCTP	82	INIT
586	22.141087026	146.64.216.186	146.64.19.227	SCTP	306	INIT_ACK
587	22.141100522	146.64.19.227	146.64.216.186	SCTP	278	COOKIE_ECHO
588	22.141823890	146.64.216.186	146.64.19.227	SCTP	60	COOKIE_ACK
589	22.141854466	146.64.19.227	146.64.216.186	S1AP	114	S1SetupRequest
590	22.142491615	146.64.216.186	146.64.19.227	SCTP	62	SACK
591	22.143431659	146.64.216.186	146.64.19.227	S1AP	106	S1SetupResponse
592	22.143437953	146.64.19.227	146.64.216.186	SCTP	62	SACK

Figure 5.4: Wireshark packet capture showing successful S1 setup request and response.

Before attempting to attach UE to the network, we need to observe that the eNodeB is successfully transmitting an LTE signal. This is done through the use of a spectrum analyser whereby the spectral power levels over the appropriate frequency range is monitored before and after eNodeB initialisation. Figure 5.5a shows an absence of transmissions over a centre frequency of 2620 MHz

prior to eNodeB initialisation, while Figure 5.5b shows the presence of an 20 MHz transmission following eNodeB initialisation. This is in line with the configured number of resource blocks and the EARFCN for downlink transmission, proving that the eNodeB successfully transmits an LTE downlink signal following its attachment to the EPC.



(a) Prior to eNodeB Initialisation **(b)** Post eNodeB Initialisation

Figure 5.5: Screenshot of spectrum analyser showing 20 MHz LTE transmission located at center frequency of 2620 MHz.

ii. eNodeB ↔ UE

After the LTE testbed has been successfully configured and initialised, an assessment of the visibility of the network to both emulated UE as well as COTS UE by means of performing a cell search is done. Following this, the UE's ability to communicate with the eNodeB by checking whether or not the eNobeB receives Random Access Preamble (RAP) from the UE during an attempt at attachment is assessed. This can be monitored through the eNodeB's console level log output which should show activity on the Random Access Control Channel (RACH) as illustrated in Figure 5.6. The methodology for performing both cell searches and network attachment is presented in Subsection 4.3.3 of the Implementation Chapter.

```
==== eNodeB started ===
Type <t> to view trace
RACH:  tti=8821, preamble=32, offset=1, temp_crnti=0x46
```

Figure 5.6: Screenshot of console output showing RAC activity on the eNodeB.

The recorded visibility and RAP results are tabulated and presented in Table 5.1. The fact that the network presents itself as visible to all of the implemented UE is testament to the testbed's LTE transmission being fully LTE 3GPP compliant. This is as the UE is able to synchronise with the transmission using the Primary (PSS) and Secondary (SSS) Synchronization symbols, whereafter it is able to successfully decode the Master Information Block (MIB). Similarly, the eNodeB's ability to decode the RAP sent by UE serves to prove that its RACH processing chain is fully compliant.

	SRS UE	OAI UE	Samsung S10	Samsung S7	Samsung J5	iPhone 6
Network Visibility	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
RA Preamble Recieved	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)

(✓) Visible\RAP Received.
(x) Not Visible\Rap Not Received..

Table 5.1: Network visibility across various UE implementations.

iii. CN ↔ UE

Following confirmation that the LTE testbed is fully operational and compliant, experimentation between the the CN and UE can proceed. While these tests focus solely on attachment procedures at the CN and UE level, they represent testing of the full end to end LTE stack. As such, each UE was first singularly attached to the network to assess compatibility with the testbed. Attachment procedures were monitored through console level log output on the EPC, eNodeB and virtualised UE, while attachment was monitored through the signal status for COTS UE. Following this, multiple COTS UE were attached to the network simulatenously. As before, this was monitored through the console level log output on the EPC, eNodeB, as well as the UE device's signal status.

Emulated UE

Figure 5.9 shows a screenshot of the console output at the EPC side following an attachment request from the emulated SRS UE. The screenshot shows that the IMSI received corresponds to the IMSI paremter set during configuration and that an IP address has successfully been allocated to the UE. Finally, it is observed that the CN sends EPS Mobility Management (EMM) information to the UE which includes information such as the network name and local time [93]. This concludes and represents a successful network attachment.

Figure 5.7 shows a screenshot of the simple console output at the eNodeB side following the attachment request from the emulated SRS UE whereby a Temporary Cell Radio Network Identifier (C-RNTI) is allocated to the UE. A more detailed console output is provided at the SRS UE end and is highlighted in Figure 5.8. It is apparent that the MCC and MNC values of the network do not match those of the configured USIM. This is a non-issue as the UE is able to attach regardless and is considered to be "roaming". Also apparent in this screenshot is the allocation of an IP address to the UE. This corresponds to the IP address allocated in Figure 5.9. The entire attachment procedure can be corroborated by monitoring traffic at network interface at the EPC. Through this we can confirm a successful attachment to the network, as well as determine time taken for attachment. For the case of SRS UE attachment the time taken to attach is **313.7 ms** as shown in Figure 5.10.

```
==== eNodeB started ===
Type <t> to view trace
RACH:  tti=8821, preamble=32, offset=1, temp_crnti=0x46
User 0x46 connected
```

Figure 5.7: SRS eNodeB console output showing successful SRS UE network attachment.

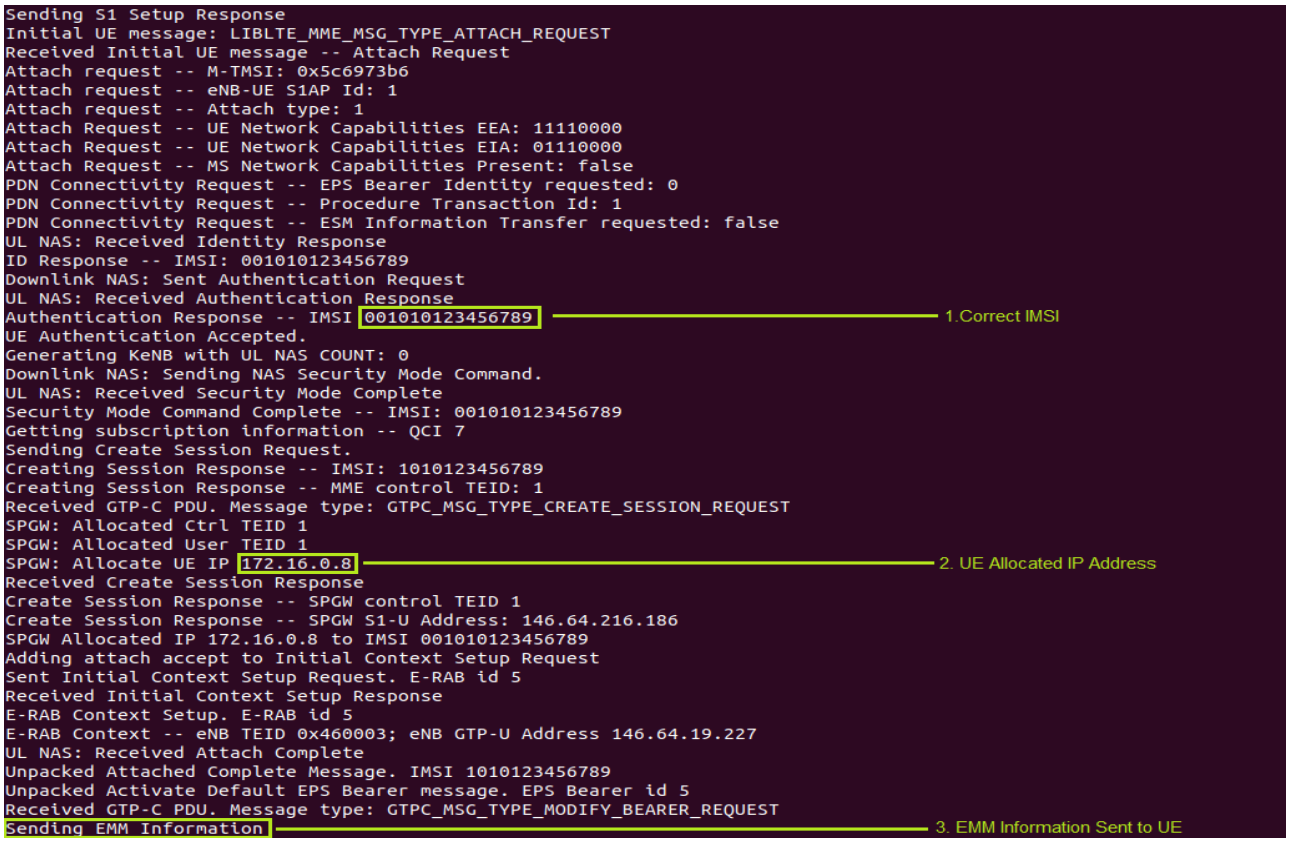


Figure 5.8: EPC console log output showing successful SRS UE attachment procedure.

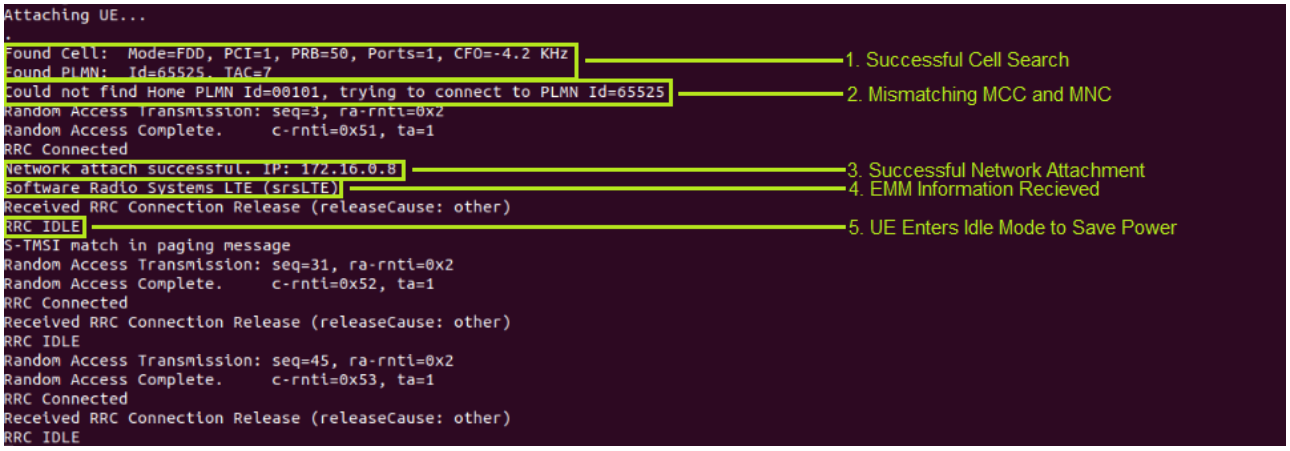


Figure 5.9: SRS UE console output showing successful network attachment.

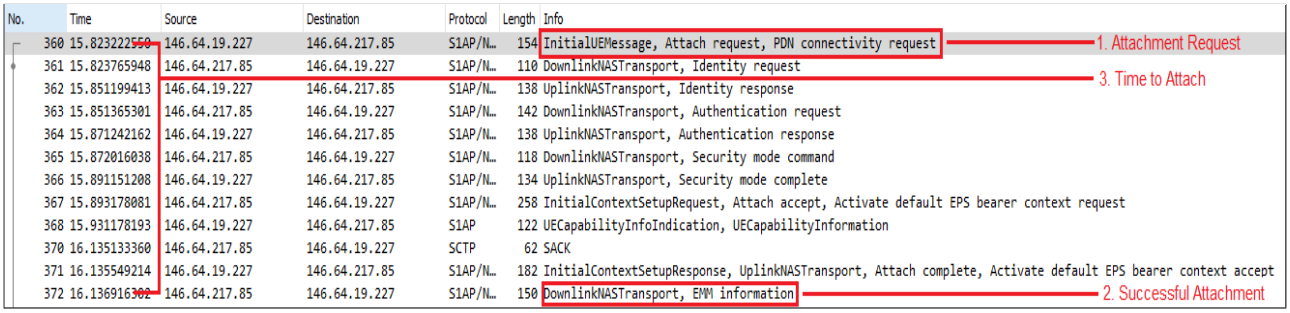
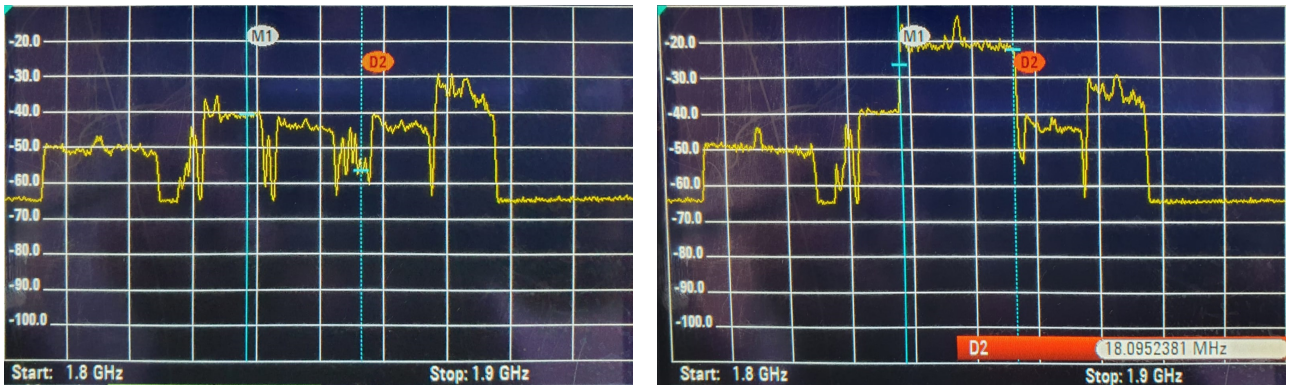


Figure 5.10: Wireshark Packet Capture Showing SRS UE Attachment Procedure

COTS UE

The same procedure was carried out for testing singular COTS UE attachment to the network. Unique USIM information was programmed onto 4 programmable USIMS and added to the HSS database according to the methods outlined in the Implementation Chapter. In practice it was challenging to find unused spectrum in an LTE band supported by all COTS UE devices (a full list of compatible LTE bands for each UE is given by Table 3.3). It was ultimately decided to transmit the LTE signal on band 3 with a downlink frequency of 1847.5 MHz as shown in Figure 5.11. While other transmission signals were present in this space, the testbed LTE transmissions were run at a high enough power to render these signals null. Furthermore, the directional nature of the antenna and short duration of testing ensured minimal disruption to external networks. Table 5.2 summarises the attachment time of each COTS UE with the results of the Linux ping test used to verify successful attachment.



(a) Prior to eNodeB initialisation. **(b)** PosteNodeB initialisation.

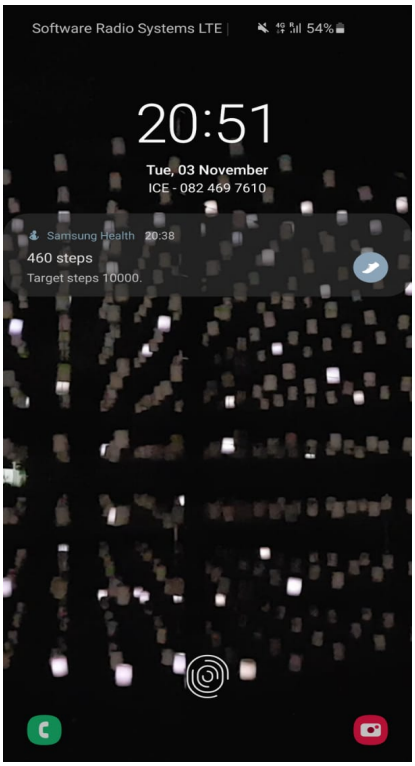
Figure 5.11: Screenshot of spectrum analyser showing 20 MHz LTE transmission located at centre frequency of 1847.5 MHz.

Following singular COTS UE attachment, an attempt was made at attaching all four UEs to the network. This was successful as all four UEs were able to attach to the network simultaneously without any visible connection loss over a five minute period. A screenshot of each UE's connection status is presented in Figure 5.12 where the "Software Radio Systems LTE — CMCC" is visible in the top left corner of each device.

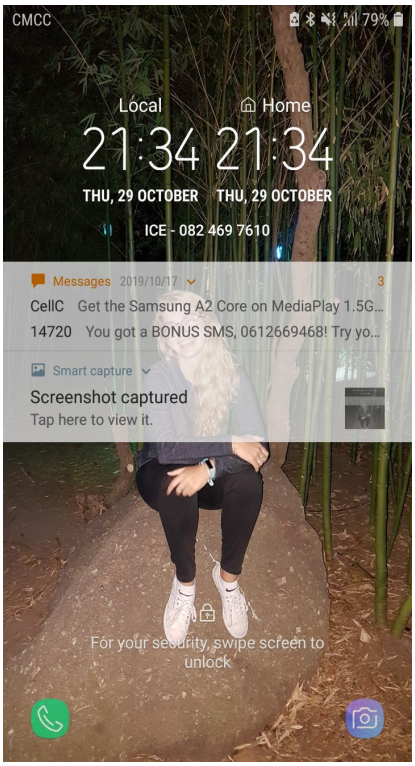
UE	Time to Attach (ms)	Ping Reply
Samsung S10	533.87 ms	(✓)
Samsung S7	582.44 ms	(✓)
Samsung J5	571.35 ms	(✓)
iPhone 6	502.86 ms	(✓)

(✓) Successful reply.
(-) Unsuccessful reply.

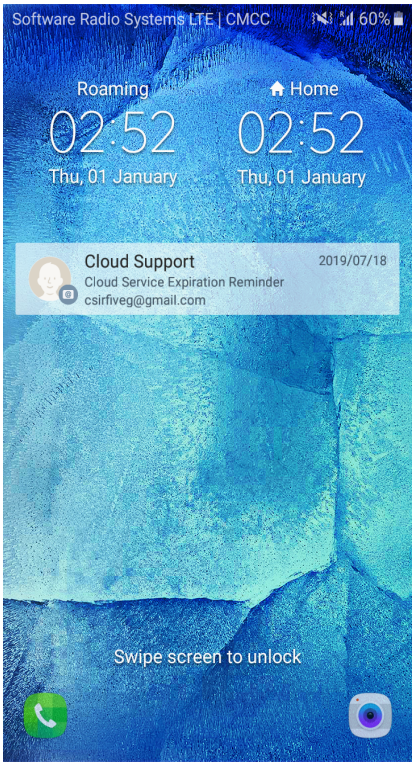
Table 5.2: COTS attachment time and linux ping reply results.



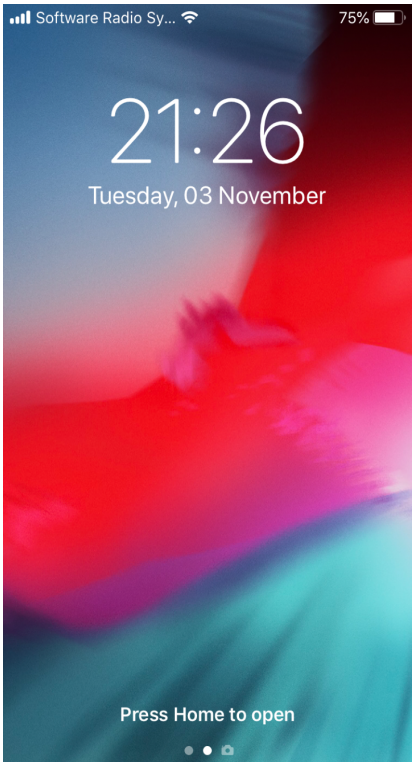
(a) Samsung S10



(b) Samsung S7



(c) Samsung J10

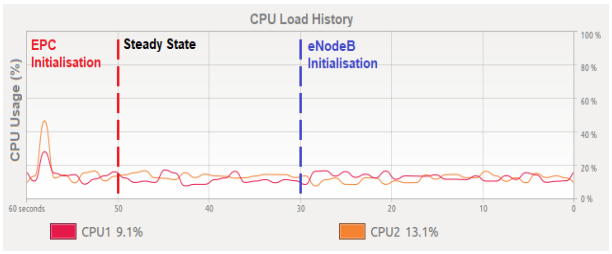


(d) iPhone 6

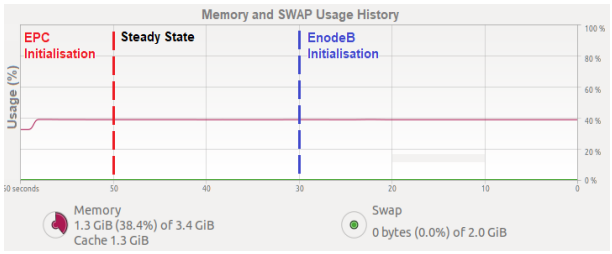
Figure 5.12: Simultaneous COTS UE attachment to software radio systems LTE.

5.1.2 Baseline Resource Utilisation

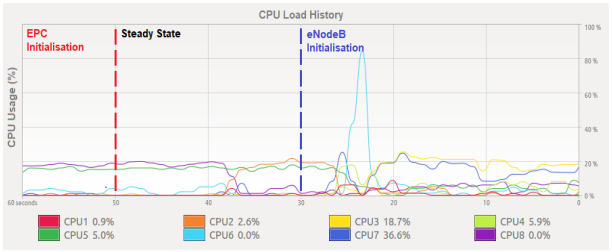
Modified screenshots of the Gnome Resource Monitor presented in Figure 5.13 showing baseline resource utilisation of the EPC and eNdodeB during component initialisation without UE attachment. From this set of results, it is clear that the eNodeB is much more computationally expensive than the EPC. While Figures 5.13a and 5.13b show a memory and CPU consumption increase of 200 MB and 32% during EPC initialisation, Figure 5.13c and 5.13d show increase of 1.1 GB and 63% during eNodeB initialisation. While CPU usage of the eNodeB is mostly recovered to previous levels following initialisation, this does not occur for the RAM usage. Also apparent from these results is the fact that eNodeB attachment has an almost entirely insignificant effect on the resource utilisation of the EPC. This is as there is no discernible increase in RAM usage and only a slight momentary increase of 10% usage of one of the CPU threads.



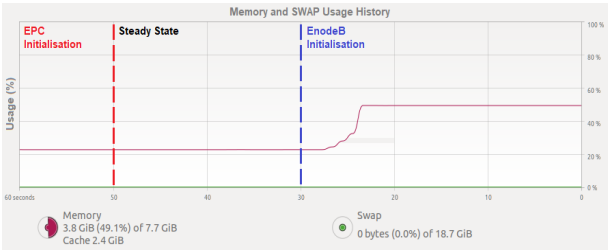
(a) EPC CPU



(b) EPC RAM



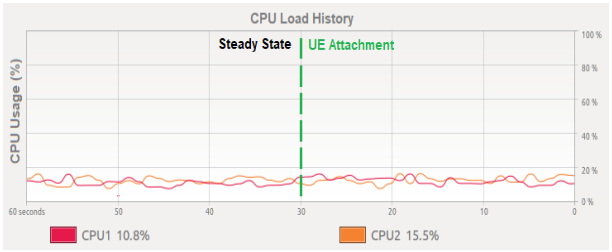
(c) eNodeB CPU



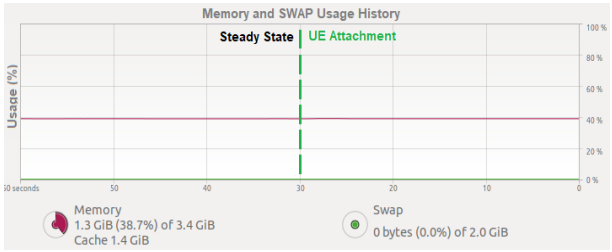
(d) eNodeB RAM

Figure 5.13: Baseline resource usage during EPC and eNodeB initialisation.

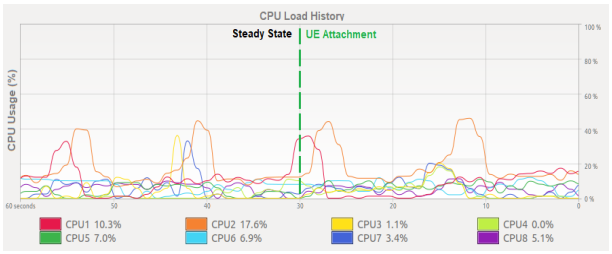
Following initialisation, the EPC and eNodeB components were monitored for a period of time until their resource usage history showed a somewhat steady or repetitive output. The utilisation results generated after attaching a Samsung S10 device to the network are presented in Figure 5.14. Once again, an indiscernible effect was observed on the EPC during UE attachment with no increase in RAM usage and a change in CPU usage that cannot firmly be correlated to the eNodeB attachment. On the other hand, the eNodeB saw a slight RAM usage increase of 100 MB during UE attachment, indicating the allocation of memory resources needed to perform UE resource management functions. Surprisingly, no clear change in the eNodeB CPU usage was observed during this period.



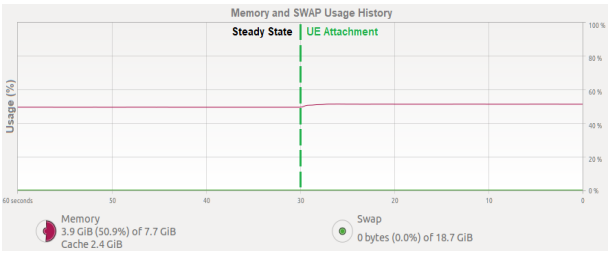
(a) EPC CPU



(b) EPC RAM



(c) eNodeB CPU



(d) eNodeB RAM

Figure 5.14: Baseline resource usage during singular COTS UE attachment.

5.2 Heterogeneous Deployment

Heterogeneous deployment in the context of this project is defined as the deployment of an LTE stack with at least two of the components implemented being developed by two different groups or vendors. While the OAI platform officially supports multi-configuration scenarios for heterogeneous deployment [59], the same cannot be said for the srsLTE. Although the literature presents many examples of heterogeneous deployments in the context of srsLTE [29], it is not apparent whether or not this is an "out of the box" feature. To this end, various heterogeneous deployments have been tested for UE attachment to two differently implemented CNs through OAI and srsLTE based eNodeBs. The analysis of UE attachment capabilities for the various configurations follows the same methodologies used for the full srsLTE stack.

Table 5.3 presents a simplified list of results for the various heterogeneous deployments tested. As expected, the OAI eNodeB is able to successfully attach to the SRS EPC. This serves to prove that the SRS EPC is indeed 3GPP compliant provides out of the box compatibility with alterative eNodeB implementations. Somewhat surprising, both the SRS eNodeB and OAI eNodeB were capable of out of the box attachment to Fraunhofer CN implementation [1]. This indicates the compatibility of OAI and srsLTE with more commercial solutions.

[1]The Fraunhofer CN [94] (known as Open5GCore) is a closed source implementation of the 3GPP 5G core network. While the Open5GCore does not fall under the scope of this project, preliminary testing was conducted with the CN as a bonus to assess compatibility of the LTE testbed with future 5G technologies. The license holder for the particular Open5GCore used for these preliminary tests is the CSIR [80].

EPC	eNodeB	UE
SRS EPC	SRS eNodeB (✓)	SRS UE (✓)
		OAI UE (x)
		COTS UE (✓)
SRS EPC	OAI eNodeB (✓)	SRS UE (-)
		OAI UE (x)
		COTS UE (-)
Fraunhofer	SRS eNodeB (✓)	SRS UE (-)
		OAI UE (x)
		COTS UE (x)
Fraunhofer	OAI eNodeB (✓)	SRS UE (-)
		OAI UE (x)
		COTS UE (x)

(✓) Working.
(x) Not Working.
(-) Not Tested.

Table 5.3: Heterogeneous architectures deployment results.

Interestingly, the only fully working LTE stack was that of an end to end srsLTE implementation. Attempts by the OAI UE to attach to the SRS eNodeB looked promising as a successful cell search would take place. Unfortunately, the OAI UE would lose synchronisation during the random access procedure and would fail to attach to the network. The same was true when attempting to attach the OAI UE to an OAI eNodeB, leading to the belief that there was a misconfiguration of the OAI UE. Failed attempts at attaching to the Fraunhofer core can be narrowed down to rejection in the authentication procedure caused by absence of an SQN [95] parameter on the USIM. This is a fundamental limitation of the USIMs aquired for this project.

5.3 System Testing

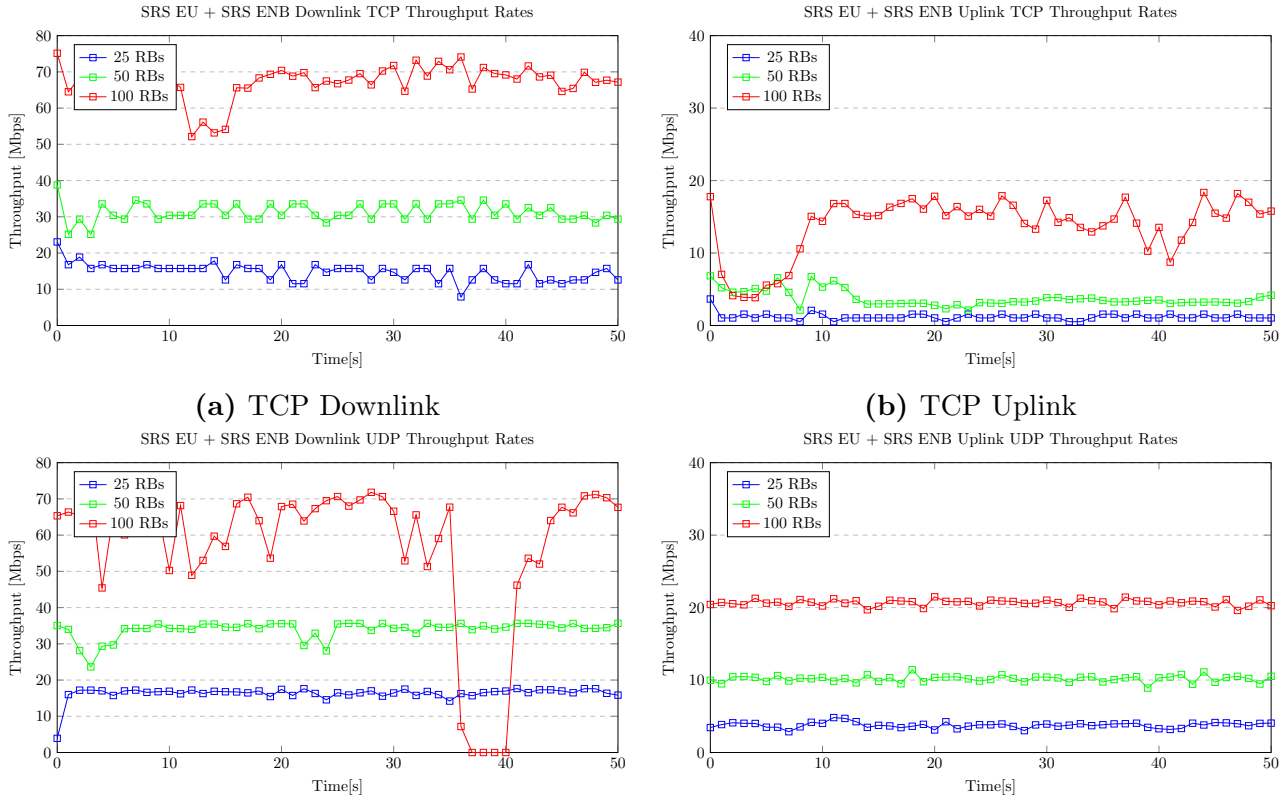
Following a thorough assessment of subsystem functionality and capabilities, system level testing of the entire LTE stack commenced. This testing served to determine the capability of the open source srsLTE framework to act as an over-the-air LTE testbed. System level testing follows on from subsystem level testing with results in this section including throughput performance, resource utilisation under load, as well as an analysis of the developmental and performance results of a simplistic coexistence strategy.

5.3.1 Throughput Performance

As discussed earlier in the Methodology Chapter, both TCP and UDP data streams are generated for a number of different network scenarios and parameters. The first scenario involving an emulated srsLTE UE underwent the most extensive experimentation with UDP traffic being generated in both the uplink and downlink directions with network parameters of 25, 50 and 100 RBs. Experimentation with COTS UEs proved to be a little less extensive with UDP and TCP traffic only being generated in the downlink direction for 100 RBs. An attempt was made to investigate the throughput for multiple UE attachments, but this ultimately proved unsuccessful.

i. Emulated UE

Figure 5.15 shows the throughput results for the emulated UE scenario. As expected, there is a close to doubling of throughput for a doubling of RBs throughout all network configurations and traffic types. Also as expected, uplink throughput rates were consistently lower than downlink throughput rates. This is in line with the SC-FDMA modulation scheme that is used in the uplink direction compared to the OFDMA modulation scheme used for downlink. The peak throughput rate achieved was 75 Mbps during a TCP downlink transfer with 100 PRBs. This is in slight contrast to the expected maximum throughput occurring during a UDP transfer. As such it must be noted that this rate was only achieved momentarily and that all transfers using 100 PRBs were significantly less stable when compared to lower PRB deployments. Overall, UDP throughput rates in both the uplink and downlink directions were greater than their TCP counterparts. This was expected as the TCP protocol lowers the targeted bandwidth using network quality indicators and feedback.



(a) TCP Downlink

(b) TCP Uplink

(c) UDP Downlink

(d) UDP Uplink

Figure 5.15: SRS UE throughput results.

UDP transfer statistics recorded during the emulated UE scenario, including jitter and packet loss, are presented in Tables 5.4 and 5.5. While the stable packet loss of arund 20% in the uplink direction might indicate a poor RF environment, the decreasing jitter for increasing PRBs indicates this is not the case. Instead, the decreasing jitter shows a successful increase in the capacity of the network while the high packet loss was probably caused by targeting bandwidths (detailed in Table 3.2) that exceed the uplink capacity [96].

RB	Average Jitter (ms)	Average Packet Loss (%)
25	4.86	27.06
50	2.36	2.59
100	1.40	1.42

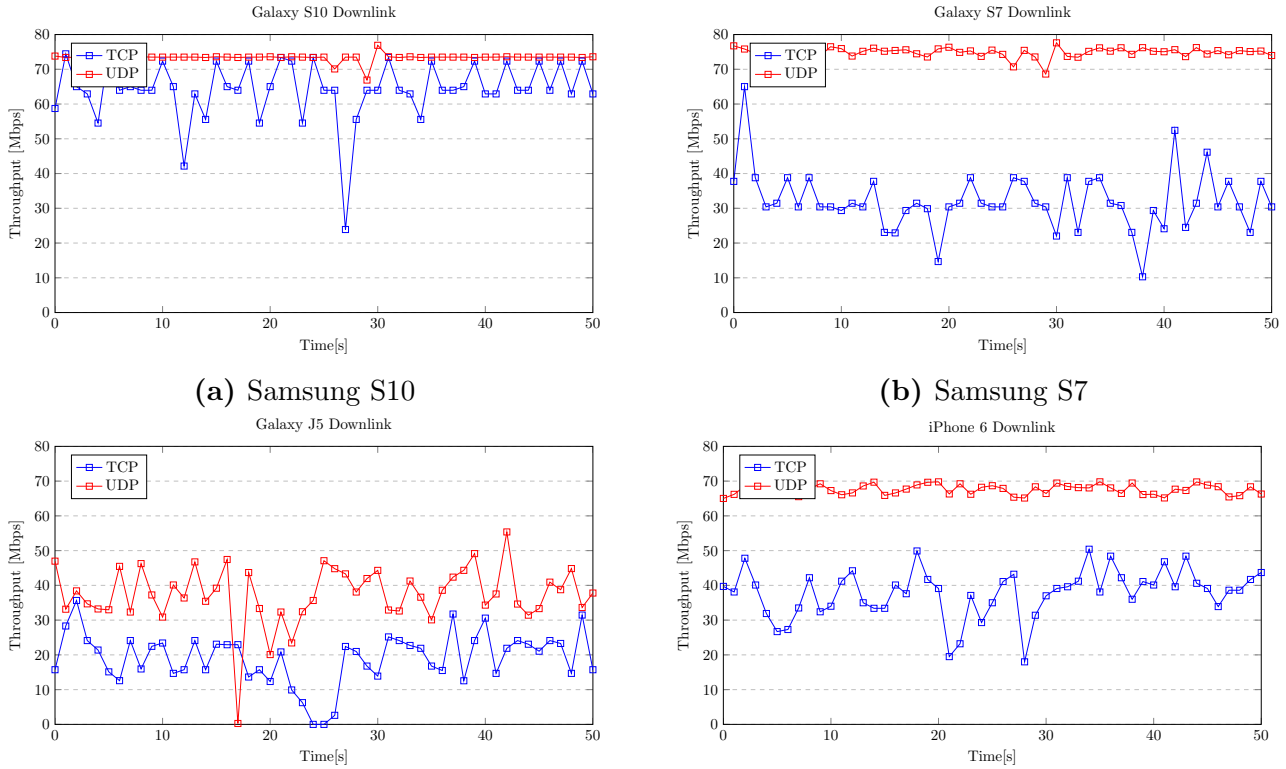
Table 5.4: SRS UE UDP downlink statistics.

RB	Average Jitter (ms)	Average Packet Loss (%)
25	1.00	23.16
50	0.95	20.53
100	0.78	25.27

Table 5.5: SRS UE UDP uplink statistics.

ii. COTS UE

As discussed in Subsection 3.6.1, throughput testing using COTS UE took place purely in the downlink direction for 100 PRBs. Figure 5.16 shows the throughput results during singularly attached COTS UE testing. As with the emulated UE, UDP results show higher throughput rates compared to their TCP counterparts. Unlike the emulated UE, performance between TCP and UDP differed considerably, with a difference as high as 40 Mbps for the Samsung S7. Furthermore, these differences varied significantly between the different devices, with the Samsung S10 having the most consistent result with only a 15 Mbps difference. While this leads to the belief that performance testing for COTS UE was heavily influenced by the COTS UE hardware, IP transfers indicate a successful LTE deployment and a compatibility with commercial equipment.



(a) Samsung S10

(b) Samsung S7

(c) Samsung J5

(d) iPhone 6

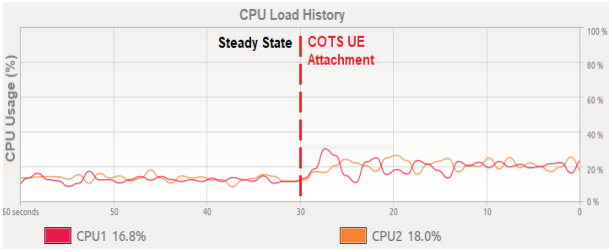
Figure 5.16: COTS UE throughput results.

Following throughput testing with singularly attached COTS UE, an attempt was made to measure the throughput with multiple COTS UE according to the method described in Subsection 3.6.1. These efforts proved futile as while the network supported multiple UE attachment, the devices would simply disconnect when two or more devices attempted to transfer IP traffic simultaneously.

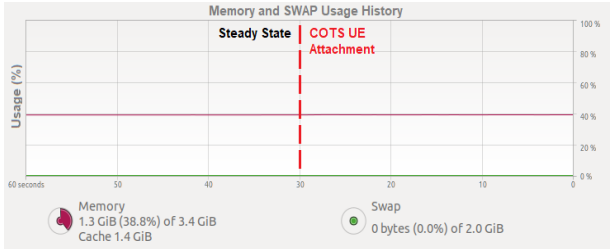
5.3.2 Resource Utilisation

While resource utilisation testing over the entire LTE stack aimed to take place under the most computationally expensive deployment scenario, the inability of the network to handle traffic for more than one user simultaneously rendered this impossible. System level resource utilisation testing was revised to a singularly attached COTS UE with UDP data being transmitted in the downlink direction with a targeted bandwidth of 80 Mbps. Methodology for testing remained consistent with procedures detailed in Subsection 3.6.2, with tests only commencing once both EPC and eNobeB were considered to be in "steady state".

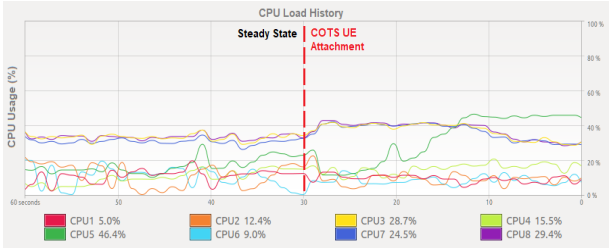
Results of this testing, shown in Figure 5.18, reveal that no additional memory is allocated on the EPC and eNodeB once traffic generation is initialised. This is expected as the memory needed for resource management had already been allocated during UE attachment. While there is an increase in EPC CPU activity during this time, this is thought to be a combination of both the core network and **iperf** server activity. A more consequential increase in CPU usage is seen by the eNodeB where an additional thread is allocated for computation and overall CPU usage of existing threads are increased by 10%. This suggests a somewhat computationally expensive signal processing chain.



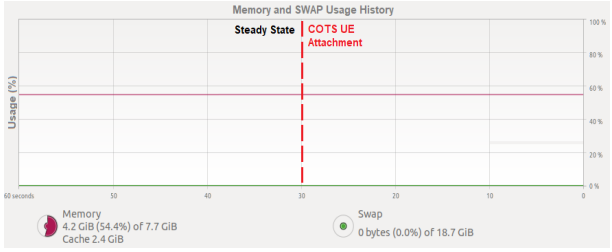
(a) EPC CPU



(b) EPC RAM



(c) eNodeB CPU



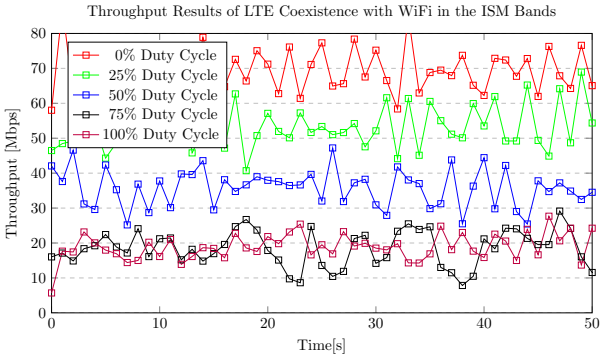
(d) eNodeB RAM

Figure 5.17: Resource usage during singular COTS traffic generation.

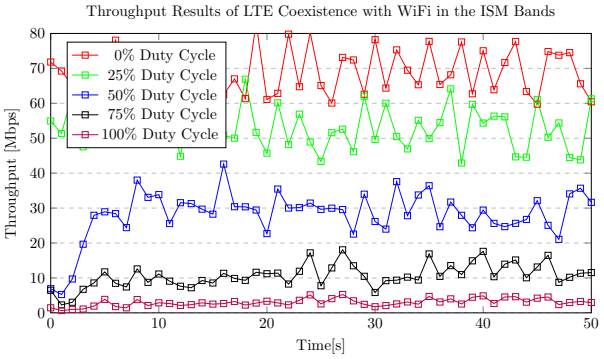
5.4 Coexistence Strategy

The final batch of testing that occurred was that of the performance impacts of a primitive coexistence strategy implemented using the srsLTE platform. The results of this experimentation include a qualitative assessment of the performance impact on co-located WiFi networks, as well as a qualitative assessment on the difficulty of expanding on the srsLTE platform.

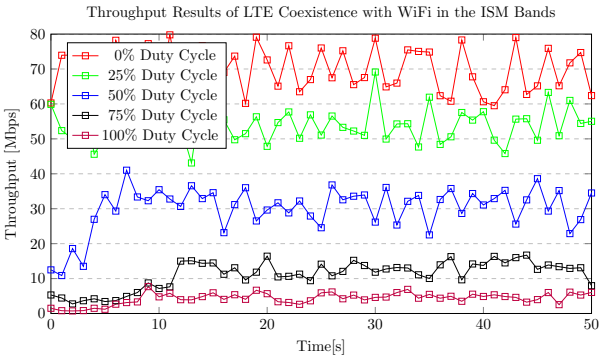
Figure 5.18 shows the resulting impact on co-located WiFi networks when a periodic LTE signal is transmitted over the same frequency for various different LTE power levels. Keeping in mind that a 0% duty cycle in essence means the absence of an LTE signal, this is used as the baseline WiFi network performance metric against which we can compare the performance of different duty cycles and power levels.



(a) Transmission gain of 80.



(b) Transmission gain of 90.



(c) Transmission gain of 100.

Figure 5.18: Impact duty cycled LTE transmissions on co-located WiFi networks.

One observation that is immediately apparent from these results is a mostly inversely proportional relationship between WiFi throughout and LTE duty cycle. As such the worst WiFi throughput results measured across the board occurs when there is no duty cycling present i.e the 100% duty cycle translates to standard LTE transmission, while the best occurs when there is no LTE signal present. In other words, as the transmission time of the LTE signal is decreased, the WiFi throughput is increased. It is also true that the performance of the WiFi network seems to scale almost linearly with the decrease in the LTE duty cycle in most instances. This serves as proof that a duty cycled LTE

transmission reduces interference impacts on co-located WiFi networks when compared to standard LTE transmissions. Unfortunately, none of the test results show the ability of the WiFi network to remain operating at its baseline performance level when a periodic LTE signal is present. This is not indicative of a fair strategy for coexistence.

As predicted, the WiFi network is starved of spectrum resources by the LTE signal during continual and high power LTE transmission. This can be seen in Figure 5.18b and 5.18c. Unexpectedly, the power level of the LTE signal played a less significant role as WiFi throughput across the board was similar for same duty cycled LTE transmissions. The exception to this is shown in Figure 5.18a for the lowest LTE transmission power where the performance of both the 75% and 100% duty cycled LTE transmissions were quite similar, and much greater than that of their counterparts in the more powerful LTE transmissions. This suggests suggests that a transmission gain selection of 80 produced a signal whose power levels were just shy of the threshold for WiFi ED functionality.

Chapter 6

Discussion

This section of the report seeks to outline some of the quantitative measurements presented in the Results Chapter, as well as to discuss some of the qualitative findings made during the process of implementing an over-the-air LTE testbed and primitive coexistence strategy. These discussions will draw on the knowledge base produced in the Literature Review Chapter of this report, and will speak to the project objectives and problem statement laid out in the Introduction. Discussions will be split into four separate sections; connectivity, throughput performance, resource utilisation, and coexistence.

6.1 Connectivity

The main purpose of connectivity testing was to ensure that the various nodes withing the LTE stack had been configured and implemented correctly, as well as to verify the compliance of the open source implementations and architectures with the 3GPP protocols.

6.1.1 Connection Between EPC and eNodeB

As shown through extensive testing in the Results Chapter, both srsLTE eNodeB and EPC were able to successfully attach to one another though the appropriate interfaces. This was verified through consol level log output, as well as through packet capture using Wireshark. This confirms the systems capability of running as a distributed deployment amongst various physical machines. This is in line with the 3GPP standards whereby multiple eNodeBs are able to connect to a separate EPC entity through IP protocols.

While Majola [7] points out that the combination of both S-GW and P-GW functionalities into a single binary by OAI is not in line with LTE specifications, the same implementation scheme is seen in SRS EPC architectures and is in fact in line with common implementations known as SAE gateways [29, 30]. One true limitation of the SRS EPC is that all functionality has been combined into a single executable. While this makes the system lightweight and highly efficient, this somewhat limits network configuration options. This prevents further analysis of the individual EPC components, and prevents a distributed deployment that is more likely in a real world scenario. This makes the system more prone to a single point of failure, whereby a component malfunction or even unauthorised access could result in loss of the entire core network. Due to the architectural design of the SRS EPC, these components could be separated into individual entities, but would significant development resources.

A more in depth analysis of the packets captured during the eNodeB attachment process, illustrated in Figure 5.4, finds that the S1 setup procedure and SCTP communication protocols used are not specific to the srsLTE platform, and are instead fully compliant with the 3GPP standard. Consequent to this, the srsLTE EPC should be fully compatible with 3rd party, or even commercial, eNodeB implementations. This idea was confirmed through results presented in the 5.2 Chapter. Here it was verified that the OAI eNodeB implementation was able to successfully connect to the SRS EPC. Likewise, both SRS and OAI implementations of the eNodeB were even able to connect to to a commercial implementation of a 5G core known as the Fraunhofer core.

In terms of the process of configuring system parameters to enable attachment between eNodeB and EPC, the srsLTE platform presents the simplest and most seamless means of reconfiguration. While srsLTE automatically parses configuration files produced during installation, the OAI requires configuration files to be specified at runtime. All in all, this presents an opportunity for quick and easy configuration of an srsLTE based network. This could hold value in an educational context where there is limited contact time between educators and students. On the other hand, the more time consuming configuration of OAI offers slightly more out of the box control over the network. This might be more appropriate in a research context where time is comparatively a less limiting factor.

6.1.2 Attachment of Emulated UE

Unlike the virtualised RAN presented by Kwezi Majola [7], emulated UE attachment by the SRS EU to the SRS eNodeB takes place in realtime as to mimic the operations of a real LTE deployment. This is true for both the completely virtualised SRS UE which was tested during preliminary investigations, as well as the over-the-air implementation presented in Subsection 3.5.1. While it is not apparent if OAI's completely virtualised RAN now supports real-time operations, through testing presented in the Heterogeneous Deployment Section, it is clear that OAI has a somewhat successful implementation of a real-time emulated UE capable of over-the-air deployment.

The SRS eNodeB's successful decoding of RAP transmitted by the SRS UE shows working LTE communications over the air interface between UE and eNodeB, while successful RAP decoding from OAI UE transmissions highlight the SRS eNodeB's protocol compliance and compatibility with 3rd party platforms. Furthermore, an analysis of the successful attachment procedure of the SRS UE following RAP decoding shows an emulated attachment procedure that is again in line with specifications. These emualtaed UE are quite powerful as they provide complete control over the full LTE stack implemented. While platforms such as OAI provide the ability to simulate RF conditions for completely virtualised deployments, this is not necessarily truly representative of a real world deployment. As such, limitations of the emulated UE for true LTE deployment include the need for SDR front-ends for signal transmission and reception. This might prove too costly and limiting for in educational context or under resourced research department.

6.1.3 Attachment of COTS UE

All results with COTS UE testing successfully showed attachment to the deployed LTE testbed. The combination of iOS and Android UEs across devices of varying ages showed full compatibility and compliance with the 3GPP standards. Initial attachment time across all UE devices was around 500 ms after performing cell searches. This result is reasonable and would be in line with subscriber expectations. Following intial attachment, the COTS UE devices would retain information of the network. Consequently, UEs were even able to reattach to the network before boot procedures had completed following a restart of the UE. A minor limitation with regards to COTS UE is the equipment and time needed for USIM programming, while a more significant limitation is the fact that these UE remain completely unconfigurable and are not capable of out of the box data logging.

6.2 Resource Utilisation

Through assessing the resource utilisation of the system under load and comparing this against baseline results, a good evaluation of the efficiency of the testbed was produced. Furthermore, these results provided insight into the level of its scalability as well as the potential limitations of the platforms.

6.2.1 EPC

As claimed by srsLTE, the EPC component proved to be extremely lightweight and efficient. Despite being deployed on the most under resourced host machine, the EPC saw no performance bottleneck due to lack of resources. CPU utilisation reached a peak value of 50% during initialisation, with almost no discernible activity during traffic flow between EPC and UE. The same is true for memory usage which only saw a marginal increase in usage during initialisation. Consequently, the srsLTE EPC holds could potentially be deployed in machines with extreme computation and memory constraints.

6.2.2 eNodeB

As expected, the eNodeB was comparatively quite aggressive in terms of resource utilisation. Drawing from the knowledge base compiled in earlier chapters, this result was somewhat expected. Due to the multiple responsibilities held by the eNodeB such as baseband signal processing, resource scheduling and management of communications with the EPC, there is no doubt as to why so many resources were allocated to the eNodeB. Fortunately, the resource demands were well within the constraints of the host system, with allocation of resources during UE attachment and traffic generation procedures still leaving a fair sized pool of unused resources. This leads to the belief that the system would work well in larger deployment scenarios and would theoretically not require overly expensive GPPs for large scale implementation.

6.3 Throughput Performance

The main purpose of throughput performance testing was to analyse the functional capability of the system by generating user traffic in both the uplink and downlink directions using the two most fundamental communication protocols. As the underlying communication protocols of most data transfers, these tests would serve to simulate real world traffic from mobile subscribers. The results of which can used to analyse the signal processing capabilities of the system, the MAC level resource scheduling functionalities of the system for multiple UE attachment, as well as performance differences between various COTS UEs.

6.3.1 Emulated UE

The ability to generate traffic between the emulated UE and CN successfully proved that the Linux tunnel interface was able to route traffic through the srsLTE provided radio link. Throughput results attained are mostly in line with all expectations. Namely; the ability of UDP to attain higher throughput rates, a doubling of throughput for a doubling of RBs, and a significantly higher throughput rate in the downlink direction. This holds the advantage of being able to execute custom traffic generation simulations under realistic RF conditions. Furthermore, the emulated UE allows for full control over the level of information logging, general system configuration, as well as the selected operating frequency.

6.3.2 COTS UE

Throughput testing with COTS UE proved to be a lot more intricate when compared to its emulated counterpart. While all COTS UE devices were capable of data transfer through the LTE network, results varied significantly between devices. It is unclear why such a significant difference in the results was observed, but is assumed to be partly due to the different processing chains and hardware implemented within these UE. Another potential reason could include the fact that LTE signal transmission took place in a preoccupied space. While transmissions from the testbed should have rendered preexisting transmissions as momentarily null, interference cannot be ruled out as a cause of the discrepancy. As such, obvious limitations of COTS UE testing include inconsistencies between hardware implementations, limited control over the UE support LTE bands, as well as limited ability to run custom software on the devices.

Furthermore, attempts made to conduct throughput testing with multiple COTS UE attached to the network proved fruitless. While all UEs was able to successfully attach to the network simultaneously, devices would simply lose connection if more than two devices tried to facilitate traffic throughput. Once again, a major cause of concern would be the RF space in which these devices were operating. Another potential reason for this failure could be a limited functional ability of the MAC level scheduler of the eNodeB. This may be a potential system limitation of srsLTE that could prevent testing of a dense deployment scenario. Additional investigation into this matter would have to be made in order to make a more concrete finding.

6.4 Coexistence

The coexistence strategy implemented does not comply with regulated attempts at coexistence, but leans more toward an LTE-U implementation. This is as the strategy does not employ any LBT functionality and works on the basis of periodic transmission. A finding from this primitive implementation is that this method of coexistence does not meet the functional requirement laid out by the LTE-U and LTE-LAA standards. This is as no level of periodicity prevented the WiFi network from experiencing some level of resource starvation. Further, throughput of the LTE network is theoretically directly proportional to the transmission duty cycle. Therefore, LTE throughput would come at a large expense when implementing periodic transmissions schemes.

While the matter of coexistence should remain in the forefront when exploring more efficient methods of spectrum utilisation for upcoming 5G technologies, the main purpose of coexistence strategy development and testing in the context of this project is more focused on ease of reconfigurability and extended development capability of open source testbed platforms.

This investigation revealed that the srsLTE platform offers high-level functions that implement the various LTE procedures. Example code provided by the srsLTE platform showcases how to appropriately use these functions to implement various LTE processing chains such as the PDSCH. For the purpose of this investigation, implementation of a simple discontinuous transmission scheme was quite straightforward and is achieved by simply setting the transmission gain to null during the duty cycled off periods. Unfortunately, this solution is quite limited as it does not realise additional functionality in the PHY and MAC layer needed for true discontinuous transmission. Through my experience working with the testbed and the advice researchers in the field, it was concluded that implementation of a true discontinuous transmission and reception scheme might be well suited to an srsLTE based implementation, but would require a great level of familiarity and experience developing mobile telecommunication stacks. Furthermore, integrating this into the full LTE stack may fall far outside of the time and technical capabilities of an undergraduate project.

Chapter 7

Conclusions

The main objectives specified at the outset of this project included delivery of literature review detailing the tools and theory needed to implement an open source over-the-air LTE testbed, a fully working implementation of such a testbed, and an investigation and implementation of a primitive coexistance strategy for LTE and WiFi in the unlicensed ISM bands. This section aims to discuss the outcomes of these goals by drawing on the results produced and the associated discussions.

7.1 Over-The-Air Testbed

Investigations into the open source tools available for implementing an over-the-air LTE testbed produced two main options; OAI and srsLTE. Apparent with either option was that an extensive knowledge base was required in order to successfully realise the tesbed. As such, a comprehensive review of the literature has been presented focusing on the history of mobile telecommunications and the cumbersome LTE specifications and functionalities. Following this, a fully operational LTE testbed was realised through a full srsLTE stack. Attempts were made at interconnecting this tested with various other OAI and even commercial components. While not fully successful, these experiments showed full 3GPP compliance of the testbed and the ability for potential heterogeneous architectures through some more tweaking. This would hold significant value to both researchers and industry as it allows for a multitude of deployment configurations. For researchers, this would limit constraints on the configurable nature of the deployments, allowing for easy swapping of the main network nodes to evaluate other platforms or even assess backwards compatibility with new 5G enabled devices as they are released. Following this, experimentation with both emulated and COTS UE confirmed the successful "over-the-air" implementation.

Through this process, it was found that the srsLTE platform has inherent limitations including in a non-distributive CN architecture and potentially not supporting resource management for multiple UEs. These limitations are offset by the platforms ease of implementation and reconfigurability, making it a perfect candidate for educational contexts. This is compounded by the fact the true IP packets produced by the network can be captured and viewed in real-time. Additionally, the well structured and transparent nature of srsLTE makes is a good candidate for developmental purposes. OAI on the other hand offers the same level of functionality in a less transparent manner, but is more poised to support future 5G technology. Ultimately, the implemented LTE testbed, as well as findings and knowledge base produced, meet the original objectives and is considered successful.

7.2 Coexistence in the ISM Bands

Though implementation of the coexistence strategy was quite simple in nature, it successfully showed that open source platforms like srsLTE can be used to realistically assess the viability of novel algorithms and techniques with both proof of concept implementations (as done in this project) as well as full integration into the LTE stack. Unfortunately, the implemented coexistence strategy does not meet the specifications for performance laid out by the LTE-U and LTE-LAA, meaning that although the purpose of implementation has been served successfully, the coexistence strategy itself is considered inadequate.

Chapter 8

Recommendations

Based on the shortcomings of the implemented srsLTE testbed, a number of recommendations are made for further development and testing of the network. These include:

- Separation of EPC components.
- Research and diagnosis of OAI UE's failure to attach to both SRS and OAI based eNodeBs.
- Research and diagnosis of srsLTE's inability to support multiple UE traffic streams.
- Conducting over-the-air testing with COTS UE in a fully RF shielded environment.

These suggestions serve to increase the scope of experimentation that can be done, as well as to create a more consistent and predictable RF environment in which to carry out experiments. This is as seperation of the CN components and diagnosis of the OAI UE would enable a wider variety of architecture configurations, and potential support for the planned OAI 5G release [60]. Likewise diagnosis of the issues related to multiple UE traffic streams would solidify the testbeds capability to realise realistic deployment scenarios. Testing should ideally take place in a fully shielded RF environment when working with COTS UE. This would serve to prevent inconsistent results due to interference from the already crowded LTE bands.

Based on observed capabilities of over-the-air testbeds, the following additions are proposed for future implementation:

- Implementation of a custom traffic generation solution.
- Deployment of a testbed using a Raspberry Pi.

Through the use of a customised traffic generator for either Linux or Android devices, network environments could be made to more realistically simulate and evaluate the performance of a system. While the OAISIM implementation [7] provides a number of different traffic generation profiles, there are no available implementations for over-the-air UE and COTS UE. Additionally, resource utilisation testing proved the testbeds to be quite lightweight. One potential use case for this is miniaturisation and deployment of an LTE network on highly resource constrained GPP hosts such as the Raspberry Pi. This holds great value as it could be used for inexpensive deployment of mobile networks to under resourced regions of South Africa.

Finally, with regards to the coexistence strategy, it is recommended that a true discontinuous transmission and reception scheme be developed for the PDSCH processing chain, perhaps also employing channel sensing functionality. Such an implementation would be more in line with LTE-U and LTE-LAA standards and would provide more meaningful results on the possibility of coexistence.

Chapter 9

References

[1] A. Addo, "The adoption of mobile phone: How has it changed us socially?" *Issues in Business Management and Economics*, vol. 1, no. 3, pp. 47–60, 2013. [Online]. Available: http://www.journalissues.org/journals-home.php?id=2

[2] Statista, "Mobile Percentage of Website Traffic 2020," jul 2020. [Online]. Available: https://www.statista.com/statistics/277125/share-of-website-traffic-coming-from-mobile-devices/

[3] O. Boric-Lubecke, V. M. Lubecke, B. Jokanovic, A. Singh, E. Shahhaidar, and B. Padasdao, "Microwave and wearable technologies for 5g," in *2015 12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*, 2015, pp. 183–188.

[4] Verizon, "When was 5G introduced? ," dec 2019. [Online]. Available: https://www.verizon.com/about/our-company/5g/when-was-5g-introduced

[5] R. Wang, Y. Peng, H. Qu, W. Li, H. Zhao, and B. Wu, "OpenAirInterface-An effective emulation platform for LTE and LTE-Advanced," *International Conference on Ubiquitous and Future Networks, ICUFN*, pp. 127–132, 2014.

[6] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "OpenAirInterface: A flexible platform for 5G research," *Computer Communication Review*, vol. 44, no. 5, pp. 33–38, 2014.

[7] K. Majola and J. Mwangama, "Implementation Of A Radio And Core Mobile Network Using Virtualisation," Undergraduate Honors Thesis, University of Cape Town, 2017.

[8] I. A. M. Balapuwaduge and F. Y. Li, "Cellular Networks: An Evolution from 1G to 4G," *Encyclopedia of Wireless Networks*, no. July, pp. 170–175, 2020.

[9] Tutorialspoint, "Difference between 1G and 2G protocols." [Online]. Available: https://www.tutorialspoint.com/difference-between-1g-and-2g-protocols

[10] A. H. Khan, M. A. Qadeer, J. A. Ansari, and S. Waheed, "4G as a next generation wireless network," *Proceedings - 2009 International Conference on Future Computer and Communication, ICFCC 2009*, no. May 2014, pp. 334–338, 2009.

[11] Qualcomm, "The Evolution of Mobile Technologies," Tech. Rep., 2014.

[12] "What are the differences between 1G, 2G, 3G, 4G and 5G?" [Online]. Available: http://net-informations.com/q/diff/generations.html

[13] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, "srsLTE: An open-source platform for LTE evolution and experimentation," *Proceedings of*

the Annual International Conference on Mobile Computing and Networking, MOBICOM, vol. 03-07-Octo, pp. 25–32, 2016.

[14] "GPRS & EDGE." [Online]. Available: https://www.3gpp.org/technologies/keywords-acronyms/102-gprs-edge

[15] Wikipedia, "GSM." [Online]. Available: https://en.wikipedia.org/wiki/GSM

[16] G. El, M. Zhioua, H. Labiod, N. Tabbane, S. Tabbane, G. El, M. Zhioua, H. Labiod, N. Tabbane, S. Tabbane, and L. T. E. Advanced, "LTE Advanced Relaying Standard : A Survey To cite this version : HAL Id : hal-00842087 LTE Advanced Relaying Standard : A survey," 2013.

[17] A. Kumar, "3G Networks: Opportunities and Challenges," *Bulletin of Mathematical Sciences and Applications*, vol. 3, no. April 2013, pp. 28–36, 2013.

[18] A. I. Pang, J. C. Chen, Y. K. Chen, and P. Agrawal, "Mobility and session management: UMTS vs. CDMA2000," *IEEE Wireless Communications*, vol. 11, no. 4, pp. 30–43, 2004.

[19] M. Iftikhar, N. A. Zaben, W. M. Al-Salih, I. A. Shoukat, M. Uddin, M. Talha, B. Landfeldt, and A. Zomaya, "On the provisioning of QoS mapping in cellular and IP Networks Using a Translation (Function) Matrix," *Information (Japan)*, vol. 16, no. 5, pp. 3033–3068, 2013.

[20] V. Niemi and H. Kaaranen, *Security in the UMTS Environment*. John Wiley Sons, Ltd, 2005, ch. 9, pp. 253–283. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/047001105X.ch9

[21] Electronics Notes, "3G UMTS Network Architecture." [Online]. Available: https://www.electronics-notes.com/articles/connectivity/3g-umts/network-architecture.php

[22] 3rd Generation Partnership Project, "Release 8 Technical Specifications," 2009. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/21{_}series/21.101/21101-800.zip

[23] C. Mehlführer, M. Wrulich, J. C. Ikuno, D. Bosanska, and M. Rupp, "Simulating the long term evolution physical layer," *European Signal Processing Conference*, no. Eusipco, pp. 1471–1478, 2009.

[24] 3rd Generation Partnership Project, "Release 12 Technical Specifications," 2016.

[25] M. A. M. Al-Shibly, M. H. Habaebi, and J. Chebil, "Carrier aggregation in long term evolution-advanced," in *2012 IEEE Control and System Graduate Research Colloquium*, 2012, pp. 154–159.

[26] M. Gahadza and S. Winberg, "Performance of massive mimo systems for future generation wireless systems," in *2019 IEEE 10th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT)*, 2019, pp. 204–211.

[27] G. E. M. Zhioua, H. Labiod, N. Tabbane, and S. Tabbane, "Lte advanced relaying standard: A survey," *Wireless Personal Communications*, vol. 72, 10 2013.

[28] Halberd Bastion, "LTE Network Evolutions." [Online]. Available: https://halberdbastion.com/technology/cellular/4g-lte/lte-evolutions

[29] A. C. Rasmussen and M. R. Kielgast, "Embedded Massive MTC Device Emulator for LTE using Software Defined Radios," p. 99, 2017.

[30] A. Länsisalmi and A. Toskala, *System Architecture Based on 3GPP SAE*. John Wiley Sons, Ltd, 2011, ch. 3, pp. 23–66. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119992943.ch3

[31] Y. Chen and X. Lagrange, "Architecture and Protocols of EPC-LTE with relay," 2013. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00830621

[32] S. Sesia, I. Toufik, and M. Baker, *LTE, The UMTS Long Term Evolution: From Theory to Practice*. Wiley Publishing, 2009.

[33] 3GPP, "LTE Overview." [Online]. Available: https://www.3gpp.org/technologies/keywords-acronyms/98-lte

[34] "What is 5G?" oct 20. [Online]. Available: https://www.thalesgroup.com/en/markets/digital-identity-and-security/mobile/inspired/5G

[35] F. Al-Ogaili and R. M. Shubair, "Millimeter-wave mobile communications for 5G: Challenges and opportunities," *2016 IEEE Antennas and Propagation Society International Symposium, APSURSI 2016 - Proceedings*, no. June 2016, pp. 1003–1004, 2016.

[36] CableFree, "5G NR gNodeB base Stations ." [Online]. Available: https://www.cablefree.net/5g-lte/5g-nr-gnodeb-base-station/

[37] O. Neji, N. Chendeb, O. Chabbouh, N. Agoulmine, and S. Ben Rejeb, "Experience deploying a 5G C-RAN virtualized experimental setup using OpenAirInterface," *2017 IEEE 17th International Conference on Ubiquitous Wireless Broadband, ICUWB 2017 - Proceedings*, vol. 2018-Janua, pp. 1–5, 2018.

[38] 3GPP, "Firm decision on Rel-17 delay in December," 2020. [Online]. Available: https://www.3gpp.org/news-events/2136-r17{_}delay

[39] "Commercial 5G deployments and subscriptions increasing worldwide," sep 2020. [Online]. Available: https://www.techrepublic.com/article/commercial-5g-deployments-and-subscriptions-increasing-worldwide/

[40] Tutorialspoint, "LTE Protocol Stack Layers." [Online]. Available: https://www.tutorialspoint.com/lte/lte{_}protocol{_}stack{_}layers.htm

[41] "Radio Link Control (RLC)."

[42] "Packet Data Convergence Protocol (PDCP)."

[43] 3GPP, "NAS." [Online]. Available: https://www.3gpp.org/technologies/keywords-acronyms/96-nas

[44] "PHY Basics," jun 2015. [Online]. Available: http://www.revolutionwifi.net/revolutionwifi/2015/3/how-ofdm-subcarriers-work

[45] Huawei, "TDD LTE Frame Structure," jun 2018. [Online]. Available: https://forum.huawei.com/enterprise/en/tdd-lte-frame-structure/thread/462385-100305

[46] C. A. Garc, A. M. Recio-p, and P. Merino-g, "PerformLTE : A Testbed for LTE Testing PerformLTE : A Testbed for Mobile Experimentation," pp. 46–59, 2015.

[47] E. Weingärtner, H. Vom Lehn, and K. Wehrle, "A performance comparison of recent network simulators," *IEEE International Conference on Communications*, no. May, 2009.

[48] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE cellular systems: An open-source framework," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 2, pp. 498–513, 2011.

[49] F. Kaltenberger, A. P. Silva, A. Gosain, L. Wang, and T. T. Nguyen, "OpenAirInterface: Democratizing innovation in the 5G Era," *Computer Networks*, vol. 176, no. May, p. 107284, 2020. [Online]. Available: https://doi.org/10.1016/j.comnet.2020.107284

[50] C. Yahiaoui, Y. Aouine, M. Bouhali, and N. Bessah, "Simulating the long term evolution (LTE) downlink physical layer," *Proceedings - UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, UKSim 2014*, pp. 531–535, 2014.

[51] S. Kukade, M. Sutaone, and R. Patil, *Evaluation of SC-FDMA Physical Link Using USRP*, 08 2020, pp. 1003–1017.

[52] P. Lin, S. Huang, and X. Li, "Teaching and learning next generation mobile communication networks through open source openairinterface testbeds," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2017, pp. 1475–1478.

[53] H. Shen, X. Wei, H. Liu, Y. Liu, and K. Zheng, "Design and implementation of an LTE system with multi-thread parallel processing on OpenAirInterface platform," *IEEE Vehicular Technology Conference*, vol. 0, 2016.

[54] J. Feng, B.-K. Hong, and S.-M. Cheng, *DDoS Attacks in Experimental LTE Networks*, 03 2020, pp. 545–553.

[55] V. Marojevic, D. Chheda, R. M. Rao, R. Nealy, J. Park, and J. H. Reed, "Software-defined lte evolution testbed enabling rapid prototyping and controlled experimentation," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.

[56] B. Johansson and T. Sundin, "Lte test bed."

[57] "OpenLTE." [Online]. Available: http://openlte.sourceforge.net/

[58] "EURECOM." [Online]. Available: http://www.eurecom.fr/en

[59] "OpenAirInterface." [Online]. Available: https://www.openairinterface.org/

[60] OpenAirInterface, "OpenAirInterface Software Alliance Commits to Provide Full 5G Software Stack by December 2020," jul 2020. [Online]. Available: https://www.openairinterface.org/?news=press-item-the-openairinterface-software-alliance-commits-to-provide-full-5g-software-stack-by-december-2020

[61] "The OpenAirInterface Software Alliance Commits to Provide Full 5G Software Stack by December 2020." [Online]. Available: https://www.openairinterface.org/?news=press-item-the-openairinterface-software-alliance-commits-to-provide-full-5g-software-stack-by-december-2020

[62] F. A. de Figueiredo, W. Liu, X. Jiao, and I. Moerman, "Demo Abstract: Packetized-LTE Physical Layer Framework for Coexistence Experiments," *SenSys 2017 - Proceedings of the 15th ACM Conference on Embedded Networked Sensor Systems*, vol. 2017-Janua, 2017.

[63] OpenAirInterface, "OPENAIRINTERFACE reorganizes repository structure and licensing for openairCN code," sep 2020. [Online]. Available: https://www.openairinterface.org/?news=openairinterface-reorganizes-repository-structure-and-licensing-for-openaircn-codes

[64] "Software Radio Systems." [Online]. Available: https://www.softwareradiosystems.com/

[65] "Vector Optimized Library of Kernels." [Online]. Available: https://www.libvolk.org/

[66] V. Maglogiannis, D. Naudts, P. Willemen, and I. Moerman, "Impact of LTE operating in unlicensed spectrum on wi-fi using real equipment," *2016 IEEE Global Communications Conference, GLOBECOM 2016 - Proceedings*, 2016.

[67] J. Jeon, H. Niu, Q. C. Li, A. Papathanassiou, and G. Wu, "LTE in the unlicensed spectrum: Evaluating coexistence mechanisms," *2014 IEEE Globecom Workshops, GC Wkshps 2014*, pp. 740–745, 2014.

[68] Independent Communications Authority of South Africa, "Invitation to Apply Notice On The Licensing Process for International Mobile Telecommunications," 2020. [Online]. Available: http://www.greengazette.co.za/pages/national-gazette-37230-of-17-january-2014-vol-583{_}20140117-GGN-37230-003

[69] "Here are the rules for South Africa's multi-billion-rand spectrum auction," oct 2020. [Online]. Available: https://mybroadband.co.za/news/wireless/369967-here-are-the-rules-for-south-africas-multi-billion-rand-spectrum-auction.html

[70] F. M. Abinader, V. A. de Sousa, S. Choudhury, F. S. Chaves, A. M. Cavalcante, E. P. Almeida, R. D. Vieira, E. Tuomaala, and K. Doppler, "LTE/Wi-Fi Coexistence in 5 GHz ISM Spectrum: Issues, Solutions and Perspectives," *Wireless Personal Communications*, vol. 99, no. 1, pp. 403–430, 2018. [Online]. Available: https://doi.org/10.1007/s11277-017-5114-2

[71] FCC, "Federal Communications Commission, Office of Engineering and Technology and Wireless Telecommunications Bureau seek information on current trends in LTE-U and LAA technology," vol. 2015, no. 21 May 2015, pp. 29–31, 2015. [Online]. Available: http://transition.fcc.gov/Daily{_}Releases/Daily{_}Business/2015/db0505/DA-15-516A1.pdf

[72] "Wi-Fi Alliance® statement on License-Assisted Access (LAA) — Wi-Fi Alliance." [Online]. Available: https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-statement-on-license-assisted-access-laa

[73] Qualcomm Incorporated, "LTE in Unlicensed Spectrum : Harmonious Coexistence with Wi-Fi," *White Paper*, no. June, pp. 1–19, 2014.

[74] "LTE-U Forum." [Online]. Available: https://lteuforum.org/

[75] National Intruments, "Real-time LTE/Wi-Fi Coexistence Testbed," mar 2019. [Online]. Available: https://www.ni.com/en-za/innovations/white-papers/16/real-time-lte-wi-fi-coexistence-testbed.html

[76] A. Mukherjee, J. F. Cheng, S. Falahati, H. Koorapaty, D. H. Kang, R. Karaki, L. Falconetti, and D. Larsson, "Licensed-Assisted Access LTE: Coexistence with IEEE 802.11 and the evolution toward 5G," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 50–57, 2016.

[77] "Real-time LTE/Wi-Fi Coexistence Testbed - NI." [Online]. Available: https://www.ni.com/en-za/innovations/white-papers/16/real-time-lte-wi-fi-coexistence-testbed.html

[78] 3GPP, "LTE Release 13," 2015.

[79] "SDLC V-Model," May 2019. [Online]. Available: https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/

[80] "CSIR." [Online]. Available: https://www.csir.co.za/

[81] "srsLTE." [Online]. Available: https://www.srslte.com/

[82] "srsLTE 20.04.0 Documentation." [Online]. Available: https://docs.srslte.com/

[83] "S1AP/NAS Server." [Online]. Available: https://www.synopsys.com/software-integrity/security-testing/fuzz-testing/defensics/protocols/s1ap-server.html

[84] TechLibrary, "SCTP Overview," sep 2018. [Online]. Available: https://www.juniper.net/documentation/en{_}US/junos/topics/topic-map/security-gprs-sctp.html

[85] "USRP™ Hardware Driver." [Online]. Available: https://github.com/EttusResearch/uhd

[86] "Wireshark." [Online]. Available: https://www.wireshark.org/

[87] "Wifi Analyzer." [Online]. Available: https://play.google.com/store/apps/details?id=com.farproc.wifi.analyzer{&}hl=en{_}ZA{&}gl=US

[88] "Rohde & Schwarz Handheld Spectrum Analyzer." [Online]. Available: https://www.rohde-schwarz.com/pl/product/fsh-productstartpage{_}63493-8180.html

[89] Ettus Research, "USRP B210 USB Software Defined Radio (SDR)." [Online]. Available: https://www.ettus.com/all-products/ub210-kit/

[90] National Instruments, "USRP-2944 ." [Online]. Available: https://www.ni.com/en-za/support/model.usrp-2944.html

[91] Rocketeck, "Smart Card Reader ." [Online]. Available: https://www.rocketek.hk/products/card-readers-2/smart-card-reader/

[92] "GPRS Tunneling Protocol (GTP) in LTE." [Online]. Available: http://www.simpletechpost.com/2013/03/gprs-tunneling-protocol-gtp-in-lte.html

[93] K. Swamy, "How LTE Stuff Works ?" pp. 1–7, 2018. [Online]. Available: http://howltestuffworks.blogspot.com/2012/01/emm-information.html

[94] "Open5GCore." [Online]. Available: https://www.open5gcore.org/

[95] "LTE Authentication." [Online]. Available: https://www.sharetechnote.com/html/Handbook{_}LTE{_}Authentication.html

[96] Dnsstuff, "How to Fix Packet Loss," oct 2019. [Online]. Available: https://www.dnsstuff.com/reduce-packet-loss

[97] “iPerf.” [Online]. Available: https://iperf.fr/

[98] “OpenAirKernelMainSetup.” [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/-/wikis/OpenAirKernelMainSetup

[99] Ettus Knowledge Base, “Building and Installing the USRP UHD on Linux ,” may 2016. [Online]. Available: https://kb.ettus.com/Building{_}and{_}Installing{_}the{_}USRP{_}Open-Source{_}Toolchain{_}(UHD{_}and{_}GNU{_}Radio){_}on{_}Linux

[100] “OpenAirInterface5G Wiki.” [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/-/wikis/HowToConnectOAIENBWithOAIUEWithoutS1Interface

Chapter 10

EBE Faculty: Assessment of Ethics in Research Projects

Application for Approval of Ethics in Research (EiR) Projects
Faculty of Engineering and the Built Environment, University of Cape Town

ETHICS APPLICATION FORM

Please Note:
Any person planning to undertake research in the Faculty of Engineering and the Built Environment (EBE) at the University of Cape Town is required to complete this form **before** collecting or analysing data. The objective of submitting this application *prior* to embarking on research is to ensure that the highest ethical standards in research, conducted under the auspices of the EBE Faculty, are met. Please ensure that you have read, and understood the **EBE Ethics in Research Handbook** (available from the UCT EBE, Research Ethics website) prior to completing this application form: http://www.ebe.uct.ac.za/ebe/research/ethics1

APPLICANT'S DETAILS		
Name of principal researcher, student or external applicant		Matthew Lock
Department		Department of Electrical Engineering
Preferred email address of applicant:		lckmat002@myuct.ac.za
If Student	Your Degree: e.g., MSc, PhD, etc.	BSc Electrical and Computer Engineering
	Credit Value of Research: e.g., 60/120/180/360 etc.	40
	Name of Supervisor (if supervised):	Dr Joyce Mwangama
If this is a researchcontract, indicate the source of funding/sponsorship		
Project Title		Over the Air Implementation of a Mobile Network

I hereby undertake to carry out my research in such a way that:

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

APPLICATION BY	Full name	Signature	Date
Principal Researcher/ Student/External applicant	Matthew Lock		16/08/2020
SUPPORTED BY	Full name	Signature	Date
Supervisor (where applicable)	Joyce Mwangama		17/08/2020

APPROVED BY	Full name	Signature	Date
HOD (or delegated nominee) Final authority for all applicants who have answered NO to all questions in Section 1; and for all Undergraduate research (Including Honours).	A/Prof F Nicolls pp J Buxey	Dept. Manager: Elec Eng Authorised to sign obo HOD	28.8.2020
Chair: Faculty EIR Committee For applicants other than undergraduate students who have answered YES to any of the questions in Section 1.			

Appendix A

Iperf Flags and Usage

Detailed in this appendix are the various use cases and flags used when operating the iperf package.

A.1 Flags

Table A.1 provides information list of all the iperf flags used throughout the duration of this project, along with a brief description of each flag's functionality [97].

Flag	Functionality
-b	Set target bandwidth to n bits/sec (default 1 Mbit/sec for UDP, unlimited for TCP).
-c	Run iPerf in client mode, connecting to an iPerf server running on host.
-i	Sets the interval time in seconds between periodic bandwidth, jitter, and loss reports.
-r	Reverse test mode. Following initial test, client and server roles switch.
-t	The time in seconds to transmit for.
-u	Use UDP rather than TCP.
-y C	Report as a Comma-Separated Values

Table A.1: Detailed list of iperf flags and their functions.

A.2 Typical Usage

Provided in this section are typical use cases of iperf for generation both UDP and TCP traffic.

A.2.1 Server Application

Listing A.1 shows the typical commands used when wanting to host a TCP server. The `-i` flag indicates a reporting interval of one second, while the `-y C` flag indicates a CSV formatted output. This will make for regular feedback that can be easily imported into a worksheet. Listing A.2 shows an identical implementation for a EDP server. The option for UDP traffic is realised though the `-u` flag. Both listings can be easily wrapped into bash scripts called **tcp_server_init.sh** and **udp_server_init.sh** for quick execution.

```
iperf -s -i 1 -y C
```

Listing A.1: Example of an iperf use case showing TCP server.

```
iperf -u -s -i 1 -y C
```

Listing A.2: Example of an iperf use case showing UDP server.

A.2.2 Client Application

Listing A.3 shows the typical commands used when wanting to initialise a TCP client. As before, these clients are initialised with a reporting interval of one second and a CSV formatted output and will run for 60 seconds as specified by the `-t` flag. The differences between server and client initialisation include the fact that the client must specify the server IP address by means of the `-c` flag. An example is provided in the listing. Listing A.4 shows the equivalent for a UDP client. Note how the UDP client specifies its targeted bandwidth through the `-b` flag. An important consideration is that the selected traffic protocol for both client and server must match.

```
iperf -c 172.16.0.1 -i 1 -y C -t 60
```

Listing A.3: Example of an iperf use case showing TCP client.

```
iperf -c 172.16.0.1 -u -i 1 -y C -b 5M -t 60
```

Listing A.4: Example of an iperf use case showing UDP client.

An improvement that can be made is the wrapping of the clients into bash scripts that can be easily executed. An example of this is shown in Listing A.5 while an example of its usage is shown in Listing A.6 where the server IP address and targeted bandwidth are parsed in as arguments. These generics scripts are used to create the **tcp_client_init.sh** and **udp_client_init.sh** scripts. The udp script has an additional `-u` flag.

```
iperf -c $1 -i 1 -y C -b $2 -t $3
```

Listing A.5: Generic bash script for a TCP client.

```
$ ./genericScript.sh 172.16.0.1 80M 60
```

Listing A.6: Running generic bash script for a TCP client.

A.2.3 Reverse Mode

Additionally, the `-r` flag can be run when initiating an iperf server. This will result in the traffic generation running as normally for the specified duration, after which the role of the client and server will switch for the duration specified.

Appendix B

Installation Guides

Detailed in this appendix is the installation process for Linux Low Latency Kernels, USRP Hardware Driver Software, the srsLTE testbed platform, as well as the relevant components of the OAI testbed platform.

B.1 Linux Low Latency Kernels Installation

The Linux Low Latency Kernels described here are prerequisite to the OAI installation process. As OAI operates under tight time constraints, a combination of low latency kernels and steady CPU frequencies through appropriately set CPU flags are required. The following instructions are taken from the official OAI installation guide [98] and has been tested on Ubuntu 18.0.4 LTS for this project.

B.1.1 Low Latency Kernels

1. Install the low-latency kernel.

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

B.1.2 Setting the CPU Flags

1. Disable p-state and c-state.
   Append the following line to Linux boot options, i.e. in the `$/etc/default/grub` file.

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_pstate=disable"
processor.max_cstate=1
intel_idle.max_cstate=0
idle=poll
```

2. Perform grub update.

```
$ update-grub
```

3. Blacklist the intel_powerclamp module.
   Append the following line to the $/etc/modprobe.d/blacklist.conf file. Should the file not exist, create one.

```
blacklist intel_powerclamp
```

4. Istall i7z and cpufrequtils.

```
$ sudo apt-get install i7z
$ udo apt-get install cpufrequtils
```

5. Disable CPU frequency scaling.
   Append the following line to the $/etc/default/cpufrequtils file. Should the file not exist, create one.

```
GOVERNOR="performance"
```

6. Disable CPU frequency scaling.
   Append the following line to the $/etc/default/cpufrequtils file. Should the file not exist, create one.

```
GOVERNOR="performance"
```

7. Restart cpufrequtils
   Append the following line to the $/etc/default/cpufrequtils file. Should the file not exist, create one.

```
$ sudo update-rc.d ondemand disable
$ sudo /etc/init.d/cpufrequtils restart
```

8. Reboot the machine

```
$ sudo reboot
```

9. Run i7z to ensure stable clock speeds and that c-states are disabled.

```
$ sudo i7z
```

B.2 USRP Hardware Driver (UHD) Installation

The following instructions are taken from the official Ettus guide for UHD installation on Linux [99]. The outcomes of this process include fully working UHD **Version 3.15.0.0** software to enable interfacing from host machine to USRP SDR frontend as well as the required USB interface setup steps required for interfacing with the USRP B210 through USB3 ports. While this process remains relevant to the USRP 2944, additional setup procedure such as FPGA image flashing and PCI configuration needed to interface with the USRP 2944 are not provided. This is as the USRP 2944 is not owned by UCT and and the exact procedure carried out by the CSIR is not known.

The Linux distribution on which this installation was carried out is Ubuntu 18.04 LTS. Make sure no USRP devices are connected to the machine prior to installation.

B.2.1 Build and Install UHD

1. Update and Install Ubuntu 18.04 Dependencies.

```
$ sudo apt-get update
$ sudo apt-get -y install git swig cmake doxygen build-essential
    libboost-all-dev libtool libusb-1.0-0 libusb-1.0-0-dev
    libudev-dev libncurses5-dev libfftw3-bin libfftw3-dev
    libfftw3-doc libcppunit-1.14-0 libcppunit-dev libcppunit-doc
    ncurses-bin cpufrequtils python-numpy python-numpy-doc python
    -numpy-dbg python-scipy python-docutils qt4-bin-dbg qt4-
    default qt4-doc libqt4-dev libqt4-dev-bin python-qt4 python-
    qt4-dbg python-qt4-dev python-qt4-doc python-qt4-doc
    libqwt6abi1 libfftw3-bin libfftw3-dev libfftw3-doc ncurses-
    bin libncurses5 libncurses5-dev libncurses5-dbg
    libfontconfig1-dev libxrender-dev libpulse-dev swig g++
    automake autoconf libtool python-dev libfftw3-dev libcppunit-
    dev libboost-all-dev libusb-dev libusb-1.0-0-dev fort77
    libsdl1.2-dev python-wxgtk3.0 git libqt4-dev python-numpy
    ccache python-opengl libgsl-dev python-cheetah python-mako
    python-lxml doxygen qt4-default qt4-dev-tools libusb-1.0-0-
    dev libqwtplot3d-qt5-dev pyqt4-dev-tools python-qwt5-qt4
    cmake git wget libxi-dev gtk2-engines-pixbuf r-base-dev
    python-tk liborc-0.4-0 liborc-0.4-dev libasound2-dev python-
    gtk2 libzmq3-dev libzmq5 python-requests python-sphinx
    libcomedi-dev python-zmq libqwt-dev libqwt6abi1 python-six
    libgps-dev libgps23 gpsd gpsd-clients python-gps python-
    setuptools
```

2. Clone git repository.

```
$ cd $HOME
$ mkdir workarea
$ cd workarea
$ git clone https://github.com/EttusResearch/uhd
```

3. Checkout version 3.15.0.0

```
$ cd uhd
$ git checkout v3.15.0.0
```

4. Build UHD.

```
$ cd host
$ mkdir build
$ cd build
$ cmake ../
$ sudo make
```

5. Verify successful completion of build process.

```
$ make test
```

6. Install UHD.

```
$ sudo make install
```

7. Update the system's shared library cache.

```
$ sudo ldconfig
```

8. Export the library path.
   The following line is added to your $HOME/.bashrc file

```
export LD_LIBRARY_PATH=/usr/local/lib
```

9. Verify installation.
   Running the $uhd_find_devices command should produce the following output:

```
linux; GNU C++ version x.x.x; Boost_106501 UHD_3.15.0.HEAD-0-
    gaea0e2de
No UHD Devices Found
```

10. Download the UHD FPGA images.

```
$ sudo uhd_images_downloader
```

B.2.2 Configuring USB

The following commands install a udev rule so that non-root users may access the USRP device through the USB interface.

1. Configure USB Rules.

```
$ cd $HOME/workarea/uhd/host/utils
$ sudo cp uhd-usrp.rules /etc/udev/rules.d/
$ sudo udevadm control --reload-rules
$ sudo udevadm trigger
```

B.2.3 Setting Thread Priority Scheduling

New threads spawned by UHD may try to boost the thread's scheduling priority. The following commands ensure that this completes successfully for non-root users.

1. Create group.

```
$ sudo groupadd usrp
$ sudo usermod -aG usrp $USER
```

2. Add group.
   Append the following line to end of the file $/etc/security/limits.conf.

```
@usrp - rtprio  99
```

B.2.4 Connect the USRP

At this point you can connect the USRP to the host device. Once the device is connected you may run the $ uhd_find_devices or $ uhd_usrp_probe to confirm that the device can be found and correctly interfaced with.

B.3 Software Radio Systems LTE (srsLTE) Installation

The following instructions are taken from the official srsLTE documentation [82] and are applicable to release **20.04.2**. Outcomes of this process include a full installation of the srsLTE stack inluding EPC, eNodeB, UE and other example applications. While srsLTE claims to support a multitude of Linux flavours, the distribution implemented for this project was Ubuntu 18.04 LTS. Further prerequisties for installation include the UHD software needed to interface with USRP SDR frontends. Details for this are provided in the USRP Hardware Driver (UHD) Installation Section.

1. Install Ubuntu 18.04 Prerequisite Dependencies.

```
$ sudo apt-get install cmake libfftw3-dev libmbedtls-dev
    libboost-program-options-dev libconfig++-dev libsctp-dev
```

2. Clone the git repository.

```
$ git clone https://github.com/srsLTE/srsLTE.git
```

3. Checkout version 20.04.2.

```
$ cd srsLTE
$ git checkout release_20_04_2
```

4. Build srsLTE.

```
$ mkdir build
$ cd build
$ cmake ../
$ sudo make
$ sudo make test
```

5. Instal srsLTE.

```
$ sudo make install
$ sudo srslte_install_configs.sh user
```

   This installs srsLTE and also copies the default srsLTE config files to the user's home directory (/.srs).

B.4 OpenAirInterface (OAI) Installation

The following instructions are adapted from official OAI wiki [100] and are applicable to release **1.1.0**. Outcomes of this process include installation of the OAI eNodeB and UE RAN components for USRP based hardware. AS srsLTE was selected as the primary LTE stack, the OAI EPC is not dealt with in this report. While OAI claims to support a multitude of Linux flavours, the distribution implemented for this project was Ubuntu 18.04 LTS. Prerequisties include the Linux Low Latency Kernels Installation while further dependencies are installed by the automatic build scripts.

1. Clone the git repository.

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

2. Checkout version 1.1.0.

```
$ cd ~/openairinterface5g/
$ git checkout v1.1.0
```

3. Set environmental variables.

```
$ source oaienv
```

4. Install dependencies through the automated build script.

```
$ cd cmake_targets
$ sudo ./build_oai -I
```

5. Run the automated build script for the eNodeB.

```
$ sudo ./build_oai -w USRP --eNB
```

6. Run the automated build script for the UE.

```
$ sudo ./build_oai -w USRP --UE
```

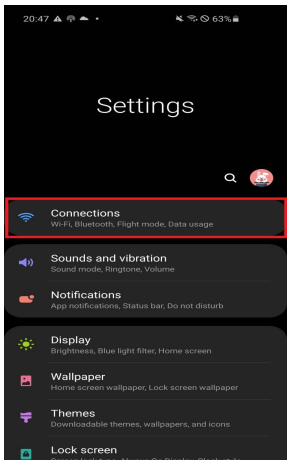
Appendix C

COTS UE APN Configuration

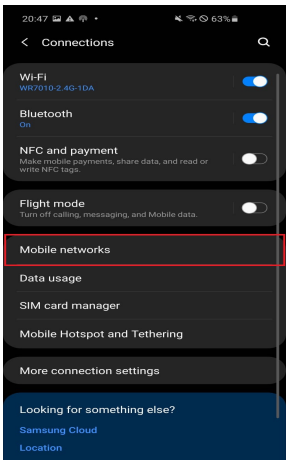
Detailed in this appendix are the procedures for correctly setting the APN settings on both Android and iOS devices. The examples shown configures the mobile devices to have an APN value of "srsapn". Note that APN settings are saved to the device for a specific USIM.

C.1 Android Devices

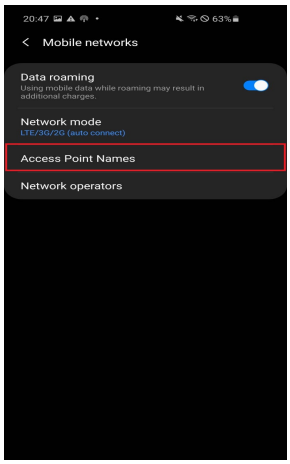
Figure C.1 showcases how to configure APN settings on Android devices. Red rectangles are placed around the suggested next step.



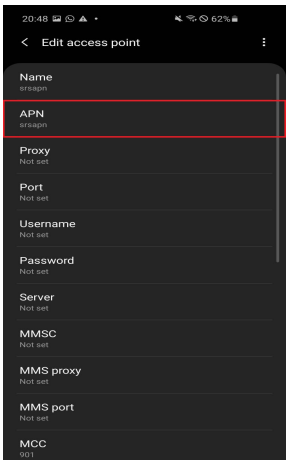
(a) Step 1



(b) Step 2



(c) Step 3

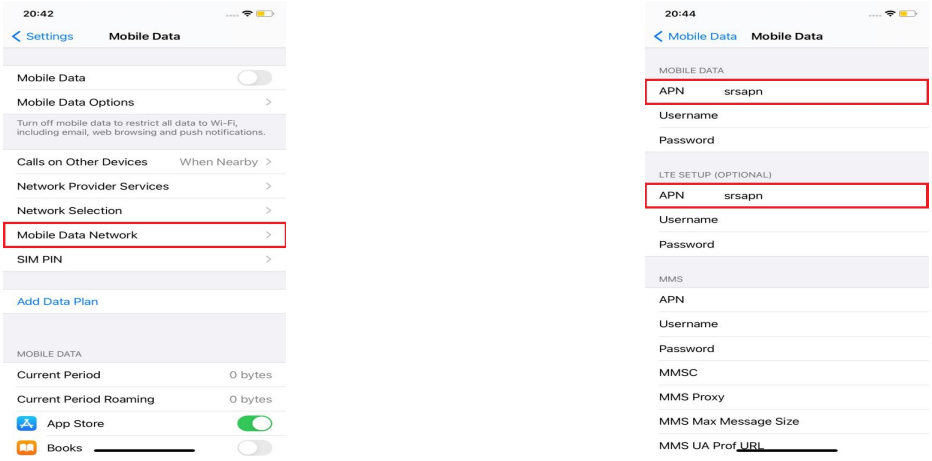


(d) Step 4

Figure C.1: Procedure for configuring APN settings on Android device.

C.2 iOS Devices

Figure C.2 showcases how to configure APN settings on iOS devices. Red rectangles are placed around the suggested next step.



(a) Step 1 **(b)** Step 2

Figure C.2: Procedure for configuring APN settings on iOS device.