

Enhanced Device FingerPrinting in 4G LTE Communication Networks

by

Nrusingha Prasad Panda
(201915006)

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
ELECTRONICS AND COMMUNICATION

with specialization in
Wireless Communication and Embedded Systems
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

A program jointly offered with
C.R.RAO ADVANCED INSTITUTE OF MATHEMATICS, STATISTICS AND COMPUTER SCIENCE



May, 2021

Declaration

I hereby declare that

- i) the thesis comprises of my original work towards the degree of Master of Technology in Electronics and Communications at Dhirubhai Ambani Institute of Information and Communication Technology & C.R.Rao Advanced Institute of Applied Mathematics, Statistics and Computer Science, and has not been submitted elsewhere for a degree,
- ii) Due acknowledgment has been made in the text to all the reference material used.

Nrusingha Prasad Panda

Nrusingha Prasad Panda

Certificate

This is to certify that the thesis work entitled "Enhanced Device FingerPrinting in 4G LTE Communication Networks" has been carried out by Nrusingha Prasad Panda for the degree of Master of Technology in Electronics and Communications at *Dhirubhai Ambani Institute of Information and Communication Technology & C.R.Rao Advanced Institute of Applied Mathematics, Statistics and Computer Science* under our supervision.



Dr. Priyanka Mekala



Dr. Supriya Goel

Acknowledgments

First and foremost I am extremely grateful to my supervisors, Dr. Priyanka Mekala and Dr. Supriya Goel. Their immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life. I would also like to thank Director of CR Rao AIMSCS and DA-IICT for providing me the opportunity as a Master's student. I would like to thank all the members in the CR Rao AIMSCS, and my classmates Mr. Meemoh, Ms. Nidhi and Ms. Somya, it is their kind help and support that made my study and life in University of Hyderabad a wonderful time. Finally, I would like to express my gratitude to my parents and family members. Without their tremendous understanding and encouragement in the past few months, it would be impossible for me to complete my work.

Contents

List of Figures	vi
1 Introduction	1
1.1 The LTE	1
1.2 What is Device Fingerprinting	2
1.3 Background & Motivation	2
1.4 Scope & Objective of the work	3
1.4.1 Our Contribution	3
1.5 Thesis Outline	4
2 Literature Survey	5
3 Overview of LTE	8
3.1 4G and Mobile Communication Generations	8
3.2 4G LTE Architecture	10
3.3 EPC	13
3.4 The OSI Layers Overview	14
3.5 LTE Registration Procedure & CallFlow	17
4 Device FingerPrinting	21
4.1 Problem Statement	21
4.2 The Classification Model	22
4.2.1 Understanding UE Capabilities	22
4.2.2 Block Diagram of Classification Model	22

5	Testbed / Framework-	25
5.1	srsLTE	26
5.2	eNodeB	27
5.3	LTE Core network	28
5.4	srsLTE Build and Install	28
5.5	eNB and EPC configuration	29
5.6	Wireshark Logs	30
6	Results	31
6.1	Classification Results	33
6.1.1	Level-1	33
6.1.2	Level-2	34
6.1.3	Level-3	35
6.1.4	Level-4	36
6.2	GUI for Device Identification	38
6.3	Fuzzy Inference System	39
7	Conclusion & Future Work	42
7.1	Summary of Findings	43
	References	44
8	Appendices	49
8.1	Matlab code for feature identification	49
8.2	Matlab GUI code	51
8.3	FIS code	57

List of Figures

3.1	LTE system architecture	10
3.2	USRP B210 Running on Ubuntu 18.04 machine	12
3.3	LTE core Running on Ubuntu 18.04 machine	13
3.4	Architecture of LTE with layer stacks	16
3.5	UE Attachment with Core network	19
4.1	Flow diagram for 4-Layer Device type identification	24
5.1	Lab Setup Environment	26
5.2	LTE software Suit - www.srslte.com	27
5.3	srsLTE eNB running on Ubuntu18.04	27
5.4	srsLTE EPC on Ubuntu18.04	28
5.5	Wireshark capture with s1ap filter	30
6.1	NAS-PDU in s1ap Filter	31
6.2	Device info in Matlab command window	33
6.3	Parameter- Voice Domain Preference and UE's Usage Settings	33
6.4	Classification- Layer1	34
6.5	Classification- Layer2	35
6.6	Parameter- MS assisted GPS	35
6.7	Classification- Layer3	36
6.8	Parameters for Baseband Processor Identification	36
6.9	Classification- Layer4	37
6.10	App designed in MATLAB for the UE classification	38
6.11	FIS for overall vulnerability level	40

6.12 Surface view for vulnerability level 41

Abstract

Long Term Evolution in the field of mobile wireless communication, its wide availability and the surge of data usage among the millennial in recent years, which led to the new world of information society, plays the pivotal role in our technology driven daily life which demands high performance data transmission, sophisticated security features and extensive integration. Having to serve Billions of 4G users worldwide, LTE network architecture compliments the aspect of user privacy and data security along with connectivity, speed, latency, throughput etc compared to its previous generations.

We want to review the LTE architecture, OSI layers and protocols, specifically focusing on the layer 3. In the layer 3 which is the network layer we can investigate the impact of fingerprinting on encrypted LTE/4G Control Plane traffic and exploiting the vulnerabilities by analysing the control signals. Then auditing different features may lead to extraction of useful information, and classifying them based on our classification model. Also a GUI is designed and a fuzzy inference system is outlined to categorise the threat factor for the individual UEs using the model.

Keywords:- 4G LTE, FingerPrinting, Device Capability, Attach Procedure, Call Flow NAS-PDU, LTE Protocols

Chapter 1

Introduction

1.1 The LTE

As the importance and use cases of 4G LTE and futuristic communication scenarios are well recognised, the transformation in India has been tremendous in the previous few years, starting with commercial 4G rollout in 2016. TRAI (Telecom Regulatory Authority of India)'s 'Wireless Data Services in India' Analytical Report[2] suggests, total number of wireless data subscribers from 281.58 million in 2014 to 424.02 million in 2017 and then 36.36% increase to 578.2 million in subsequent year, such an increment also lead to the volume of total wireless data consumption of 20,092 million GB during the year 2017 to 46,404 million GB during the year 2018 with yearly growth of 131%, which was only 828 million GB during the year 2014. The share of 4g data usage in total volume of wireless data usage has been 86.85% during the year 2018. The average cost to subscriber for per GB wireless data usage was Rs.11.78 per GB during the year 2018 as compare to Rs. 226 per GB before the introduction of 4g LTE [2].

Individual privacy, data security, and the analysis of 4G LTE data traffic on various stages/layers are of highest importance in a country where the average data cost per GB is the lowest in the world and where half a billion people are yet to become smart phone users in the near future.

1.2 What is Device Fingerprinting

Fingerprinting is the capability of identify or re-identify a user, device and collect information from which inferences can be drawn via observable characteristics or parameters. Extraction of distinctive parameters that can lead to a reliable and robust way of device identification (Fingerprinting) by capturing cryptographic credentials & analyzing information across the particular protocol stacks.

1.3 Background & Motivation

Various security Advancements has been made in subsequent generation namely 2G, 3G and 4G . In 2G there was the concern of fake base station attacks and also there was only one way authentication [13]. In the 3G standard, many Internet Protocol (IP) related vulnerabilities have been detected. 4G was more IP based network and due to the increase in the IP traffic it encountered various attacks eg. Denial of service (DOS) attacks.

One of the protocol exploitations that occurred in LTE was the catching of the International mobile subscriber identifier (IMSI). To address this issue, 5G introduced the Subscription Permanent Identifier (SUPI), which replaced the IMSI, and SUPI was encrypted into Subscription Concealed Identifier (SUCI) with the introduction of public key infrastructure (PKI) . Pre-authentication messages were discovered in LTE prior to security establishment.

various device capabilities were exposed in clear text and attacker by disabling it can downgrade the performance of the user , As 4G device will come more enhance device capabilities the operator must make sure to secure these capabilities.

So for device identification, layer-3 RRC control signal message analysis and extracting important feature/parameters to track or identify the nature of a connected device is the only way forward. As future work, this could include analysis of layer-2 User plane traffic and classification of Data Traffic for

Website Fingerprinting which will lead to what kind of device along with what kind of website is being browsed by the connected UE. Making an Embedded Device FingerPrinting and Website FingerPrinting for the improved tracking of UEs in an LTE network is the future possibility. As the call flow and initial attach procedure in both 4G and 5G is exactly same, this work must easily get extended to next generation.

1.4 Scope & Objective of the work

- Implementation of a complete end to end 4G standalone network using open source tools (srsLTE).
- With the help of the 4G network the aim is to analyse the protocols and the call flow to find any device capability transmission as per 3GPP specifications.
- Capture capability from many number of devices and classify them according to extracted features.
- Making of a Graphical User Interface for the data set of devices to visualise the classifications, to indicate vulnerability level of individual UE by a FIS.

1.4.1 Our Contribution

- Implementation of a 4G standalone test bed which consist of all the specification and procedure as per the 3GPP standards.
- Attach different type of 4G devices and capture their traces.
- Identify the important parameters which are responsible for device type identification and classify them in different use case levels.
- According the classification model enhancing it by designing a Graphical User Interface based application and Fuzzy Inference System.

1.5 Thesis Outline

The thesis has been structured as follows

- Chapter 1 is the introduction to the Long Term Evolution, which is the platform of this project.
- Chapter 2 consists of the Literature survey, i.e Which particular aspect of LTE we are focused on and discussing what other related work is been done.
- Chapter 3 contains introduction to LTE, basics, architecture, protocol stack and Call flow log message Analysis.
- Chapter 4 is briefing about this thesis work and the need of this project, also the reason behind the requirement of Classification and extraction of Device Capabilities
- Chapter 5 consists of the Experimental setup in detail.
- Chapter 6 & 7 consists of Results obtained and the Conclusion & Further research can be carried out.

Chapter 2

Literature Survey

The current problem with the LTE stack protocol is that, researches are focused on physical layer (layer 1) and the network layer (layer 2)[1]. Early 2G systems were known to have different vulnerabilities. One of the most prominent attack on 2G systems will be, creation of 'rogue base station'. As there is no mutual authentication (i.e authentication from both User and network) between 2G system (device) and the network which enabled "fake base station" creation and convincing legitimate devices to send connection request to it by the attacker. The use of temporary mobile subscriber identifiers (TMSI) was one of the idea to minimize exposure of user identifiers (known as International Mobile Subscriber Identifier or IMSI) which is transmitted over the air, 2G systems. But still in the absence of mutual authentication (i.e Authentication from both UE and Network side), fake base stations were used as "IMSI catchers" to get the users' IMSIs and to track movement of users[3]. Just like earlier generations, active and passive adversaries can obtain subscriber identity (SIM details) as the IMSI transmission happens over the air in plain text in 4G also. However, to enable device privacy, the transmission of International Mobile Equipment Identity (IMEI), (which is the hardware equipment identity, can be found by dialing *#06# in the dial pad) in plaintext is restricted over LTE networks by the 3GPP.

Due to its importance, LTE motivates various attacks that range from denial-of-service through jamming [25, 20, 29, 31], by Aziz et al.(2015), Jover et al.(2013),

Lichtman et al.(2016), to downgrade attacks that enforce a more insecure communication standard [17, 33, 3], to identification and localization attacks that reveal the presence of a user within a radio cell [3] by Shaik et al.(2015). The majority of these attacks set a focus on either the physical layer (layer one) or the network layer (layer-three) of the protocol stack and leave a blind spot in-between on the second layer (data link layer), which ranges from the LTE Medium Access Control (MAC) to the Packet Data Convergence Protocol (PDCP). Recently, Rupprecht et al. [11] presented the first collection of attacks on layer two. Besides an active DNS redirection attack (called aLTER), their work also introduces an identity mapping that enables website fingerprinting on encrypted LTE traffic. Their results predict severe consequences for the privacy of users. An adversary with the ability to fingerprint encrypted traffic, either actively [18, 19] by Wang et al.(2005,2007) or passively Levine et al.(2004)[24], is often in a position to recover sensitive information about a user. Privacy leaks by traffic fingerprinting attacks first emerged when Cheng et al. [28] in 1998 found out that—even without access to the encrypted payload of a transmission—we can distinguish websites just from meta information like the number of packets sent over time. Since then, advances in classification techniques [26, 27], models of the user behavior Panchenko et al.(2018)[23], and modern machine-learning algorithms [30, 16] by Rimmer et al.(2018) helped to improve the success of fingerprinting attacks in more challenging scenarios. Systems with additional security features, e. g., the Tor anonymity network [34], limit the threat of traffic analysis. Nevertheless, there is a large body of powerful attacks that also succeed in the context of Tor [32, 21, 22, 34, 35]. While this area of research emerged to a state where we find advanced attack concepts, there is a lot of opportunity and yet-to-be-done areas in fingerprinting attacks on LTE layer-three traffic.

The hierarchical evolution of mobile communication generations specified by 3GPP have

Not only improvements in functionality but also improved security aspects is taken care of by 3GPP(Third Generation Partnership Project)'s hierarchial

evolution of generation of different mobile communication from 1G to 4G, which is discussed in detail in the subsequent sections (section 3.1).

Usage of well-analyzed algorithms for stronger cryptography along with "mutual authentication" is introduced by 3G specifications. Further strengthening of signaling protocols (in the control plane) is done in LTE specifications by requiring authentication and encryption ("ciphering" in 3GPP terminology) in more protocol layers than was previously required. Consequently, the LTE specifications strengthen privacy and availability guarantees to mobile users. Previously known attacks, such as the ability to track user movement or theft of user data are difficult in LTE.[3]

That's where a passive protocol analyzing technique is required to be researched for the sake of device/user identification in 4g LTE networks, which is essentially the scope this research work.

Even though 4g LTE overcomes many security issues of previous standards (2G/3G), several attacks have been discussed & performed on physical layer [1, 10] and datalink layer[3, 13]. **In the network layer we can investigate the impact of fingerprinting attacks on encrypted LTE/4G layer-three (Control Plane) traffic.**

Chapter 3

Overview of LTE

3.1 4G and Mobile Communication Generations

LTE, also referred to as 4G, is the wireless broadband communication standard for cellular and data terminals providing higher data rates. It was designed by the Third Generation Partnership Project (3GPP). 3GPP is an alliance of telecommunication standards associations, known as the organizational partners. 3GPP alliance covers cellular telecommunication network technologies, like radio access network (RAN), services, systems aspects, core network and terminals by providing complete system specifications in these areas. LTE is evolved from previous generation of mobile network known as the Universal Mobile Telecommunication System (UMTS)/High Speed Packet Access (HSPA) [3rd Generation], which in turn evolved from the Global System for Mobile Communications (GSM) and Enhanced Data rates for GSM Evolution (EDGE)[2nd generation]. The specifications for LTE are specified by release 8 of 3GPP project. It was completed in December 2008 and this has been the basis for the starting point of LTE. LTE specifications are very stable, with the added benefit of enhancements having been introduced in all successive 3GPP releases. Mobile or cellular data usage demands of users have increased exponentially in the previous years. The demand has increased in terms of both bandwidth and quality of services. As a result of increasing demands second generation (2G) and third generation (3G) networks data traffic was in-

creased drastically and started to become congested. To cater this increasing user data demands technologies like HSPA, evolution of UMTS were introduced, but could not satisfy the users needs. Keeping in view these increasing demands, high quality of service requirements and shifting of broadband services to more handy mobiles were the main requirement for the evolution of LTE. High data rate, low latency and packet optimized radio access technology supporting variable bandwidth implements, packet-switched (IP based) network architecture has been designed with the goal to support flawless mobility and better quality of service is the main aim of LTE.

The first generation of mobile communication was introduced in the 1980s. It was based on the analog radio system technology and the users could only make phone calls. No SMS or data services were available in the first generation. However, with the introduction of 2G in the 1990s, mobile communication was digitized. So in the 2G SMS and data services were offered along with the voice. The major 2G technologies were: GSM/GPRS, EDGE, CDMAOne, PDC, iDEN, IS-136 and D-AMPS. Furthermore, to cater the increasing user demands third 3G was introduced. 3G offered much faster data transfer and ability to transfer high amounts of data. So 3G supported services like video call, file sharing, internet surfing, watching TV online and playing online games.

Next the 4G or the LTE was introduced in 2008 when 3GPP finalized its release 8. The LTE is purely IP based. Both real time services and data communication services are carried by the IP protocol. An IP address is assigned to the user equipment as soon as it turns on and released when it is switched off. LTE is based on Orthogonal Frequency Division Multiple Access (OFDMA), supports higher order modulation scheme (up to 256QAM), large bandwidths (up to 20 MHz) and high data rates can be achieved (Multiple Antenna for Transmitting and receiving) in the downlink (up to 4x4 MIMO). Using spatial multiplexing on the transport channel the highest theoretical peak data rate is 75 Mbps in the uplink, and in the downlink the rate can be as high as 300 Mbps.

3.2 4G LTE Architecture

LTE data transmission through internet is based on packet-switching. In such architecture, sender divides any sort of data - Email, web pages, Video streaming - into small packets of equal size, each of which carries 'address label' having the destination. When packets are sent over the network, at every 'node' they are directed towards the destination where they reassemble and delivered to the recipient.

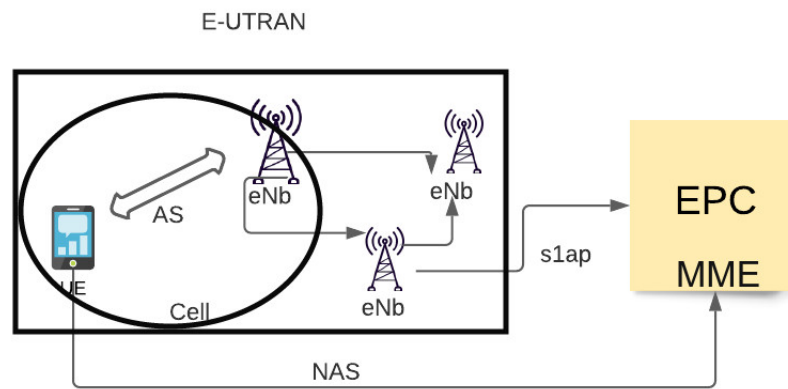


Figure 3.1: LTE system architecture

The major components of LTE are User Equipment (UE), Evolved Universal Terrestrial Access Network (E-UTRAN), and Evolved Packet Core (EPC). In technical or 3GPP terms the whole system is termed as Evolved Packet System (EPS). The overall network architecture is shown in figure 3.1, where the core network EPC consists of many logical nodes and the access network E-UTRAN consists of Evolved NodeB (eNB) or the base station which connects to the UEs. The LTE access network architecture is a flat architecture based on network of base stations (eNB). The eNBs are usually inter-connected via X2-interface. Whereas, S1-interface connection is used between the eNB and core network. In LTE there is no centralized control or controller.

To reduce the time required for handover (Switching connection from one eNB to other without losing the connectivity) and increase the connectivity speed, the control plane or the intelligence of the base stations (eNBs) are made

distributed. For ideal handovers very small time delay is required for real-time services where end-users will definitely end calls if the handover takes too long, as it becomes impractical.

We will be focusing on 3 components. First, we have the User Equipment (UE) which is basically the end device providing services to the user. Then we have the eNodeB which acts like a base station and performs radio resource management, user data encryption, paging messages and so on. And finally we have the Evolved Packet Core of the network which is responsible for authentication, mobility management(MME) and forwarding user data.

User Equipment- The actual mobile device is called UE in 3GPP terminology, refers to the ground based modem which is actually, a smartphone. It has a USIM (Universal Subscriber Identity Module), which contains the IMSI and keeps the corresponding authentication credentials. This IMSI is useful for identifying the unique LTE user (generally referred to as “subscriber” in 3GPP terminology).

The USIM is just equal to the SIM card which we use in a GSM device. Subscriber related information like 10 digit number, IMSI and service recharge details are stored here, also for authentication and ciphering, implementation of security functions are there on the user side in LTE devices. In other words, the ground based modem in end-user’s hand which customer actually buys is termed as UE. The LTE UEs are divided into different categories based on their UL and DL capabilities. These categories allows the eNB to communicate effectively with all the connected UEs. UE radio access capabilities are listed in 3GPP 36.306. In this project our focus will be UE Cat 4 with a maximum data rate of 50Mbps and 150Mbps in the UL and the DL respectively using MIMO configurations. And upto 75Mbps in DL using single input, single output (SISO) configurations.

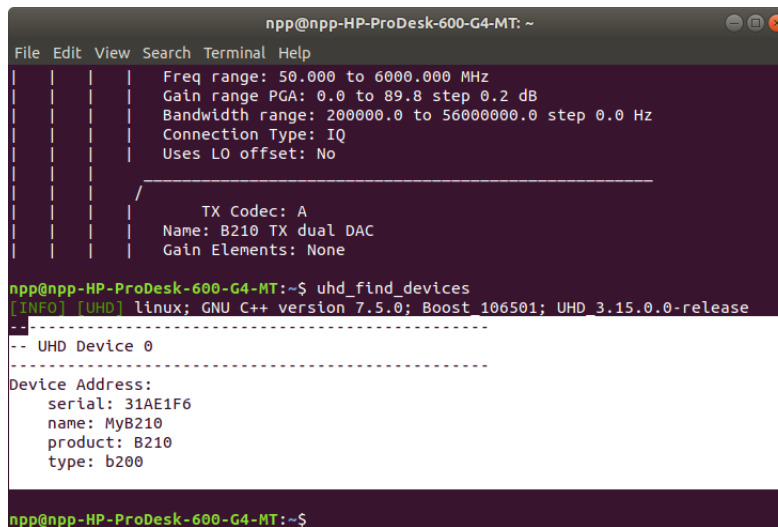
E-UTRAN- E-UTRAN contains base stations(BS in 3G) along with UEs. It supervises the radio connection with the UE and comes up with communi-

cation between the UE and EPC. Which was called base station in 2G/3G is technically (in 3GPP terminology) referred as “evolved NodeB (eNodeB)” in LTE.

Access part of LTE is termed as E-UTRAN. It handles the radio communication between UE and the EPC both Uplink (UL) and Downlink (DL) communication. Apart from this, eNB also transmit signaling messages like handover commands. E-UTRAN consists of eNB base stations to control the mobile units/UE in different cells.

An overall E-UTRAN architecture is shown in fig-3.1. The eNBs are interconnected through x2 interface with each other and there is no centralized controller unlike in the case of earlier generations(BS) in E-UTRAN. So in short, only the EUTRAN is completely responsible for all connectivity and resource management and header compression and other radio related resources to the EPC.

USRP B210 is configured with UHD (USRP Hardware Driver) which is the open-source software driver and API for the Universal Software Radio Peripheral (USRP) SDR platform. The lab setup is shown below in fig-3.2.



```
npp@npp-HP-ProDesk-600-G4-MT: ~
File Edit View Search Terminal Help
-----
Freq range: 50.000 to 6000.000 MHz
Gain range PGA: 0.0 to 89.8 step 0.2 dB
Bandwidth range: 200000.0 to 56000000.0 step 0.0 Hz
Connection Type: IQ
Uses LO offset: No

-----
TX Codec: A
Name: B210 TX dual DAC
Gain Elements: None

npp@npp-HP-ProDesk-600-G4-MT:~$ uhd_find_devices
[INFO] [UHD] linux; GNU C++ version 7.5.0; Boost 106501; UHD 3.15.0.0-release
-----
-- UHD Device 0
-----
Device Address:
  serial: 31AE1F6
  name: MyB210
  product: B210
  type: b200

npp@npp-HP-ProDesk-600-G4-MT:~$
```

Figure 3.2: USRP B210 Running on Ubuntu 18.04 machine

Access Stratum (AS) is used by the eNodeB for exchanging signaling messages with its UEs, which are a set of network protocols for access control. Radio Resource control (RRC) protocol is included in these AS control signalling

messages. Other functions of eNodeB include physical layer data connectivity, over-the-air security, paging and handovers while mobility[3]. An interface named S1 (Application protocol,s1ap) is the connection between each eNodeB and the LTE core (EPC).

3.3 EPC

EPC is the latest development in the core network architecture of the 3GPP standards. In EPC based network Internet Protocol (IP) is used for all services i.e. voice, data, SMS etc. It is the evolution of the packet-switched architecture used in GPRS and UMTS. In EPC system user plane and the data plane have been separated, thus allow more flexibility and scalability. The user and data plane have been separated in a basic LTE system. The UE is connected to the EPC through E-UTRAN (LTE access network). The EPC consists of four basic elements the serving gateway (Serving GW), the public data network gateway (PDN GW), the Mobile Management Entity (MME) and the Home Subscriber Server (HSS). The gateways (Serving GW and PDN GW) defines the user plane and are responsible for transporting the IP data traffic between the UE and the external networks. Whereas, MME and HSS deals with the control plane. MME handles the control signalling related to mobility and security of LTE network. It is responsible for tracking and paging of UE in idle-mode. The HSS is a database of subscribers apart from this it also provides mobility management functions like call and session setup, user authentication and access authorization. The LTE core network which is the EPC (Evolved packet Core) is installed by srsLTE and running on open source, shown in the below figure 3.3.

```
npp@npp-HP-ProDesk-600-G4-MT:~/srsLTE/srsepc$ sudo srsepc epc.conf
[sudo] password for npp:
Built in Release mode using commit d045213fb on branch HEAD.

--- Software Radio Systems EPC ---

Reading configuration file epc.conf...
Warning parsing mme.integrity_algo:EIA0 - EIA0 will not supported by UEs use EIA
1 or EIA2
HSS Initialized.
MME S11 Initialized
MME GTP-C Initialized
MME Initialized. MCC: 0xf404, MNC: 0xff49
SPGW GTP-U Initialized.
SPGW S11 Initialized.
SP-GW Initialized.
```

Figure 3.3: LTE core Running on Ubuntu 18.04 machine

MME in EPC-

Mobility Management Entity (MME) plays the most important part in the SAE (System Architecture Evolution). The Evolved Packet Core (EPC)'s main functionalities are handled by MME. MME is the main signaling node in the EPC. LTE MME is responsible for

- Authentication of the mobile device, and initiating paging request when hexagonal cell changes.
- Retaining location information according to TAC (Tracking Area Code) at the tracking area or cell level
- S1-MME is the interface through which MME connects to the evolved nodeB (eNB) and through the S1-U interface connects to S-GW.
- Increasing signaling load in the network is met by connecting multiple MMEs together. MME also plays a pivotal role in handover (X2 interface) signaling between LTE and 2G/3G networks.

Non-Access Stratum (NAS), is the set of protocols between UE and MME as depicted in fig 3.1.

3.4 The OSI Layers Overview

In LTE, UE and the eNodeB has a protocol stack which is shown in below figure 3.4.

The highest layer of the control plane is the NAS layer. NAS layer does the managing of the mobility, management and establishment of IP connectivity. It communicates between the UE and MME. During the initial connection setup, it transmits the USIM information to the MME. Next layer in the stack is the RRC defined in 3GPP TS 36.331 document [4]. The RRC layer is responsible for radio connection establishment and release functions, radio bearer establishment, mobility management, paging and broadcasting of the system information. After RRC is the PDCP layer, as we can see in the figure 3.4, it handles packets

from both the control plane (RRC) and data plane (IP). Header compression, ciphering and integrity protection tasks are performed at this layer. Further on, there is RLC layer, specifications are defined in 3GPP TS 36.322. The main responsibilities of this layer are segmentation, reassembly and re-transmission of lost packets. Next in the stack is the medium access (MAC) layer, basic purpose of this layer is to handle the channel access procedures or the random access procedure. It performs the multiplexing and demultiplexing of logical and transport channels. Tasks like scheduling requests, buffer status reporting, and hybrid automatic repeat request (HARQ) are also handled by the MAC layer.

The lowest layer in the stack is the PHY layer, it transmits all the data from the MAC layer to the air interface (Via Antenna). The main tasks performed are link adaptation, power control and cell search.

- **Physical Layer-** The lowest of OSI layers, Physical layer responsible for transmitting information over the air interface, controlling the transmission power of the physical channel and implements encoding and modulation schemes. In the UE protocol stack, MAC (Medium Access Control) protocol adjusts the these physical layer parameters by CQI (Channel Quality Indicator) which is regularly monitored in this 1st sublayer of second layer.
- **Data Link Layer-** The datalink layer extends moving of data into and out of a physical link in a network providing additional services towards the upper layer and mechanisms for reliability, security and integrity.

Medium Access Control(MAC)- MAC is the first sublayer of the second layer which is the link between Physical layer and Data-Link layer. The access to Radio resources of LTE is managed by the MAC layer protocols. For this MAC scheduling is done through assigning every UE a unique identity i.e RNTI (Radio Network Temporary Identifier). An unencrypted

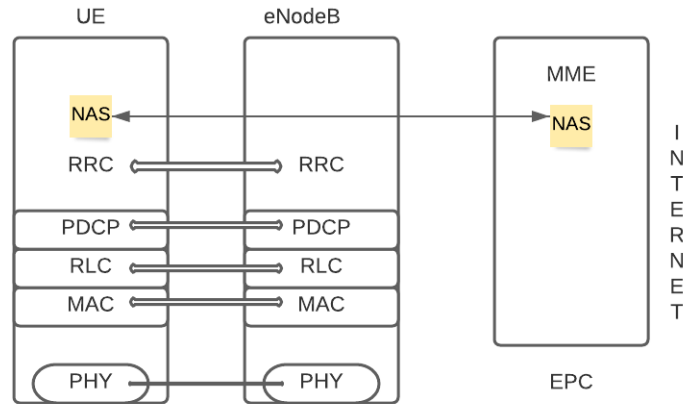


Figure 3.4: Architecture of LTE with layer stacks

Random Access Response (RAR) is exchanged by the UE to the eNodeB via a Random Access preamble (RAP)[11]. The RNTI is the matching between MAC layer of UE and eNodeB, which determines the available Radio Resources and finally transmission takes place in this procedure (signalling information is given to the UE).

Radio Link Control(RLC)- It is the second sublayer of Data-link layer. There are 3 modes of operation on which RLC operates: Transparent Mode (TM), Unacknowledged Mode (UM), and Acknowledged Mode (AM). [12] The responsibilities of RLC layer can be listed as-

- Transfer of Upper Layer (layer 7 to 3) packet data units (PDUs)
- In the acknowledged data transfer mode, error correction through ARQ
- In both Acknowledged and Unacknowledged mode, service data units (SDUs) and concatenated, segmented and reassembled in RLC layer.

Re-segmentation and re-ordering of RLC data PDUs, duplicate packet detection, discarding some SDUs, RLC re-establishment are some of the responsibilities of RLC layer. [12]

Each time a user/UE tries to send or receive signal or data through the network 'radio layer connection establishment is triggered as the radio

packets contain both their radio layer identity (RNTI) and the TMSI of the overlying Non-Access Stratum.

Packet Data Convergence Protocol(PDCP)-

The task of encryption and integrity protection (for control plane messages) to the overlying Radio Resource Control(RRC) layer is done by PDCP layer. PDCP protocol transfers encrypted user plane data to upper-level protocols like IP (Internet Protocol). The PDCP layer is the first to apply encryption algorithms, hence, we can directly read the payload (user data) and header information of all packets below this sublayer (i.e layer 2 or below), within the data link layer. Which means that passively analysis of the meta information of layer two transmissions, e.g, the PDCP length of a packet.[11] The lack of data integrity of LTE in this sublayer can be exploited, which is the reason behind Fingerprinting of either User plane data or control plane data, subsequently leads to Website Fingerprinting and Device fingerprinting.

- Network Layer- There are 3 sublayer is Network layer- NAS,RRC and IP. System Information Blocks (SIBs) are broadcasted in the RRC sublayer. Both Access Stratum and Non Access Stratum SIBs which are exchanged during Attach procedure is done in this sublayer. The highest stratum of the control plane between the user equipment (UE) and MME is the non-access stratum (NAS) protocols. The mobility of the UE and the session management procedures to establish and maintain IP connectivity between the UE and a PDN GW (packet data network gateway all of these done through NAS layer protocol.[12]

3.5 LTE Registration Procedure & CallFlow

In an LTE based system when a UE tries to connect to the available network, the process is generally termed as handshake. The whole handshake process

can be divided into two major phases. 1. UE and eNB RRC connection setup
2. Authentication and security setup These two major steps can be divided further into minor steps as explained below.

In this first step the UE detects the available eNB base-stations and try to establish a radio link connection with the base-station. Following major steps are performed in this phase i.e. synchronization, system information acquisition, random access procedure and RRC connection setup. In the first step the UE gets the frequency and time synchronization by the decoding the primary and secondary synchronization signals (PSS and SSS), transmitted by the eNB after every 5ms. Sub frame level synchronization and physical layer cell identity is attained through this step [8]. In the next step the UE gets the network specification information which is stored in the master information block (MIB) and system information blocks (SIBs) messages. MIB is transmitted on PBCH which contains the system bandwidth and PHICH format information. Next the UE decodes the downlink control information (DCI) message transmitted on PDCCH. DCI contains the positions of SIB messages. The two SIB messages, SIB1 and SIB2 contains the cell access information, control and shared channel configuration and random access (RA) information necessary for the RA procedure. Once above steps are performed, the UE know has the information of the available network. At the this time UE tries to get the access to the network by initiating the RA procedure and sends the RA preamble over the PRACH channel to the eNB. After sending RA preamble UE gets the response from the eNodeB called the random access response (RAR, or MSG2). In RAR UE is assigned with a temporary identity e.g. temporary C-RNTI along with the information of UL timing adjustment and the slot information for the next message to be transmitted on PUSCH.

The control signal flow for UE registration steps are shown in the below figure.

After the registration procedure the network initiates the authentication procedure by sending a challenge in the form of authentication request which contains parameters like RAND and sequence number. If the UE succeeds it

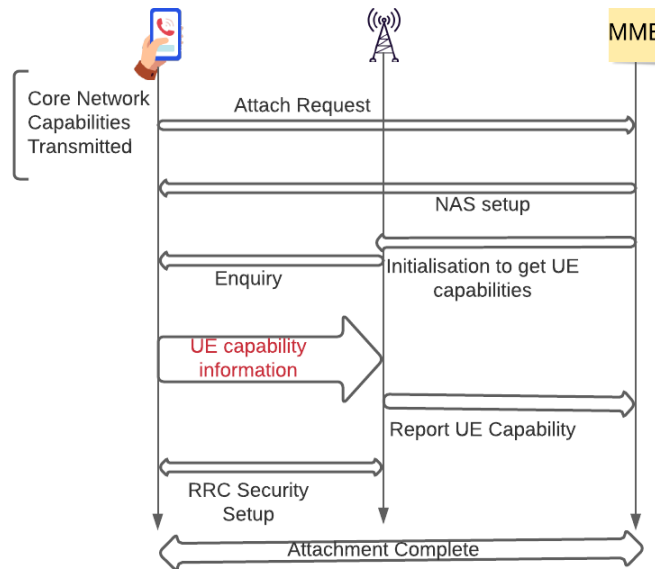


Figure 3.5: UE Attachment with Core network

sends authentication response with the unique value RES otherwise it sends authentication reject with cause of either MAC or SYNC failure. After the authentication procedure network and UE need to reach to an agreement regarding the keys and algorithm to be used in encryption which is done by security mode command. If the UE needs to de register and want to logout from all the resources allocated to it it sends deregistration request to the network and finally if UE due to roaming or any other reason lost the temporary identity information it initiates the identity request to the network. Next is the final step of the phase 1 where the UE sends the RRC connection request or the MSG3 which includes the temporary C-RNTI number and the subscriber details. In reply the eNB sends the RRC connection setup or MSG4 on PDSCH and assigns the available UL resources to the UE. On successful reception of MSG4 UE send the RRC connection setup complete message which contains the NAS service request message explained in the following section

The second phase of the two major steps are performed; NAS authentication and security setup and RRC connection re-configurations. Once the RRC connection is setup between the eNB and the UE, the authentication step is performed. UE and eNB authenticate each other using the defined algorithm

in 3GPP TS 35.206 V4.0.0 (2001-04). It is important to mention that in the authentication phase communication is done between the UE and the MME of the EPC via eNB.

After authentication, a security mode command is transmitted by the MME via eNB on PDSCH in order to activate integrity protection and ciphering. The UE after verifying the security mode command responds with security mode complete message by applying integrity protection and ciphering to this message. Once the security algorithms are implemented the eNodeB transmits a RRC connection reconfiguration message, to establish SRBs and the data radio bearer (DRB). The UE responds with a RRC connection reconfiguration complete message, allowing for user data transfer using a DRB. Details about the security aspects are defined in the series 33 of 3GPP specifications. [15]

With 4G there have been various developments made in various important procedures compared to 3G. The call flow is quite complex as compared to 2G/3G. In 4G the initial attach message has been replaced by the registration request message. After the registration procedure the network initiates the authentication procedure by sending a challenge in the form of an authentication request which contains parameters like RAND and sequence number. If the UE succeeds it sends an authentication response with the unique value RES; otherwise it sends an authentication reject with a cause of either MAC or SYNC failure. After the authentication procedure, the network and UE need to reach an agreement regarding the keys and algorithm to be used in encryption, which is done by the security mode command. If the UE needs to deregister and wants to log out from all the resources allocated to it, it sends a deregistration request to the network and finally, if the UE, due to roaming or any other reason, loses the temporary identity information, it initiates the identity request to the network.

Chapter 4

Device FingerPrinting

4.1 Problem Statement

-

As discussed earlier there is no mutual authentication between 2G system (device) and the network, which led to possibility of an attack by setting up "fake base station" and persuade permissible mobile devices to get connected to it. Use of TMSI was introduced to stop the exposure of IMSI (as in 2G it is transmitted over the air unencryptedly), but still in the absence of mutual authentication (i.e Authentication from both UE and Network side), fake base stations were used as "IMSI catchers" to get the users' IMSIs and to track movement of users[3].

However, in 5G system IMSI is encrypted, and in place of IMSI, SUPI (Subscription Permanent Identifier) or SUCI (Subscription Permanent Identifier) is transmitted which is not in plain text.

So for device identification, layer-3 RRC control signal message analysis and extracting important feature/parameters to track or identify the nature of a connected device is the only way forward. As future work, this could include analysis of layer-2 User plane traffic and classification of Data Traffic for Website FingerPrinting which will lead to what kind of device along with what kind of website is being browsed by the connected UE. Making an Embedded Device FingerPrinting and Website FingerPrinting for the improved tracking

of UEs in an LTE network is the future possibility. As the call flow and initial attach procedure in both 4G and 5G is exactly same, this work must easily get extended to next generation.

4.2 The Classification Model

4.2.1 Understanding UE Capabilities

The term device-type in our work speaks to device specifics such as the combination of the maker, model, software and the application(s) on the device. Based on these particular implementations, we found that it is conceivable to identify a device-type and its corresponding features. Device identification is based on the differential investigation of the capabilities that are obtained from a UE. The first level recognizes either voice centric or data centric gadget and the second level separates between a Mobile phone with VoLTE enabled and a mobile phone with VoLTE disabled. The third level determines the device's Operating System and the fourth level distinguishes device manufacturer or the baseband vendor. By utilizing our eNodeB setup, we acquire both the core network and radio access capabilities from the test devices and analyze them. In specific, UE initiate a registration procedure with our eNodeB and we extract the capabilities.

4.2.2 Block Diagram of Classification Model

Every UE has several capabilities (by the manufacturer) for various LTE services and operations. Core network capabilities (like non-radio capabilities, hardware or chip level features security algorithms etc) and radio access capabilities (like radio aspects of mobile, supported frequency bands, transmit and receive speeds (samples per second), modulation scheme etc) are exercised/reported between the MME and the eNodeB respectively during the attach procedure/registration.

The UE capabilities or device fingerprints identifying, classifying and ex-

tracting any useful information will lead to various possibilities in the LTE datalink layer.

As shown in the below block diagram, fig- 4.1, we have classified all the 4G LTE UEs which tries to get attached to our network into 4 levels. Whenever a device send attach request, we shall be able to classify them into these 4 categories. Four layes has been shown in 4 different colours. A parameter of UE's usage settings, either voice centric or data centric will determine the service type of the UE, i.e either a Mobile device or a data centric device (with out any voice capability, eg-jioFi). The 2nd layer would be, if it's a voice centric the VoLTE is enabled or disabled.Which means a Volte disabled device can't use both voice and 4g data service at a time. It has to fall back (CSFB) to 2G for data access while call is ongoing. The 3rd layer distinguishes between the Operating system of mobile device whether android or iOS. The 4th layer may classify the devices among 3 popular baseband processors used. The 4 layer classification is visualised in the below block diagram.

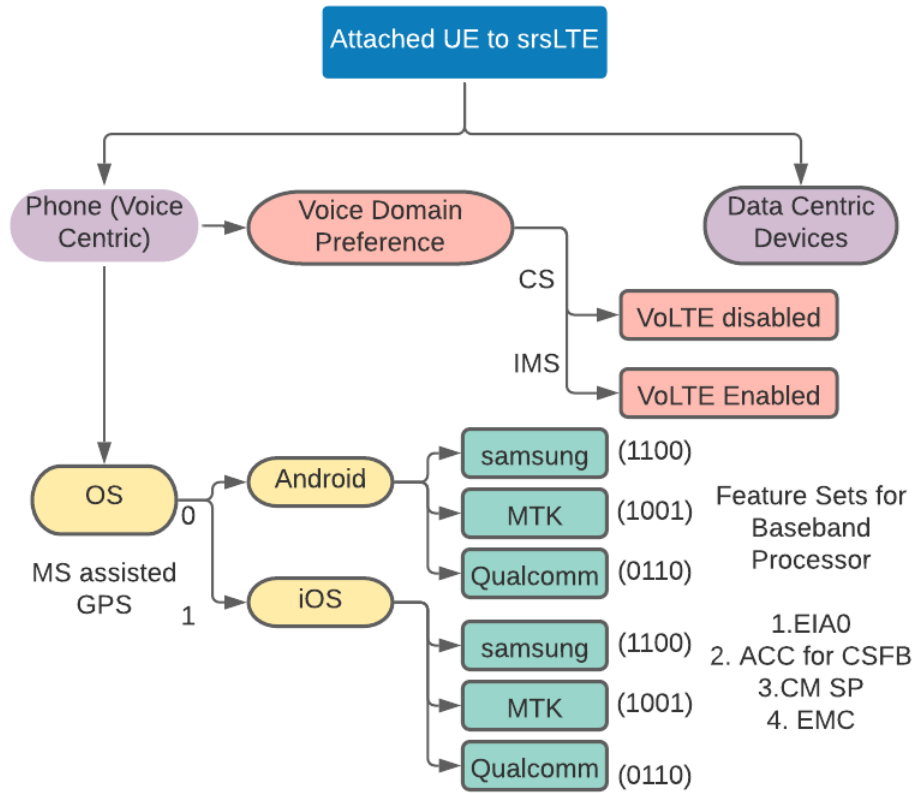


Fig:- Classification Model- Layer-1, Layer-2, Layer-3, Layer=4

Figure 4.1: Flow diagram for 4-Layer Device type identification

Chapter 5

Testbed / Framework-

The overall framework can be depicted in the figure 5.2. srsLTE (Software Radio Systems LTE) provides open source platform to run the LTE architecture. This project has both a software and a hardware part. As the LTE application or the srsLTE is running on a linux based machine and for RF requirements an SDR hardware platform is used. First of all we begin with understanding the features and limitations of SDR technology as the SDR is the motivational technology behind this project. Next we get ourselves familiarize with the LTE technology and all the components in the LTE protocol stack. After getting the understanding of SDR and LTE technology next step was implementation. The srsLTE application was installed in a linux based pc and was used with the selected SDR hardware platform, details are mentioned in section 5.4 and 5.5.

An end to end 4G standalone test bed is build using two open source software is as shown in Figure 5.1 to analyze the call flow in 4G network. The experimental setup consist of one i7 PC using Linux OS (Ubuntu 18.04). In the system open source tool is used for building up an end to end test bed. The open source tool is srsLTE which is the core network consisting of various 4G network function. For the RAN, srsENB is used and commercial mobiles for UE. The detail explanation of the tool and its features will be explained in the subsequent section. The integration of the platform has been done. it supports 4G-AKA which is the enhanced version of EAP-AKA. It also supports exchange of various crucial signalling message between UE, enodeB and the core network.

For integrating the 4G core and the gnodeB it was made sure that the internet protocol(IP) address of AMF and enodeB is same , the tracking area code (TAC) is maintained same for both UE and enodeB. The figure 5.5 shows the wireshark capture of various Signalling messages which is exchanged between UE and the 4G network.

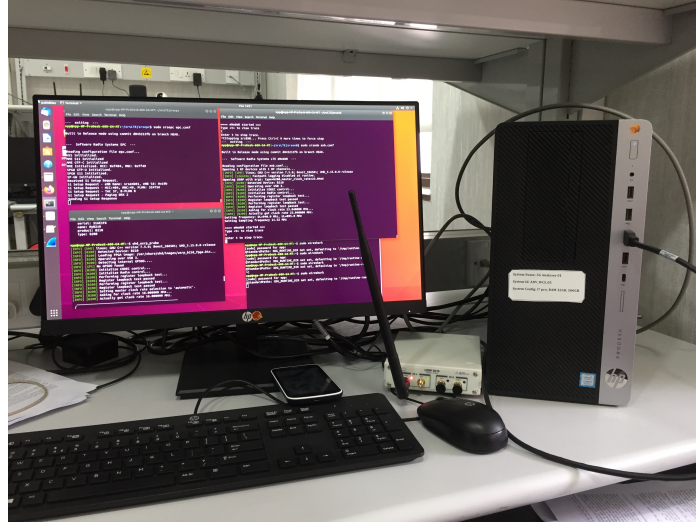


Figure 5.1: Lab Setup Environment

With the help of our testbed we were able to capture various device capabilities as well as pre-authentication messages.

5.1 srsLTE

srsLTE is the open source LTE compliant linux based application developed by software radio systems (SRS), a private limited company in Ireland. srsLTE application includes the functionalities of an LTE eNodeB, EPC and UE. The software application was initially designed according to release 8 of the LTE standard. However, current open source version of the application is LTE release 15 compliant supporting both Frequency Division Duplex (FDD) and Time Division Duplex (TDD) modes.

The application has been tested with different SDR platforms like USRP, bladeRF and limeSDR. The implementation is highly modular and can be modified to specific needs. For commercial purposes the srs sell the commercial

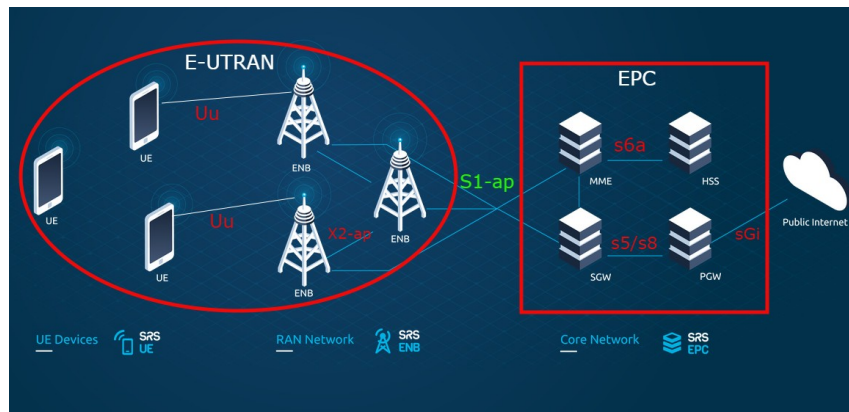


Figure 5.2: LTE software Suit - www.srslte.com

licenses and also sell the proprietary products based on srsLTE code base

5.2 eNodeB

The srsENB is the LTE base station, the current open source code include the FDD configuration, Round Robin MAC scheduler, encryption, Channel Quality Indicator (CQI) feedback support and standard S1AP and GTP-U interfaces to the core network. The srsENB can support maximum of 150 Mbps DL in 20 MHz MIMO TM3/TM4 with commercial UEs. USRP B210 is used as eNodeB, configured from [14],

```

--- Software Radio Systems LTE eNodeB ---
Reading configuration file enb.conf...
Opening 1 RF devices with 1 RF channels...
[INFO] [UHD] linux; GNU C++ version 7.5.0; UHD_3.15.0.0-release
[INFO] [LOGGING] Fastpath logging disabled at runtime.
Opening USRP with args: type=b200, master_clock_rate=23.04e6
[INFO] [B200] Detected Device: B210
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Asking for clock rate 23.040000 MHz...
[INFO] [B200] Actually got clock rate 23.040000 MHz.
Setting frequency: DL=889.0 Mhz, UL=844.0 Mhz
Setting Sampling Frequency 11.52 MHz

==== eNodeB started ====
Type <t> to view trace
t
Enter t to stop trace.
RACH: ttl=2961, preamble=34, offset=1, temp_crnt1=0x46

-----DL-----UL-----
rnti  cqi  ri  mcs  brate  bler  snr  phr  mcs  brate  bler  bsr
46   13.9  0  0.0  328    0%  13.8  0.0  0.100  56.0    0%  0.0

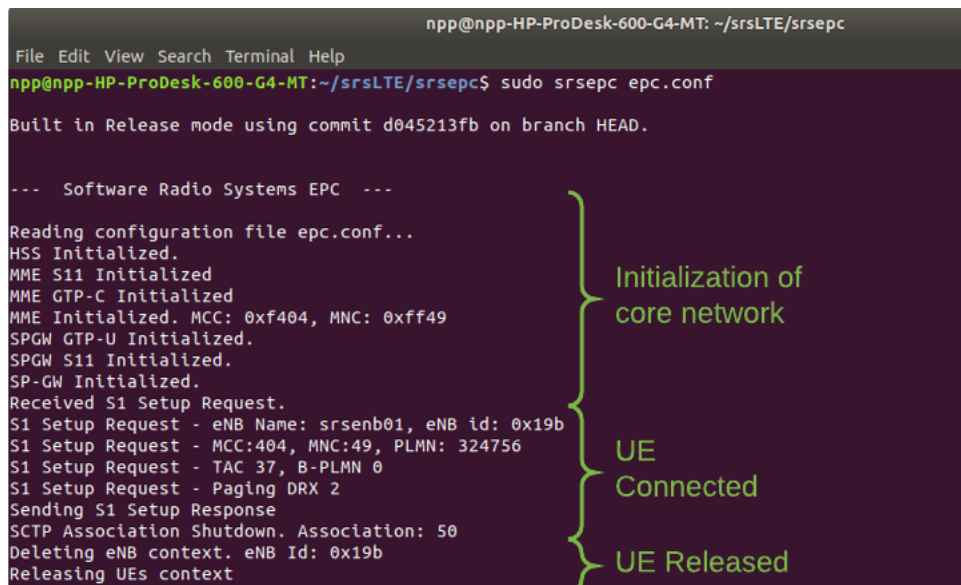
```

Figure 5.3: srsLTE eNB running on Ubuntu18.04

5.3 LTE Core network

srsLTE which is available as open source[14], which is used here to deploy the LTE core network by building it in a Linux system (Ubuntu 18.04 machine is used here).

The srsEPC is the very basic implementation of LTE core network. It includes MME, gateways and HSS containing the user database. The HSS directory is configurable and a user can be added or deleted from the data base. Further more paging support has been added.



```
npp@npp-HP-ProDesk-600-G4-MT: ~/srsLTE/srsepc
File Edit View Search Terminal Help
npp@npp-HP-ProDesk-600-G4-MT:~/srsLTE/srsepc$ sudo srsepc epc.conf
Built in Release mode using commit d045213fb on branch HEAD.

--- Software Radio Systems EPC ---

Reading configuration file epc.conf...
HSS Initialized.
MME S11 Initialized
MME GTP-C Initialized
MME Initialized. MCC: 0xf404, MNC: 0xff49
SPGW GTP-U Initialized.
SPGW S11 Initialized.
SP-GW Initialized.
Received S1 Setup Request.
S1 Setup Request - eNB Name: srsenb01, eNB id: 0x19b
S1 Setup Request - MCC:404, MNC:49, PLMN: 324756
S1 Setup Request - TAC 37, B-PLMN 0
S1 Setup Request - Paging DRX 2
Sending S1 Setup Response
SCTP Association Shutdown. Association: 50
Deleting eNB context. eNB Id: 0x19b
Releasing UEs context
```

Initialization of core network

UE Connected

UE Released

Figure 5.4: srsLTE EPC on Ubuntu18.04

5.4 srsLTE Build and Install

This section discusses the installation steps of srsLTE on a Linux-based PC and supporting software required for interfacing with the SDR platform used. The srsLTE application software has three main components srsepc, srsUE and srseNB. The srsUE and srseNB are the radio parts which require an interface to a SDR transceiver. Whereas, EPC part is software based having modules like MME, SGW, PGW and HSS which are the main components of the LTE core EPC architecture. Before moving on with the srsLTE application installation, it

is necessary to install all the device driver software as per the SDR platforms used. Below section give a brief overview of supporting software required for the project.

First we started with installing the UHD driver for the USRP B210. We used the UHD driver version 3.9 LTS as it is recommended by the srsLTE team. To verify the successful installation "UHD find devices" command is used to verify if the driver detects the attached USRP as shown.

After installing the required supporting software, the srsLTE code was cloned and build. It is pertinent to mention that it is always recommended to rebuild the srsLTE application if any new supporting software is installed. The application was installed on linux 18.04 based machine. Once the software installation is done the application is run by using the default configurations. First run the "sudo epc" on 1st machine to initialize the core network. Once the epc is running, execute "sudo srseNB" command on the same machine but in another console to start the srseNB. The eNB will start and will connect to the srsepc.

5.5 eNB and EPC configuration

The EPC configurations are stored in two main files epc.conf containing the general configurations for the MME, HSS and GW parameters. Second file is the "user db.csv" which contains the users information (IMSI, authentication algorithms, K, OP or OPc, etc) and is used by the HSS. In this project lab tests are conducted using the default EPC configurations.

The default eNB configurations can be found in "enb.conf.example" file. It includes the eNB configurations like eNB id, cell id, MCC, MNC etc. Next it has the supporting configurations file section containing sib.conf file to configure the system information blocks (SIBs), rr.conf for radio resource configurations and drb.conf file for data bearers configurations. RF configuration sections parameters are used to specify operating frequencies, transmit/ receive power levels and device antenna modes. Enb.conf also has the log file configuration option and PCAP options as described for UE. The UL and DL MCS values

can be specified in the scheduler section of the enb.conf. Lastly there is a section defining the expert options. It is important to mention while running the eNB with PRB 6 change the "prach freq offset" value from 2 to 0 in sib.conf file. And as mentioned in above section all the configurations in the config file can be edited using the command line or by changing the configurations files in the "/srslte/srsenb" folder and execute using the "sudo srsenb enb.conf" command.

5.6 Wireshark Logs

Connection of UE to the eNodeB and the "UE attachment" is captured on wireshark, which is an open source tool to observe the data/control flow in the network in fig 5.5.

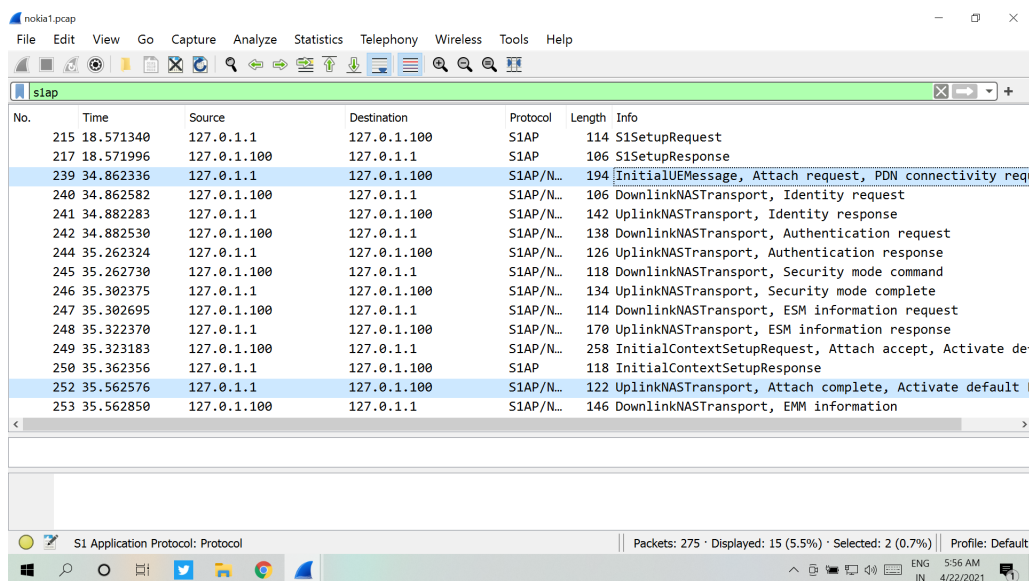


Figure 5.5: Wireshark capture with s1ap filter

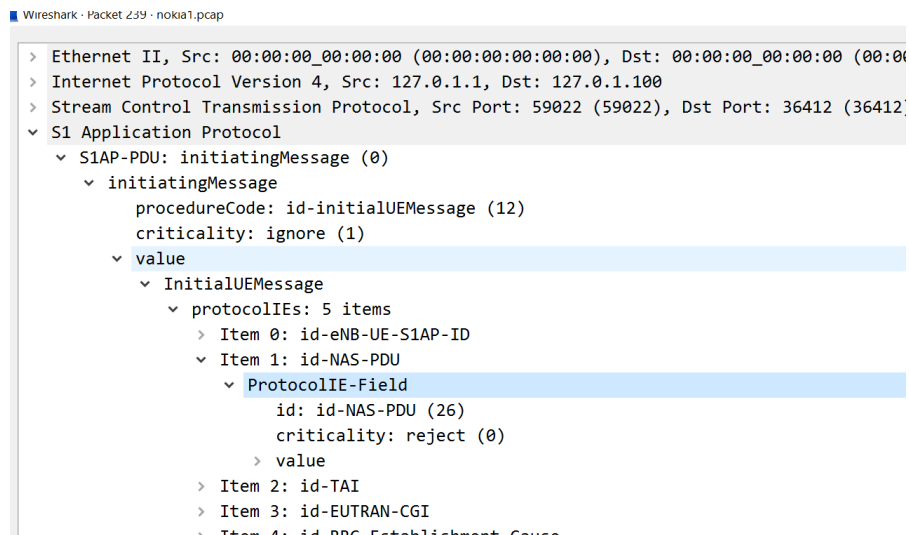
After applying the s1ap filter in the captured logs of UE attachment to srsLTE, we can see the attach procedure from initial attach request to attach complete which are the control plane messages in s1-ap(i.e the interface between eNB and MME; fig-5.2) filter.

Chapter 6

Results

Device capabilities can be sub-divided into two categories ie. core network and radio capabilities. In the below subsection a detailed analysis of crucial device capabilities is listed out that are exchanged during the registration procedure, along with the device capabilities we have listed out various NAS signalling messages which is not integrity or security protected.

As discussed earlier, during the NAS-PDU in the s1-ap interface, we can find many device capabilities being transmitted. The wireshark logs can capture those capabilities for further analysis. Out of the many parameters there



```
Wireshark - Packet 239 - nokia1.pcap
> Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 127.0.1.1, Dst: 127.0.1.100
> Stream Control Transmission Protocol, Src Port: 59022 (59022), Dst Port: 36412 (36412)
v S1 Application Protocol
  v S1AP-PDU: initiatingMessage (0)
    v initiatingMessage
      procedureCode: id-initialUEMessage (12)
      criticality: ignore (1)
      v value
        v InitialUEMessage
          v protocolIEs: 5 items
            > Item 0: id-eNB-UE-S1AP-ID
            v Item 1: id-NAS-PDU
              v ProtocolIE-Field
                id: id-NAS-PDU (26)
                criticality: reject (0)
                v value
            > Item 2: id-TAI
            > Item 3: id-EUTRAN-CGI
            v Item 4: id-RRC-Establishment-Cause
```

Figure 6.1: NAS-PDU in s1ap Filter

are certain important parameters which are used here for various levels of Device Identification (ref- fig 4.1) The parameters are (also mentioned in fig 4.1)

- 1. EIA0
- 2. Access Class Control for CSFB
- 3. CM Service Prompt
- 4. Extended Measurement Capability
- 5. MS Assisted GPS
- 6. UE's Usage Settings
- 7. Voice Domain Preference for E-UTRAN.

Whenever a device tries to get connected to the core network, the MME will check for the authentication credentials from HSS (the master database), then the device will get connected. But even if the device is not authorized by the HSS, due to the transmission of initial attach request message itself, we can see the logs in wireshark and this "initial attach request" can provide us with the NAS-PDU (fig-6.1), eventually the device parameters are exposed.

We can save the packet captures of wireshark (called .pcap files) for each device and analysis of those has led to our classification which is discussed in subsequent sections.

But, before classification we have parsed these pcap files through matlab functions to identify the parameter 'strings' and hence recognition of device feature is done. The parsing for each device and what is the exact parameter string can be seen in the Appendix section, where the matlab codes are shown.

The string identification done such a way that the matlab command window waits for a device to extract the features then show them graphically, as shown in fig-6.2.

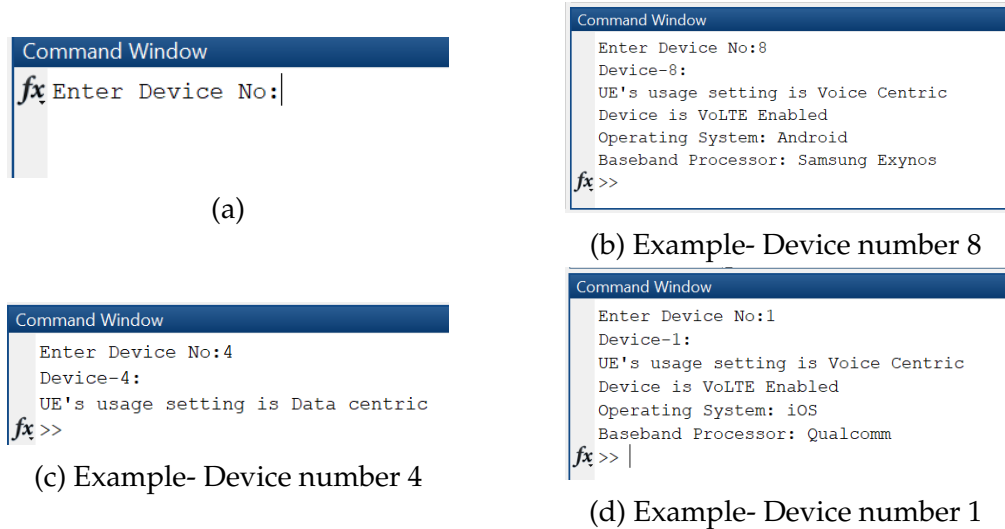


Figure 6.2: Device info in Matlab command window

6.1 Classification Results

Based on the parameter extracted from each device (as shown in fig-6.2), we have classified all the available devices in 4 different level and the results are shown.

6.1.1 Level-1

As explained in the Block diagram (fig- 4.1), the level-1 classification would be differentiate between a Phone which is voice centric or a Data centric device where voice capabilities/codecs are not supported. fig-6.3 shows the wireshark capture for the above mentioned parameter.

- > Supported Codec List - Supported Codecs
- ▼ Voice Domain Preference and UE's Usage Setting
 - Element ID: 0x5d
 - Length: 1
 - 0000 0... = Spare bit(s): 0
 -0.. = UE's usage setting: Voice centric
 -00 = Voice domain preference for E-UTRAN: CS Voice only (0)
- ▼ GUTI type - Old GUTI type
 - 1110 = Element ID: 0xe-

Figure 6.3: Parameter- Voice Domain Preference and UE's Usage Settings

Using this parameter the Later-1 classification is been done by capturing

different devices (Mobile handsets for Voice Centric and Hotspot device for Data Centric) and the pcap captures are accessed in MATLAB to classify them according to the exposed parameter, shown in fig-6.4.

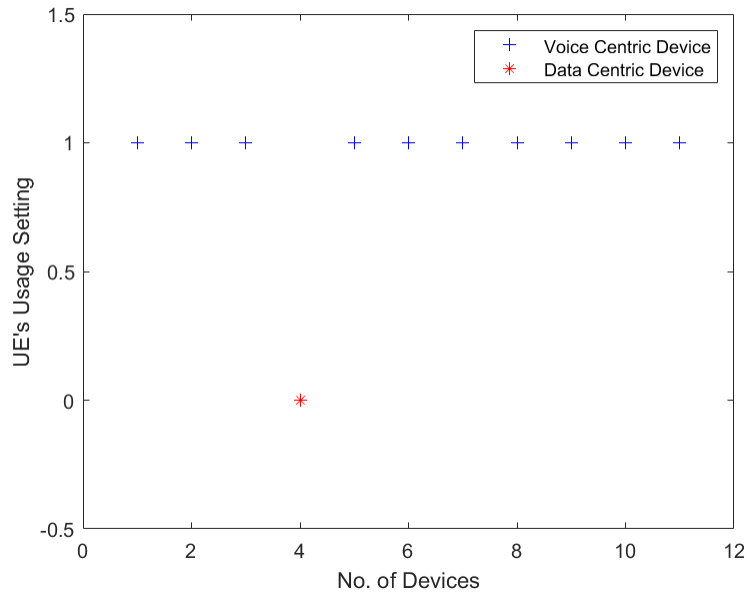


Figure 6.4: Classification- Layer1

6.1.2 Level-2

The level-2 classification would be differentiate between a Mobile phone with VoLTE enabled and a mobile phone with VoLTE disabled. fig-6.3 shows the wireshark capture for parameter, "Voice domain preference for E-UTRAN. For VoLTE disabled devices, it is CS Voice only, i.e the phone will fall back to Circuit Switched mode every time 4G data bearer is trying to established.

Using this parameter the Later-2 classification is been done by capturing different devices (One device is 'CS Voice only', others are 'IMS PS Voice preferred, CS Voice Secondary) and the pcap captures are accessed in MATLAB to classify them according to the exposed parameter, shown in fig-6.5.

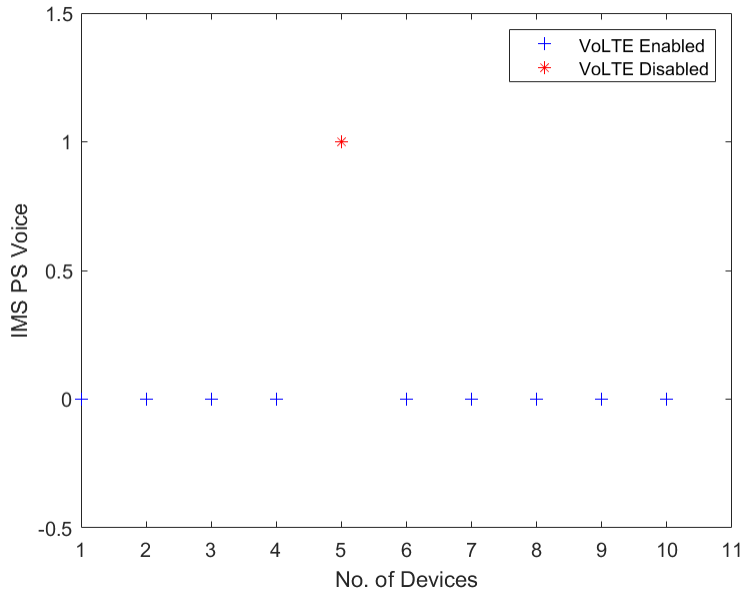


Figure 6.5: Classification- Layer2

6.1.3 Level-3

The level-3 classification would be differentiate between a Phone’s Operating system whether Android Device or iOS device is trying to get connected to our srsLTE. fig-6.6 shows the wireshark capture for the parameter ‘MS assisted GPS’.

```

... ..00 111. .... = MS POSITIONING Method: 0x0/
.... ..0. = MS assisted E-OTD: MS assisted E-OTD not supported
.... ...0 = MS based E-OTD: MS based E-OTD not supported
1... .... = MS assisted GPS: MS assisted GPS supported
.1... .... = MS based GPS: MS based GPS supported
1      MS Conventional GPS: Conventional GPS supported

```

Figure 6.6: Parameter- MS assisted GPS

Using this parameter the Later-3 classification is been done by capturing different devices (for iOS devices the parameter initialised as ‘NOT Supported’ and otherwise for Android) and the pcap captures are accessed in MATLAB to classify them according to the exposed parameter, shown in fig-6.7.

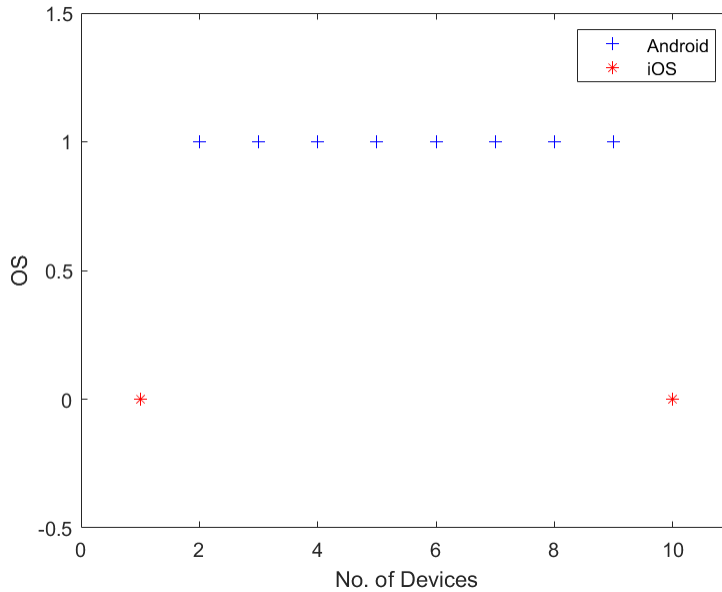


Figure 6.7: Classification- Layer3

6.1.4 Level-4

The level-4 classification would be differentiate between a Phone’s different Baseband Processors like Qualcomm or Mediatek or Samsung Exynos. Fig-6.8 shows the wireshark capture for the parameters like EIA0,Access Class Control for CSFB, CM Service Prompt and Extended Measurement Capability.

```

.0.. .... = HSCSD Multi Slot Capability: False
..1. .... = UCS2 treatment: the ME has no preference
...1 .... = Extended Measurement Capability: True
.... 0... = MS measurement capability: False
.... .1.. = MS Positioning Method Capability present
... .. = spare: 0
..1. .... = LCS VA capability (LCS value added location request notification capability): LCS
...1 .... = UCS2 treatment: the ME has no preference between the use of the default alphabet and
.... 0... = SoLSA: The ME does not support SoLSA
.... .0.. = CMSP: CM Service Prompt: Network initiated MO CM connection request not supported
.... .1.. = AS/3 algorithm supported: encryption algorithm AS/3 available
.... ..0 = AS/2 algorithm supported: encryption algorithm AS/2 not available
Mobile station classmark 3
Element ID: 0x20
... .. = ProSe direct discovery: Not supported
.0.. .... = ProSe: Not supported
..0. .... = H.245 After SRVCC Handover: Not supported
...0 .... = Access class control for CSFB: Not supported
... 1... = LTE Positioning Protocol: Supported
.... .1.. = Location services (LCS) notification mechanism
... .. = spare: 0

v UE network capability
  Length: 5
  1... .... = EEA0: Supported
  .1.. .... = 128-EEA1: Supported
  ..1. .... = 128-EEA2: Supported
  ...1 .... = 128-EEA3: Supported
  .... 0... = EEA4: Not supported

```

Figure 6.8: Parameters for Baseband Processor Identification

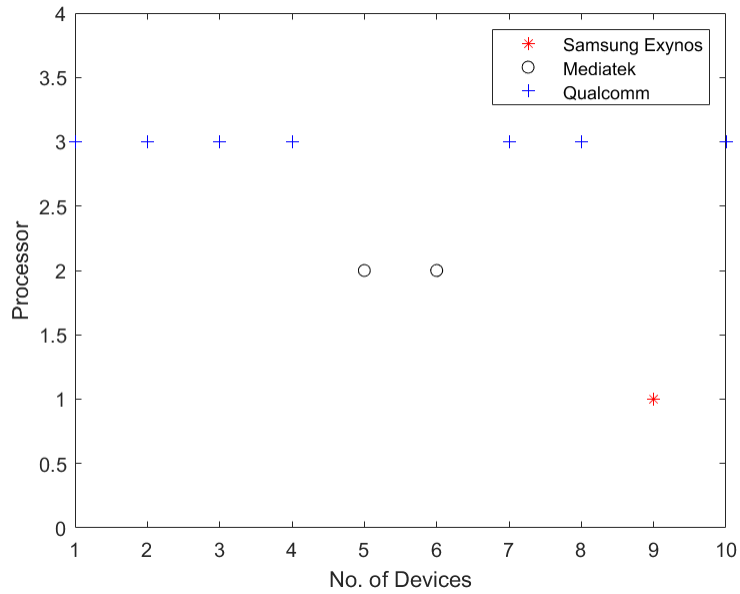


Figure 6.9: Classification- Layer4

Using these four parameters the Later-4 classification is been done by capturing different devices and the pcap captures are accessed in MATLAB to classify them according to the exposed parameter. The findings are if we take the four parameters serially as 1 for supported and 0 for Not supported, then 1100 will indicate Samsung Exynos processor, 1001 for Mediatek and 0110 for Qualcomm processor. Subsequently the pcap files are captured and classified as shown in fig 6.9.

6.2 GUI for Device Identification

Fig-6.10 is the MATLAB application which is a Graphical User Interface designed for the above project work. This Consists of all the devices attached to the srsLTE network and upon clicking on any device the four level classification will be shown, hence Enhancing the overall classification model for the 4G LTE system. In the figure shown, as the device 6 is clicked, we can see it is a Voice Centric Android Device (Mobile Phone) with VoLTE not enabled and has a Mediatek as Baseband processor.

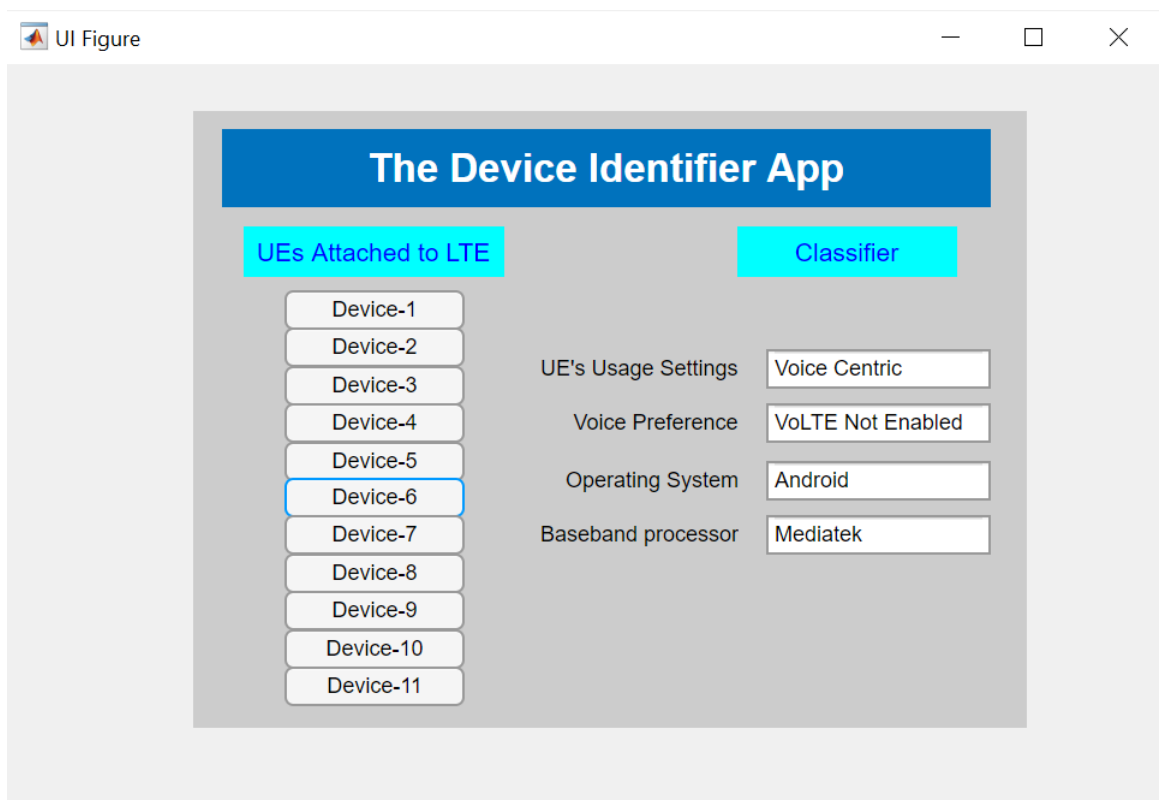


Figure 6.10: App designed in MATLAB for the UE classification

6.3 Fuzzy Inference System

As discussed earlier we have extracted seven important parameters which are transmitted in plain text in NAS-PDU and are responsible for our four level device classification. We have designed a (mamdani logic based) Fuzzy Inference System (FIS) which will indicate how vulnerable is our connected UE for different kind of passive or Active attacks in the initial-registration layer.

The inputs to the FIS are four device features, such as 1.ACC for CSFB (p1), Voice Domain Preference (p2), CM Service Prompt (p3), MS Assisted GPS (p3), For the 0 or 1 values of the given 4 parameters, here the FIS will have the ability to indicate 3 levels of vulnerability like high, medium or low vulnerable device based on the defined membership functions (MFs).

For the above written 4 parameters p1,p2,p3 and p4, The output is determined by following rules-

- For p2=1 and any values of p1,p3 and p4 AND p4=1, p3=0, p2=0, p1=0, it is defined as High vulnerability (values 0-0.33) category and UE capabilities can be easily compromised
- For p1=1, p2=0, p3=0, p4=1 AND p1=0, p2=1, p3=0, p4=1 AND p1=0, p2=0, p3=1, p4=1, it is defined as Medium vulnerability(value 0.33-0.66) where bidding down attacks are probable.
- For p1=1, p2=1, p3=1, p4=0, it is concluded as low vulnerable (value 0.67-1) or more secure device where passive/active attacks can't be done.

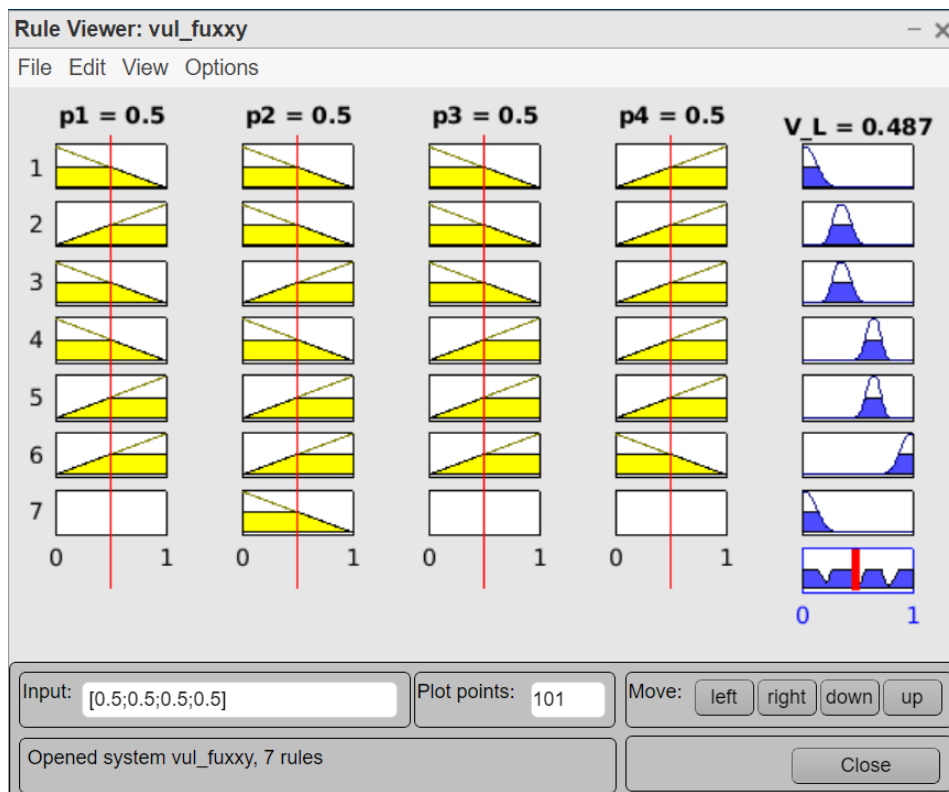


Figure 6.11: FIS for overall vulnerability level

Fig-6.11 shows overall vulnerability level is 0.487 which can be categorized as "medium Vulnerability". Similarly p1,p2,p3,p4 can be varied and the FIS will show the desired indication.

Fig 6.12 shows the surface viewer for any two parameter in x and y axes and how the vulnerability indicator varies can be seen in z axis among any of the p1, p2, p3, p4 defined parameters.

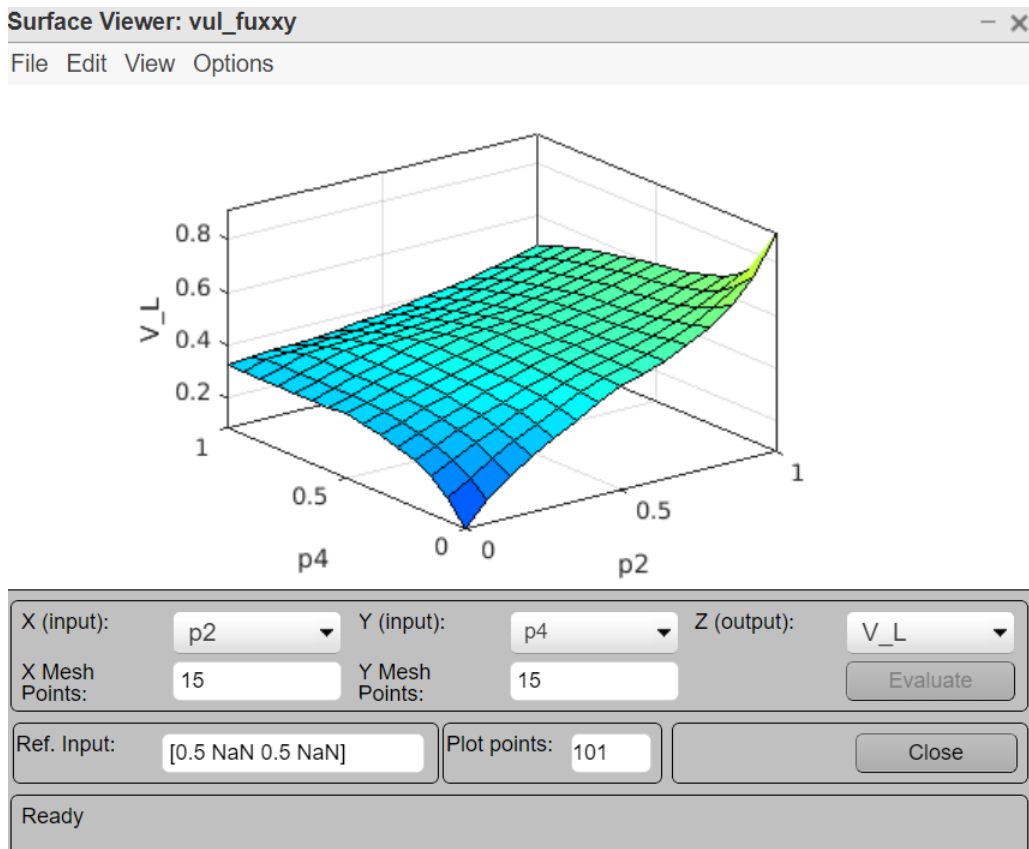


Figure 6.12: Surface view for vulnerability level

For example- P3 is CM service prompt which is responsible for emergency mode notifications, and this should be 1 for a highly secured device, but for 0 value, we can consider it as a mid_vulnerable device, as the bidding down attack will be possible without notifying the user. Also how vulnerable is the device towards attack will depend other 3 values of the parameters too. The relative variance between any two can be seen in the surface viewer fig-6.12

Chapter 7

Conclusion & Future Work

We have successfully completed the Device FingerPrinting and Classification work by exploiting the layer-3 Control signal messaging of LTE.

A 4G standalone test bed is presented along with its functionality. It was found that in 4G UE core network and radio access capabilities are exposed in clear text which can be used to identify Hardware and software specifications of UEs. Along with the device capabilities it was also found that various pre authentication NAS signalling messages like registration request/reject and authentication request/reject which are also not security protected.

As future work, this could include analysis of layer-2 User plane traffic and classification of Data Traffic for Website Fingerprinting which will lead to what kind of device along with what kind of website is being browsed by the connected UE. Making an Embedded Device FingerPrinting and Website Fingerprinting for the improved tracking of UEs in an LTE network is the future possibility. As the call flow and initial attach procedure in both 4G and 5G is exactly same, this work must easily get extended to next generation.

7.1 Summary of Findings

- We have an end to end 4G LTE setup with srsLTE core network and and USRP B210 eNodeB which will accept attach request from any commercial UE with given dl_earfcn number and other PLMN credentials and the logs are captured with wireshark tool
- Based on the pcap packet captures analysis of protocols is done and important 7 parameters (mentioned in chapter 6) are extracted which will be further used for Device Classification.
- Four layer classification is done based on the classification Model flow chart as explained in section 4.2 and figure 4.1.
- In section 6.1, where classification results are mentioned, all 4 levels are explained along with which extracted parameter(s) is/are leads to this conclusion for all the UE datasets.
- These captured parameter credentials with classification is visualised in MATLAB plotting and command window(figures are shown in section 6.1- classification results).
- In section 6.2 and 6.3, A Graphical User Interface and a Fuzzy Inference System is also designed in MATLAB to show the usecases of these classification results.

References

- [1] Y. Zou, J. Zhu, X. Wang and L. Hanzo, "A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends," in Proceedings of the IEEE, vol. 104, no. 9, pp. 1727-1765, Sept. 2016, doi: 10.1109/JPROC.2016.2558521.
- [2] https://traf.gov.in/sites/default/files/Wireless_Data_Service_Report_21082019_0.pdf
- [3] Shaik, A., Borgaonkar, R., Asokan, N., Niemi, V. and Seifert, J.P., 2015. Practical attacks against privacy and availability in 4G/LTE mobile communication systems. arXiv preprint arXiv:1510.07563.
- [4] Kohls, K., Rupperecht, D., Holz, T. and Pöpper, C., 2019, May. Lost traffic encryption: fingerprinting LTE/4G traffic on layer two. In Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks (pp. 249-260).
- [5] 3GPP. Characteristics of the Universal Subscriber Identity Module(USIM application) 3GPP TS 31.102 version 12.5.0 Release 12.[Online]. Available: <http://www.3gpp.org/dynareport/31102.htm>
- [6] 3GPP. Universal Mobile Telecommunications System (UMTS);Numbering,addressing and identification(3GPPTS23.003version 12.5.0 Release 12). [Online]. Available: <http://www.3gpp.org/dynareport/23003.htm>
- [7] 3GPP. Network Architecture ; Specification 3GPP TS 23.002 version 12.7.0 Release 12. [Online]. Available: <http://www.3gpp.org/DynaReport/23002.htm>

- [8] 3GPP. System Architecture Evolution (SAE); Security architecture;(3GPP 33.401 version 12.14.0 Release 12). [Online]. Available:<http://www.3gpp.org/dynareport/33.401.htm>
- [9] Zyren, J. and McCoy, W., 2007. Overview of the 3GPP long term evolution physical layer. Freescale Semiconductor, Inc., white paper, 7, pp.2-22.
- [10] Wang, M., Zhang, J., Ren, B., Yang, W., Zou, J., Hua, M. and You, X., 2015. The evolution of LTE physical layer control channels. *IEEE Communications Surveys & Tutorials*, 18(2), pp.1336-1354.
- [11] D. Rupprecht, K. Kohls, T. Holz and C. Pöpper, "Breaking LTE on Layer Two," 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2019, pp. 1121-1136, doi: 10.1109/SP.2019.00006.
- [12] https://www.tutorialspoint.com/lte/lte_protocol_stack_layers.htm
- [13] Shaik, A., Borgaonkar, R., Park, S. and Seifert, J.P., 2019, May. New vulnerabilities in 4G and 5G cellular access network protocols: exposing device capabilities. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks* (pp. 221-231).
- [14] srsLTE. Open source 3GPP LTE library. [Online]. Available:<https://github.com/srsLTE/srsLTE>
- [15] <https://www.3gpp.org/DynaReport/33-series.htm>
- [16] Zhanyi Wang. 2015. The Applications of Deep Learning on Traffic Identification. Technical Report. Black Hat USA.
- [17] Roger Piqueras Jover. 2016. LTE Security, Protocol Exploits and Location Tracking Experimentation with Low-Cost Software Radio. arXiv (1607.05171) (2016). arXiv:1607.05171 <http://arxiv.org/abs/1607.05171>
- [18] Xinyuan Wang, Shiping Chen, and Sushil Jajodia. 2005. Tracking Anonymous Peer-to-Peer VoIP Calls on the Internet. In *ACM Conference on Computer and Communications Security (CCS '05)*. ACM, Alexandria, VA, USA, 81-91.

- [19] Xinyuan Wang, Shiping Chen, and Sushil Jajodia. 2007. Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems. In IEEE Symposium on Security and Privacy (SP '07). IEEE, Oakland, CA, USA, 116–130.
- [20] Roger Piqueras Jover. 2013. Security Attacks Against the Availability of LTE Mobility Networks: Overview and Research Directions. In International Symposium on Wireless Personal Multimedia Communications (WPMC '13). IEEE, Atlantic City, NJ, USA.
- [21] Stig F. Mjølsnes and Ruxandra F. Olimid. 2017. Easy 4G/LTE IMSI Catchers for Non-Programmers. In Mathematical Methods, Models, and Architectures for Computer Network Security (MMM-ACNS '17). Springer, Warsaw, Poland, 235–246.
- [22] Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. 2014. A Critical Evaluation of Website Fingerprinting Attacks. In ACM Conference on Computer and Communications Security (CCS '14). ACM, Scottsdale, AZ, USA, 263–274.
- [23] Andriy Panchenko, Fabian Lanze, Andreas Zinnen, Martin Henze, Jan Pennekamp, Klaus Wehrle, and Thomas Engel. 2018. Website Fingerprinting at Internet Scale. In Network and Distributed System Security Symposium (NDSS '16). Internet Society, San Diego, CA, USA.
- [24] Brian N. Levine, Michael K. Reiter, Chenxi Wang, and Matthew Wright. 2004. Timing Attacks in Low-Latency Mix Systems. In International Conference on Financial Cryptography (FC '04). Springer, Key West, FL, USA, 251–265
- [25] FarhanF M. Aziz, Jeff S. Shamma, and Gordon L. Stüber. 2015. Resilience of LTE Networks Against Smart Jamming Attacks: Wideband Model. In Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '15). IEEE, Hong Kong, China, 1344–1348.

- [26] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. 2009. Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier. In *ACM Workshop on Cloud Computing Security (CCSW '09)*. ACM, Chicago, IL, USA, 31–42.
- [27] Andrew Hintz. 2002. Fingerprinting Websites Using Traffic Analysis. In *International Workshop on Privacy Enhancing Technologies (PET '02)*. Springer, San Francisco, CA, USA, 171–178.
- [28] Heyning Cheng and Ron Avnur. 1998. Traffic Analysis of SSL Encrypted Web Browsing. (1998).
- [29] Marc Lichtman, Roger Piqueras Jover, Mina Labib, Raghunandan Rao, Vuk Marojevic, and Jeffrey H. Reed. 2016. LTE/LTE-A Jamming, Spoofing, and Sniffing: Threat Assessment and Mitigation. *IEEE Communications Magazine* 54, 4 (April 2016), 54–61.
- [30] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. 2018. Automated Website Fingerprinting through Deep Learning. In *Network and Distributed System Security Symposium (NDSS '18)*. Internet Society, San Diego, CA, USA.
- [31] Marc Lichtman, Jeffrey H. Reed, T. Charles Clancy, and Mark Norton. 2013. Vulnerability of LTE to Hostile Interference. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP '13)*. IEEE, Austin, TX, USA, 285–288.
- [32] George Danezis, Claudia Diaz, and Carmela Troncoso. 2007. Two-Sided Statistical Disclosure Attack. In *International Workshop on Privacy Enhancing Technologies (PET '07)*. Springer, Ottawa, ON, Canada, 30–44.
- [33] Rob Jansen, Marc Juarez, Rafael Galvez, Tariq Elahi, and Claudia Diaz. 2017. Inside Job: Applying Traffic Analysis to Measure Tor from Within. In *Network and Distributed System Security Symposium (NDSS '17)*. Internet Society, San Diego, CA, USA.

- [34] The Tor Project. [n. d.]. The Onion Router. <https://www.torproject.org>. ([n. d.]). [Online; accessed 15-Nov-2018].
- [35] Steven J. Murdoch and Piotr Zieliński. 2007. Sampled Traffic Analysis by Internet- Exchange-Level Adversaries. In International Workshop on Privacy Enhancing Technologies (PET '07). Springer, Ottawa, ON, Canada, 167–183.
- [36] Steven J. Murdoch and George Danezis. 2005. Low-Cost Traffic Analysis of Tor. In IEEE Symposium on Security and Privacy (SP '05). IEEE, Oakland, CA, USA, 183–195.

Chapter 8

Appendices

8.1 Matlab code for feature identification

```
clc; clearvars; close all; i= input ("Enter Device No:");
    if i==1 file_id = fopen('D: captures/iphone6.txt'); elseif i==2 file_id = fopen('D:
captures/j8.txt'); elseif i==3 file_id = fopen('D: captures/j82.txt'); elseif i==4
file_id = fopen('D: captures/jioFi.txt'); elseif i==5 file_id = fopen('D: captures/moto
one.txt'); elseif i==6 file_id = fopen('D: captures/nokia1.txt'); elseif i==7 file_id
= fopen('D: captures/nokia5.txt'); elseif i==8 file_id = fopen('D: captures/on7
prime.txt'); elseif i==9 file_id = fopen('D: captures/pocox2.txt'); elseif i==10
file_id = fopen('D: captures/realmeX.txt'); elseif i== 11 file_id = fopen('D: cap-
tures/redmi note9pro.txt'); else fprintf("Not Enough Device"); end
    format= '%c'; file=fscanf(file_id,format); fprintf ("Device-%d:", i); find1_1 =
strfind(file,"UE's usage setting: "); %searching the position whole_feature_1_1
= file(find1_1: find1_1+32); %storing the whole feature from the text file
    feature_UE_1_1= "UE's usage setting: Voice centric"; %storing the actual
string which has to be compared
    tf1_1 =strcmp(feature_UE_1_1,whole_feature_1_1); %Comparison of two strings
and store the binary value
    if (tf1_1==1) % Writing the output according to binary value
        fprintf ("UE's usage setting is Voice Centric");
        find1_2 = strfind(file,"Voice domain preference for E-UTRAN"); %searching
```

```

the position whole_feature_1_2 = file(find1_2: find1_2+53); %storing the whole
feature from the text file
    feature_UE_1_2= "Voice domain preference for E-UTRAN: CS Voice only
(0)"; %storing the actual string which has to be compared
    tf1_2=strcmp(feature_UE_1_2,whole_feature_1_2); %Comparison of two strings
and store the binary value
    if (tf1_2 == 1) % Writing the output according to binary value fprintf("Device
is VoLTE not enabled"); else fprintf ("Device is VoLTE Enabled");
    end
    find1_3 = strfind(file,"MS assisted GPS:"); %searching the position whole_feature_1_3
= file(find1_3: find1_3+41); %storing the whole feature from the text file
    feature_UE_1_3= "MS assisted GPS: MS assisted GPS supported"; %storing
the actual string which has to be compared
    tf1_3=strcmp(feature_UE_1_3,whole_feature_1_3); %Comparison of two strings
and store the binary value
    if (tf1_3 == 1) % Writing the output according to binary value fprintf("Operating
System: Android"); else fprintf ("Operating System: iOS");
    end
    find1_4 = strfind(file,"EIA0:"); %searching the position whole_feature_1_4 =
file(find1_4: find1_4+14); %storing the whole feature from the text file
    feature_UE_1_4= "EIA0: Supported"; %storing the actual string which has
to be compared
    tf1_4=strcmp(feature_UE_1_4,whole_feature_1_4); %Comparison of two strings
and store the binary value
    find1_5 = strfind(file,"Access class control for CSFB:"); %searching the posi-
tion whole_feature_1_5 = file(find1_5: find1_5+39); %storing the whole feature
from the text file
    feature_UE_1_5= "Access class control for CSFB: Supported"; %storing the
actual string which has to be compared
    tf1_5=strcmp(feature_UE_1_5,whole_feature_1_5); %Comparison of two strings
and store the binary value

```

```

    find1_6 = strfind(file,"CMSP: CM Service Prompt:"); %searching the posi-
tion whole_feature_1_6 = file(find1_6: find1_6+105); %storing the whole f
    feature_UE_1_6= "CMSP: CM Service Prompt: Network initiated MO CM
connection request supported for at least one CM protocol"; %storing the actual
string which has to be compared
    tf1_6=strcmp(feature_UE_1_6,whole_feature_1_6); %Comparison of two strings
and store the binary value
    find1_7 = strfind(file,"Extended Measurement Capability:"); %searching the
position whole_feature_1_7 = file(find1_7: find1_7+36); %storing the whole
feature from the text file
    feature_UE_1_7= "Extended Measurement Capability: True"; %storing the
actual string which has to be compared
    tf1_7=strcmp(feature_UE_1_7,whole_feature_1_7); %Comparison of two strings
and store the binary value
    if (tf1_4 == 0 && tf1_5 == 1 && tf1_6 == 1 && tf1_7 ==0 ) % Writing the
output according to binary value fprintf("Baseband Processor: Qualcomm");
elseif (tf1_4 == 1 && tf1_5 == 0 && tf1_6 == 0 && tf1_7 == 1 ) fprintf("Baseband
Processor: Mediatek"); elseif (tf1_4 == 1 && tf1_5 == 1 && tf1_6 == 0 && tf1_7
==0 ) fprintf("Baseband Processor: Samsung Exynos"); else fprintf ("Other");
end
    else fprintf("UE's usage setting is Data centric"); end

```

8.2 Matlab GUI code

```

classdef app1 < matlab.apps.AppBase
    % Properties that correspond to app components properties (Access = pub-
lic) UIFigure matlab.ui.Figure Panel matlab.ui.container.Panel Device1Button
matlab.ui.control.Button Device2Button matlab.ui.control.Button Device3Button
matlab.ui.control.Button Device4Button matlab.ui.control.Button Device5Button
matlab.ui.control.Button Device6Button matlab.ui.control.Button Device7Button
matlab.ui.control.Button Device8Button matlab.ui.control.Button Device9Button

```

```

matlab.ui.control.Button Device10Button matlab.ui.control.Button Device11Button
matlab.ui.control.Button TheDeviceIdentifierAppLabel matlab.ui.control.Label
UEsAttachedtoLTELabel matlab.ui.control.Label ClassifierLabel matlab.ui.control.Label
UEsUsageSettingsEditFieldLabel matlab.ui.control.Label UEsUsageSettingsEd-
itField matlab.ui.control.EditField VoicePreferenceEditFieldLabel matlab.ui.control.Label
VoicePreferenceEditField matlab.ui.control.EditField OperatingSystemEditField-
Label matlab.ui.control.Label OperatingSystemEditField matlab.ui.control.EditField
BasebandprocessorEditFieldLabel matlab.ui.control.Label BasebandprocessorEd-
itField matlab.ui.control.EditField end

    % Callbacks that handle component events methods (Access = private)
    % Button pushed function: Device1Button function Device1ButtonPushed(app,
event) app.UEsUsageSettingsEditField.Value = "Voice Centric"; app.VoicePreferenceEditField.Value
="VoLTE Enabled"; app.OperatingSystemEditField.Value = "iOS"; app.BasebandprocessorEditField
="Qualcomm"; end

    % Button pushed function: Device2Button function Device2ButtonPushed(app,
event) app.UEsUsageSettingsEditField.Value = "Voice Centric"; app.VoicePreferenceEditField.Value
="VoLTE Enabled"; app.OperatingSystemEditField.Value = "Android"; app.BasebandprocessorEdit
="Qualcomm"; end

    % Button pushed function: Device4Button function Device4ButtonPushed(app,
event) app.UEsUsageSettingsEditField.Value = "Data Centric"; app.VoicePreferenceEditField.Value
="N/A"; app.OperatingSystemEditField.Value = "N/A"; app.BasebandprocessorEditField.Value
="N/A"; end

    % Button pushed function: Device3Button function Device3ButtonPushed(app,
event) app.UEsUsageSettingsEditField.Value = "Voice Centric"; app.VoicePreferenceEditField.Value
="VoLTE Enabled"; app.OperatingSystemEditField.Value = "Android"; app.BasebandprocessorEdit
="Qualcomm"; end

    % Button pushed function: Device5Button function Device5ButtonPushed(app,
event) app.UEsUsageSettingsEditField.Value = "Voice Centric"; app.VoicePreferenceEditField.Value
="VoLTE Enabled"; app.OperatingSystemEditField.Value = "Android"; app.BasebandprocessorEdit
="Qualcomm"; end

    % Button pushed function: Device6Button function Device6ButtonPushed(app,

```

```

event) app.UEsUsageSettingsEditField.Value = "Voice Centric"; app.VoicePreferenceEditField.Value
="VoLTE Not Enabled"; app.OperatingSystemEditField.Value ="Android"; app.BasebandprocessorEditField
="Mediatek"; end

    % Button pushed function: Device7Button function Device7ButtonPushed(app,
event) app.UEsUsageSettingsEditField.Value = "Voice Centric"; app.VoicePreferenceEditField.Value
="VoLTE Enabled"; app.OperatingSystemEditField.Value ="Android"; app.BasebandprocessorEditField
="Mediatek"; end

    % Button pushed function: Device8Button function Device8ButtonPushed(app,
event) app.UEsUsageSettingsEditField.Value = "Voice Centric"; app.VoicePreferenceEditField.Value
="VoLTE Enabled"; app.OperatingSystemEditField.Value ="Android"; app.BasebandprocessorEditField
="Samsung Exynos"; end

    % Button pushed function: Device9Button function Device9ButtonPushed(app,
event) app.UEsUsageSettingsEditField.Value = "Voice Centric"; app.VoicePreferenceEditField.Value
="VoLTE Enabled"; app.OperatingSystemEditField.Value ="Android"; app.BasebandprocessorEditField
="Qualcomm"; end

    % Button pushed function: Device10Button function Device10ButtonPushed(app,
event) app.UEsUsageSettingsEditField.Value = "Voice Centric"; app.VoicePreferenceEditField.Value
="VoLTE Enabled"; app.OperatingSystemEditField.Value ="Android"; app.BasebandprocessorEditField
="Qualcomm"; end

    % Button pushed function: Device11Button function Device11ButtonPushed(app,
event) app.UEsUsageSettingsEditField.Value = "Voice Centric"; app.VoicePreferenceEditField.Value
="VoLTE Enabled"; app.OperatingSystemEditField.Value ="iOS"; app.BasebandprocessorEditField
="Qualcomm"; end end

    % Component initialization methods (Access = private)
    % Create UIFigure and components function createComponents(app)
    % Create UIFigure and hide until all components are created app.UIFigure
= uifigure('Visible', 'off'); app.UIFigure.Position = [100 100 640 480]; app.UIFigure.Name
= 'UI Figure';

    % Create Panel app.Panel = uipanel(app.UIFigure); app.Panel.ForegroundColor
= [0 1 1]; app.Panel.BackgroundColor = [0.8 0.8 0.8]; app.Panel.Position = [104
113 463 342];

```

```

    % Create Device1Button app.Device1Button = uibutton(app.Panel, 'push');
app.Device1Button.ButtonPushedFcn = createCallbackFcn(app, @Device1ButtonPushed,
true); app.Device1Button.Position = [51 221 100 22]; app.Device1Button.Text =
'Device-1';

    % Create Device2Button app.Device2Button = uibutton(app.Panel, 'push');
app.Device2Button.ButtonPushedFcn = createCallbackFcn(app, @Device2ButtonPushed,
true); app.Device2Button.Position = [51 200 100 22]; app.Device2Button.Text =
'Device-2';

    % Create Device3Button app.Device3Button = uibutton(app.Panel, 'push');
app.Device3Button.ButtonPushedFcn = createCallbackFcn(app, @Device3ButtonPushed,
true); app.Device3Button.Position = [51 179 100 22]; app.Device3Button.Text =
'Device-3';

    % Create Device4Button app.Device4Button = uibutton(app.Panel, 'push');
app.Device4Button.ButtonPushedFcn = createCallbackFcn(app, @Device4ButtonPushed,
true); app.Device4Button.Position = [51 158 100 22]; app.Device4Button.Text =
'Device-4';

    % Create Device5Button app.Device5Button = uibutton(app.Panel, 'push');
app.Device5Button.ButtonPushedFcn = createCallbackFcn(app, @Device5ButtonPushed,
true); app.Device5Button.Position = [51 137 100 22]; app.Device5Button.Text =
'Device-5';

    % Create Device6Button app.Device6Button = uibutton(app.Panel, 'push');
app.Device6Button.ButtonPushedFcn = createCallbackFcn(app, @Device6ButtonPushed,
true); app.Device6Button.Position = [51 117 100 22]; app.Device6Button.Text =
'Device-6';

    % Create Device7Button app.Device7Button = uibutton(app.Panel, 'push');
app.Device7Button.ButtonPushedFcn = createCallbackFcn(app, @Device7ButtonPushed,
true); app.Device7Button.Position = [51 96 100 22]; app.Device7Button.Text =
'Device-7';

    % Create Device8Button app.Device8Button = uibutton(app.Panel, 'push');
app.Device8Button.ButtonPushedFcn = createCallbackFcn(app, @Device8ButtonPushed,
true); app.Device8Button.Position = [51 75 100 22]; app.Device8Button.Text =

```

```

'Device-8';
    % Create Device9Button app.Device9Button = uibutton(app.Panel, 'push');
app.Device9Button.ButtonPushedFcn = createCallbackFcn(app, @Device9ButtonPushed,
true); app.Device9Button.Position = [51 54 100 22]; app.Device9Button.Text =
'Device-9';
    % Create Device10Button app.Device10Button = uibutton(app.Panel, 'push');
app.Device10Button.ButtonPushedFcn = createCallbackFcn(app, @Device10ButtonPushed,
true); app.Device10Button.Position = [51 33 100 22]; app.Device10Button.Text
= 'Device-10';
    % Create Device11Button app.Device11Button = uibutton(app.Panel, 'push');
app.Device11Button.ButtonPushedFcn = createCallbackFcn(app, @Device11ButtonPushed,
true); app.Device11Button.Position = [51 12 100 22]; app.Device11Button.Text
= 'Device-11';
    % Create TheDeviceIdentifierAppLabel app.TheDeviceIdentifierAppLabel
= uilabel(app.Panel); app.TheDeviceIdentifierAppLabel.BackgroundColor = [0
0.4471 0.7412]; app.TheDeviceIdentifierAppLabel.HorizontalAlignment = 'cen-
ter'; app.TheDeviceIdentifierAppLabel.FontSize = 22; app.TheDeviceIdentifierAppLabel.FontWei-
ght = 'bold'; app.TheDeviceIdentifierAppLabel.FontColor = [1 1 1]; app.TheDeviceIdentifierAppLabel
.Position = [16 289 427 43]; app.TheDeviceIdentifierAppLabel.Text = 'The Device Identi-
fier App';
    % Create UEsAttachedtoLTELabel app.UEsAttachedtoLTELabel = uilabel(app.Panel);
app.UEsAttachedtoLTELabel.BackgroundColor = [0 1 1]; app.UEsAttachedtoLTELabel.Horizontal
Alignment = 'center'; app.UEsAttachedtoLTELabel.FontSize = 14; app.UEsAttachedtoLTELabel.FontColor
= [0 0 1]; app.UEsAttachedtoLTELabel.Position = [28 250 145 28]; app.UEsAttachedtoLTELabel.Text
= 'UEs Attached to LTE';
    % Create ClassifierLabel app.ClassifierLabel = uilabel(app.Panel); app.ClassifierLabel.Backgroun-
dColor = [0 1 1]; app.ClassifierLabel.HorizontalAlignment = 'center'; app.ClassifierLabel.FontSize
= 14; app.ClassifierLabel.FontColor = [0 0 1]; app.ClassifierLabel.Position =
[302 250 122 28]; app.ClassifierLabel.Text = 'Classifier';
    % Create UEsUsageSettingsEditFieldLabel app.UEsUsageSettingsEditFieldLabel
= uilabel(app.Panel); app.UEsUsageSettingsEditFieldLabel.HorizontalAlignment

```



```

= 'right'; app.UEsUsageSettingsEditFieldLabel.Position = [188 188 115 22]; app.UEsUsageSettingsE
= 'UE"s Usage Settings';
    % Create UEsUsageSettingsEditField app.UEsUsageSettingsEditField = uied-
itfield(app.Panel, 'text'); app.UEsUsageSettingsEditField.Position = [318 188
125 22];
    % Create VoicePreferenceEditFieldLabel app.VoicePreferenceEditFieldLabel
= uilabel(app.Panel); app.VoicePreferenceEditFieldLabel.HorizontalAlignment
= 'right'; app.VoicePreferenceEditFieldLabel.Position = [206 158 97 22]; app.VoicePreferenceEditFie
= 'Voice Preference';
    % Create VoicePreferenceEditField app.VoicePreferenceEditField = uiedit-
field(app.Panel, 'text'); app.VoicePreferenceEditField.Position = [318 158 125
22];
    % Create OperatingSystemEditFieldLabel app.OperatingSystemEditFieldLabel
= uilabel(app.Panel); app.OperatingSystemEditFieldLabel.HorizontalAlignment
= 'right'; app.OperatingSystemEditFieldLabel.Position = [201 126 102 22]; app.OperatingSystemEd
= 'Operating System';
    % Create OperatingSystemEditField app.OperatingSystemEditField = uied-
itfield(app.Panel, 'text'); app.OperatingSystemEditField.Position = [318 126 125
22];
    % Create BasebandprocessorEditFieldLabel app.BasebandprocessorEditFieldLabel
= uilabel(app.Panel); app.BasebandprocessorEditFieldLabel.HorizontalAlignment
= 'right'; app.BasebandprocessorEditFieldLabel.Position = [187 96 116 22]; app.Basebandprocessor
= 'Baseband processor';
    % Create BasebandprocessorEditField app.BasebandprocessorEditField =
uieditfield(app.Panel, 'text'); app.BasebandprocessorEditField.Position = [318
96 125 22];
    % Show the figure after all components are created app.UIFigure.Visible =
'on'; end end
    % App creation and deletion methods (Access = public)
    % Construct app function app = app1
    % Create UIFigure and components createComponents(app)

```

```

% Register the app with App Designer registerApp(app, app.UIFigure)
if nargout == 0 clear app end
% Code that executes before app deletion function delete(app)
% Delete UIFigure when app is deleted delete(app.UIFigure) end end

```

8.3 FIS code

```

[System] Name='vul_fuxxy' Type='mamdani' Version=2.0 NumInputs=4 NumOutputs=1 NumRules=7 AndMethod='min' OrMethod='max' ImpMethod='min' AggMethod='max' DefuzzMethod='centroid' [Input1] Name='ACC_for_CSFB' Range=[0 1] NumMFs=2 MF1='high_vul':'trimf',[-1 0 1] MF2='low_vul':'trimf',[0 1 2] [Input2] Name='Voice_dom_preference' Range=[0 1] NumMFs=2 MF1='low_vul':'trimf',[0 1 2] MF2='high_vul':'trimf',[-1 0 1] [Input3] Name='CM_Service_Prompt' Range=[0 1] NumMFs=2 MF1='high_vul':'trimf',[-1 0 1] MF2='low_vul':'trimf',[0 1 2] [Input4] Name='MS_assisted_GPS' Range=[0 1] NumMFs=2 MF1='low_vul':'trimf',[-1 0 1] MF2='high_vul':'trimf',[0 1 2] [Output1] Name='Vul_level' Range=[0 1] NumMFs=4 MF1='high_vul':'gauss2mf',[0.2691 -0.0792 0.09852 0.029] MF2='med_vul':'gauss2mf',[0.333 0.06795 0.37] MF3='low_vul':'gauss2mf',[0.08493 0.975 0.1699 1.05] MF4='med_ack':'gauss2mf',[0.634 0.04756 0.664] [Rules] 1 2 1 2, 1 (1) : 1 2 2 1 2, 2 (1) : 1 1 1 1 2, 2 (1) : 1 1 2 2 2, 4 (1) : 1 2 1 2 2, 4 (1) : 1 2 1 2 1, 3 (1) : 1 0 2 0 0, 1 (1) : 1

```