2021-06

Spoofed Networks: Exploitation of GNSS Security Vulnerability in 4G and 5G Mobile Networks

Lanoue, Matthew J.

Monterey, CA; Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

SPOOFED NETWORKS: EXPLOITATION OF GNSS SECURITY VULNERABILITY IN 4G AND 5G MOBILE NETWORKS

by

Matthew J. Lanoue

June 2021

Thesis Advisor: Chad A. Bollmann
Co-Advisors: James B. Michael
Darren J. Rogers

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2021	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
SPOOFED NETWORKS: EXPLOITATION OF GNSS SECURITY VULNERABILITY IN 4G AND 5G MOBILE NETWORKS

5. FUNDING NUMBERS
REP76

6. AUTHOR(S) Matthew J. Lanoue

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Navy Cyber Warfare Development Group (NCWDG) 4251 Suitland Rd, Washington, D.C. 20395

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

Fifth Generation New Radio (5G NR) represents a shift in mobile telephony whereby the network architecture runs containerized software on commodity hardware. In preparation for this transition, numerous 4G Long Term Evolution software stacks have been developed to test the containerization of core network functions and the interfaces with radio access network protocols. In this thesis, one such stack, developed by the OpenAirInterface Software Alliance, was used to create a low-cost, simplified mobile network compatible with the Naval Operational Architecture. Commercial off-the-shelf user equipment was then connected to the network to demonstrate how a buffer overflow vulnerability found in Qualcomm Global Navigation Satellite System chipsets and identified as CVE-2019-2254 can be leveraged to enable a spoofed network attack. The research also yielded an extension of the attack method to 5G NR networks.

14. SUBJECT TERMS
4G, Long Term Evolution, LTE, 5G, New Radio, NR, mobile telephony, radio access network, RAN, security vulnerabilities, OpenAirInterface Software Alliance, OSA, Global Navigation Satellite System, GNSS, networks, spoofing attacks, commercial off-the-shelf, COTS, user equipment, UE

15. NUMBER OF PAGES
167

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

SPOOFED NETWORKS: EXPLOITATION OF GNSS SECURITY VULNERABILITY IN 4G AND 5G MOBILE NETWORKS

Matthew J. Lanoue
Lieutenant, United States Navy
BS, United States Naval Academy, 2014

Submitted in partial fulfillment of the
requirements for the degrees of

ELECTRICAL ENGINEER

and

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2021

Approved by: Chad A. Bollmann
Advisor

James B. Michael
Co-Advisor

Darren J. Rogers
Co-Advisor

Douglas J. Fouts
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Fifth Generation New Radio (5G NR) represents a shift in mobile telephony whereby the network architecture runs containerized software on commodity hardware. In preparation for this transition, numerous 4G Long Term Evolution software stacks have been developed to test the containerization of core network functions and the interfaces with radio access network protocols. In this thesis, one such stack, developed by the OpenAirInterface Software Alliance, was used to create a low-cost, simplified mobile network compatible with the Naval Operational Architecture. Commercial off-the-shelf user equipment was then connected to the network to demonstrate how a buffer overflow vulnerability found in Qualcomm Global Navigation Satellite System chipsets and identified as CVE-2019-2254 can be leveraged to enable a spoofed network attack. The research also yielded an extension of the attack method to 5G NR networks.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

3GPP	Third Generation Partnership Project
AMPS	Advanced Mobile Phone System
APN	access point name
BPSK	binary phase shift keying
BMC	best master clock
CUPS	control and user plane split
CN	core network
CVE	common vulnerabilities and exposures
CVSS	Common Vulnerability Scoring System
eNB	evolved NodeB
EPC	evolved packet core
gNB	next-generation NodeB
ICCID	integrated circuit card identifier
IMEI	international mobile equipment identifier
IMSI	international mobile subscriber identifier
GLONASS	Global'naya Navigatsionnaya Sputnikovaya Sistema
GNSS	global navigation satellite system
GPS	Global Positioning System
LTE	long term evolution
MCC	mobile country code
MNC	mobile network code
MNO	mobile network operator
MSISDN	mobile subscriber integrated services digital network
MVNO	mobile virtual network operator
NMT	Nordic Mobile Telephony
NTP	Network Time Protocol
OAI	OpenAirInterface
OS	operating system
OSA	OpenAirInterface Software Alliance
OTA	over the air

NTP	network time protocol
PNT	positioning, navigation, and timing
PTP	Precision Time Protocol
RAN	radio access network
RRC	radio resource control
RTK	real-time kinematics
RTT	round-trip time
SIM	subscriber identity module
SUCI	subscription concealed identifier
SUPI	subscription permanent identifier
SMS	short message service
SMSC	short message service center
SNTP	Simple Network Time Protocol
TACS	Total Access Communication System
UE	user equipment
UICC	universal integrated circuit card
USIM	universal subscriber identity module
USRP	Universal Software Radio Peripheral

EXECUTIVE SUMMARY

As the rollout of fifth generation (5G) of mobile telephony networks progresses, industry is already ramping up its research and development to transition from 5G New Radio (5G NR) to sixth generation (6G) mobile telephony. However, there is at least one technological gap in the R&D needed to support both the rollout of 5G and advancement of 6G technology. That gap is a dearth of R&D on managing cyber security risk pertaining to 5G networks. This thesis explores the identification of, classification of, and experimentation with 5G cyber-attack vectors. The thesis research resulted in the: creation and demonstration of the use of a scheme tailored specifically for classifying 5G attack vectors; development and demonstration of an apparatus and procedure for experimenting with a specific type of 5G attack vector; and recommendations for using the classification scheme, experimental apparatus and procedure to explore 5G attack vectors.

The classification scheme introduced in this thesis involves mapping attack vectors based on their relationships to the 5G control and user plane split (CUPS). This scheme is a significant departure from what is typically used in the cyber security community. Instead of focusing on the attack method, the proposed classification scheme accounts for the network mechanisms utilized as the source and target of the attack. This thesis documents the scheme and its use in classifying attack vectors from fourth generation (4G) networks, some of which may be applicable to 5G networks because 5G inherits aspects of the 4G standard. The results of this part of the thesis were published in the April 2021 issue of the IEEE Computer Society *Computer* magazine.

In order to advance the practice of security engineering for 5G network telephony, this thesis documents the development and demonstration of a new procedure for experimenting with a specific type of 5G attack vector. This part of the thesis research is based partially on the publicly disclosed security vulnerability referred to in the common vulnerabilities and exposures (CVE) program catalog as CVE-2019-2254. The vulnerability is that an attacker can cause a buffer overflow to occur in certain global navigation satellite system (GNSS) chips manufactured by Qualcomm. The buffer overflow can then be used to execute arbitrary commands on the target cellular modem.

For instance, an attacker could covertly copy sensitive subscriber information such as the International Mobile Subscriber Identity (IMSI), International Mobile Equipment Identifier (IMEI), and encryption keys. The subscriber information can then be used by the attacker to pretend to be the target UE, eavesdrop on any communications to and from the user equipment (UE), or even force the target UE to attach to a spoofed mobile network.

The EC20 modem produced by Quectel uses vulnerable Qualcomm GNSS chipsets identified in CVE-2019-2254. The modem is capable of operating in a variety of modes and cellular bands. While the EC20 is 4G capable, it is not capable of operating on standalone 5G networks. Although significant differences exist between 4G and 5G networks, these differences are transparent to the attack vector identified. The spoofed mobile network attack vector serves as the case study for this thesis.

The apparatus used in this thesis for experimentation with the attack vector is an open stack 4G Long Term Evolution (LTE) network. We created this network from a single UE (the EC20) connected via SMA cables to an eNodeB (eNB) and an evolved packet core (EPC). The eNB and EPC utilize open-source software from the OpenAirInterface (OAI) Software Alliance (OSA)—a French non-profit organization that develops cellular Radio Access Network (RAN) and Core Network (CN) solutions in conjunction with commercial and academic entities. Some member organizations include: Eurecom, Qualcomm, Facebook Connectivity, Fujitsu, Xilinx, Inmarsat, Nokia Bell Labs, Rutgers Wireless Information Network Laboratory, Rice University, and Fraunhofer. This thesis research uses code from the OSA 4G LTE RAN and 4G EPC repositories. The eNB also utilizes a radio front-end from National Instruments, the Universal Software Radio Peripheral (USRP) B200.

In addition to the open stack 4G LTE network, we configured a universal integrated circuit card (UICC) for use in testing the attack vector. This involved a detailed study of the UICCs, including file systems, applications, commands, and reading and writing data. Both 4G and 5G UICCs contain a Java-based application called a universal subscriber identity module (USIM). An important element in getting the USIM to work within the experimental setting was to configure service flags within the USIM to show that the user is provisioned for 5G service. However, those same memory locations are used to store

information such as the IMSI and cryptography keys. The implication here is that 5G UEs, despite improvements in the authentication and keying agreement (AKA) procedure and protection of subscription permanent identifiers (SUPI), will still be vulnerable to the identified attack vector. The UICC contains additional resources that are used by mobile network operators (MNO) to provide applications for billing, or by third-party vendors to provide services such as mobile banking. Once the attacker gains initial access to the UICC, additional applications can be loaded to maintain a persistent presence on the target device. Experimentation with the 4G network resulted in key parameters and utilities being identified, integrated, and used within the setup of the open stack network.

The results from experimentation indicate that the open stack network concept is low-cost, flexible, and scalable to meet the demands of both network users and security researchers. In particular, the 4G LTE open stack network was utilized to explore how CVE-2019-2254 could be exploited on the Quectel EC20 modem. While the OSA is developing software for 5G RAN and CN, the 5G software did not reach maturity in time for incorporation into this thesis. However, the underlying concepts and many of the commands and utilities demonstrated in this thesis will be similar for the 5G software.

Through exploration with the open stack network, the constructs for a spoofed network attack vector were developed. The identified attack vector expands the scope of false base station attacks to reveal the possibility of spoofed networks. Additional utilities are required to form a spoofed network with full-service capabilities, such as a simple message service center (SMSC) and integration with MNO phone number registries. Furthermore, the UICC read and write utilities required for configuring subscriber information in the USIM application provides a steppingstone for additional research into the security mechanisms for protecting subscriber information in the UE. Finally, the utility of the mobile telephony attack vector taxonomy published in *Computer* magazine was validated as this research chained multiple vulnerabilities together to expose inherent weaknesses in UICCs in the context of 4G and 5G networks.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank the faculty and staff at NPS for their hard work and insight throughout my studies. I was fortunate enough to take classes with most of the professors in the Electrical and Computer Engineering Department and then to apply some of what they had taught me as I worked on my thesis. Additional debts of gratitude are owed to CDR Bollmann, Professor Michael, Professor Roth, and Darren Rogers. Thank you for guiding me as I explored the enormous world of mobile communications and then tried to break it. Darren, I want to offer an additional thanks for being my sounding board throughout the process and for always making time to help troubleshoot.

Most importantly, I want to thank my wife, Mary. All my accomplishments are possible because of your love and support. You enabled me to pursue my dreams as I wandered around with my head in the clouds and you helped pull me back when I got lost. Thank you for always listening to me and sympathizing when I was unable to impress my will upon tiny rectangles. We did it.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The 5th generation (5G) of mobile telephony standards, like the 4th generation (4G) before it, will require precision timing which can be provided by GNSS constellations or transported between nodes via Partial Timing Support (PTS) or Full Timing Support (FTS). In order to provide timing redundancy and resiliency, some modem deployments will include global navigation satellite system (GNSS) receivers. Due to the prevalence of GNSS chipsets in mobile devices and a limited set of manufacturers, a single security vulnerability in a GNSS receiver chip could disproportionately affect numerous models and generations of cellular modems.

On July 2, 2019, an article released by the Center for Internet Security warned of multiple vulnerabilities found within the Google Android operating system (OS) that could be used by an attacker to execute arbitrary code on a target device [1]. In addition to OS-level vulnerabilities, several component-level vulnerabilities were detailed for Qualcomm chips. One of the Qualcomm component vulnerabilities is common vulnerabilities and exposures (CVE) -2019-2254 which received a 9.8 out of 10, that is "critical," severity rating using the Common Vulnerability Scoring System (CVSS) version 3.x criteria [2], [3]. The vulnerability description states that "position accuracy may be degraded due to wrongly decoded information." An additional vulnerability is CVE-2018-13887 which also received a 9.8 out of 10 rating [4]. In this vulnerability, certain Qualcomm modems could experience integer overflows through untrusted header fields in the GNSS XTRA3 function.

Both vulnerabilities affect similar Qualcomm modem product lines through GNSS components and could be used to conduct buffer overflow attacks to execute arbitrary code on the target device. The severity of these vulnerabilities led us to ask the following questions:

- Are GNSS chipsets required components for 4G or 5G-capable cellular telephones?
- What GNSS chipsets or capabilities exist in the current cellphone market?

- Can CVE-2019-2254 or CVE-2018-13887 be used in 4G or 5G networks?

Given that the Navy is interested in 5G technology, instead of just focusing on these two vulnerabilities it is prudent to understand how the vulnerabilities fit into the bigger picture: What are the categories of attack vectors for 5G communications? Answering this question led to the construction of an attack vector categorization scheme [5]. This taxonomy utilizes the control and user plane separation (CUPS) in the 5G NR architecture to provide an alternative method for analyzing mobile telephony attack vectors.

In this thesis, we further investigated the vulnerabilities announced in CVE-2018-13887 and CVE-2019-2254. To support the research, we constructed an open stack 4G LTE network. The open stack network utilizes open-source software running on commodity hardware to provide a low-cost, scalable, and adaptable platform for use in security research. Beyond providing accessibility to academics and security researchers, the open stack network could also provide key communications infrastructure that interfaces with the Naval Operational Architecture [6], [7].

While setting-up the open stack network, we developed a spoofed network attack that builds upon the false base station attack used against legacy networks. This attack vector leverages the security vulnerabilities in the GNSS chipset to execute modem-level commands and exfiltrate sensitive subscriber information and cryptography keys. With this information, the attacker could force the target mobile device to attach to the spoofed network. If the spoofed network parameters are configured to mirror legitimate commercial networks and sufficient services are implemented, the attacker could covertly copy, filter, or alter traffic to and from the mobile device. Some aspects of this proposed attack vector remain for future work; however, the key concepts of modem-level commands to discover subscriber information and forcibly attach a target device to the spoofed network are demonstrated in this work.

The remainder of this thesis consists of five chapters: background, equipment and network set-up, spoofed network attack theory, results, and conclusions and recommendations for follow-on research, respectively.

II. BACKGROUND

This chapter includes material adapted from work to be published by the author. Specifically, Sections II.A.2, II.A.3, and II.B.3 contain some revised material from "Spoofed Networks: Exploitation of GNSS Security Vulnerability in 4G and 5G Mobile Networks" by Matthew J. Lanoue, James Bret Michael, and Chad A. Bollmann, to be published in the 2021 *International Symposium on Performance Evaluation of Computer and Telecommunication Systems*.

A. 4G/5G FUNDAMENTALS

1. Historical Overview

Mobile telephony has advanced from the analog first-generation technologies of the 1980s such as Advanced Mobile Phone System (AMPS), Total Access Communication System (TACS), and Nordic Mobile Telephony (NMT), retroactively labeled 1G, to delivering a fifth-generation standard, 5G New Radio (5G NR or 5G), in the 2020s. Currently, 4G LTE is the global.

Mobile telephony standards are created and maintained by the Third Generation Partnership Project (3GPP), an international standards organization that has overseen the development and maintenance of mobile standards since 2G. 3GPP standards are consolidated into a release that governs multiple features or aspects of mobile telephony. Release 8 in 2008 was the first release that covered 4G LTE. Release 15 in 2018 is the first step in defining the 5G New Radio (NR) standard. It is important to note that although 5G is currently the most cutting-edge technology, the majority of the mobile telephone market is still dominated by 4G technology, as shown in Figure 1. As a result, newer standards must incorporate some degree of backwards compatibility.

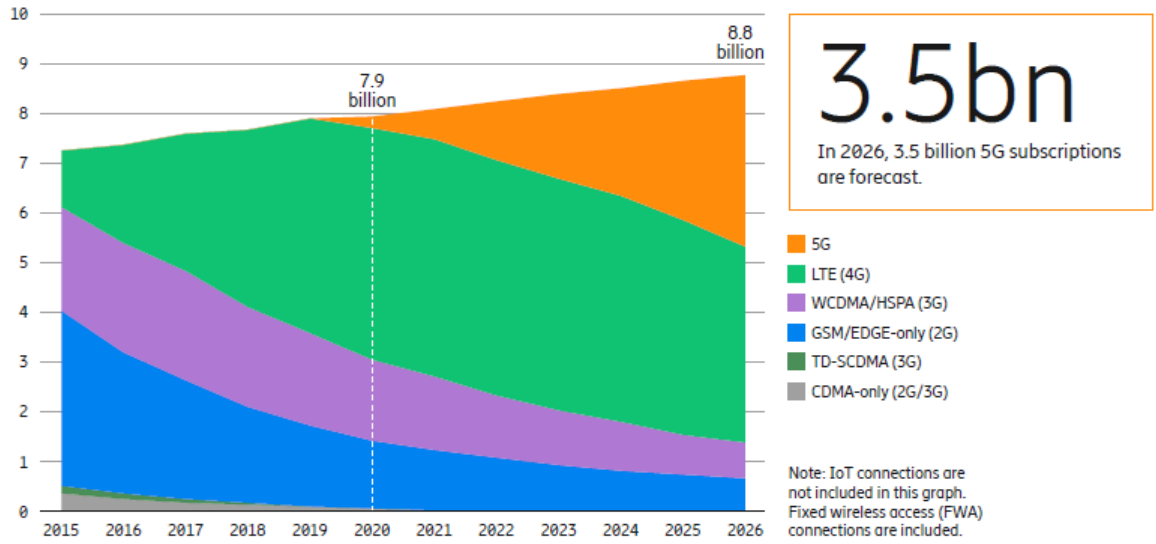

Figure 1. Global Mobile Telephone Market Share by Technology. Source: [8].

While there are currently over 3 billion 2G and 3G devices, many mobile network operators (MNOs) are starting to shutdown these legacy networks. In the United States, AT&T led the way in 2017 by decommissioning its 2G network [9]. AT&T announced its intention to shut down its 3G network service in early 2022 [10]. Verizon will sunset its 3G code division multiple access (CDMA) network at the end of 2022 [11]. Security researchers must upgrade their mobile telephony hardware to more modern 4G and 5G set-ups or risk losing market share.

2. 4G LTE Architecture and Overview

LTE (Long Term Evolution) was designed to improve upon 3G by incorporating spectrum flexibility in the form of utilizing unpaired and paired channels via Frequency-Division Duplex (FDD) and Time-Division Duplex (TDD), respectively [12]. LTE uses orthogonal frequency-division multiplexing (OFDM) to transmit multiple symbols per subframe. At a high level, the 4G network architecture contains a capable user equipment (UE), the Evolved UMTS Terrestrial Radio Access Network (E-UTRAN), and the Evolved Packet Core (EPC). The E-UTRAN consists of multiple eNodeB, which are evolved base

stations that combine the functions of the Base Station (BS) and Radio Network Controller (RNC) from the 3G standard. Figure 2 depicts the major parts of a 4G network architecture.

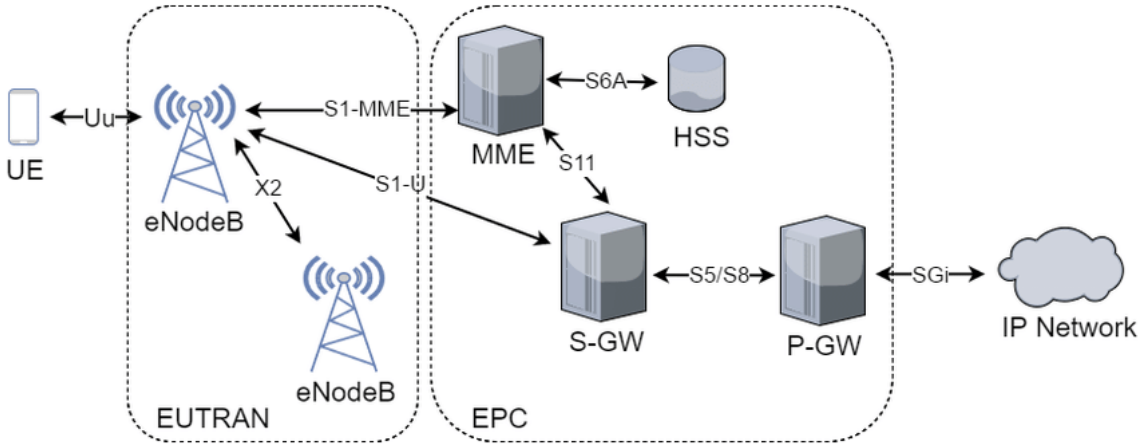

Figure 2. 4G Network Architecture. Source: [13].

In the core network (CN), the home subscriber server (HSS) contains subscriber information such as the IMSI and master key associated with each user. The mobility management entity (MME) controls the security and mobility of all UEs attached to the network and acts as an intermediary between the UE and HSS during the authentication and key agreement (AKA) process. The serving gateway (S-GW) routes data between the RAN and the packet data network (PDN). The PDN gateway (P-GW) allocates IP addresses to UEs attached to the network and communicates with external data networks over the SGi interface.

3. 5G Architecture and Overview

5G NR is designed to incorporate three primary usage cases: enhanced mobile broadband (eMBB), massive machine-type communication (mMTC), and ultra-reliable low-latency communication (URLLC). Each of the usage cases has its own set of requirements and is used to simplify the creation of technical specifications, and there may be actual usage requirements that do not fit neatly within one of the three primary usage cases. Some key enhancements provided by 5G include operation in higher frequency bands—including mm-wave frequencies, minimizing the overhead associated with

"always-on" signals, flexible subcarrier spacing, and channel-dependent scheduling. 5G also uses OFDM, with various TDD and FDD channelization schemes.

5G NR network architecture, like 4G LTE, contains a RAN and CN. Figure 3 shows a basic 5G network with a detailed insert for some of the most relevant core network functions, and Figure 4 shows a more complex RAN with a disaggregated next-generation base station (gNB). In the 5G core network, the EPC is expanded to included service-based architecture, support for network slicing, and a functionality split between the control plane and user plane. Additionally, network functionality is split across a larger number of network functions than in the 4G EPC. Within the RAN, gNodeB (gNB) and ng-eNodeB (ng-eNB) can connect to the 5G core network. The difference between the two types of nodes is that the gNB is used for NR devices via the 5G user and control-plane protocols, while the ng-eNB is used for LTE devices via the 4G user- and control-plane protocols. Figure 5 shows several RAN options used for 4G and 5G networks, where the "c" represents the control-plane link and the "u" represents the user-plane link. Option 1 is a standard 4G RAN, option 2 is a 5G standalone (SA) RAN, and option 3 is a 5G non-standalone (NSA) RAN.

For comparison with 4G EPC network functions, we briefly review the User Plane function (UPF), Session Management function (SMF), Access and Mobility Management function (AMF), Unified Data Management (UDM), and Authentication Server function (AUSF). The UPF routes data between the RAN and external data networks. The SMF allocates IP addresses to the UE, conducts traffic steering at the UPF, and enforces network policies. The AMF is responsible for control signaling with the device and RAN, including connection, registration, and mobility management. The AUSF supports authentication for network access using credentials and subscription information stored in the UDM.

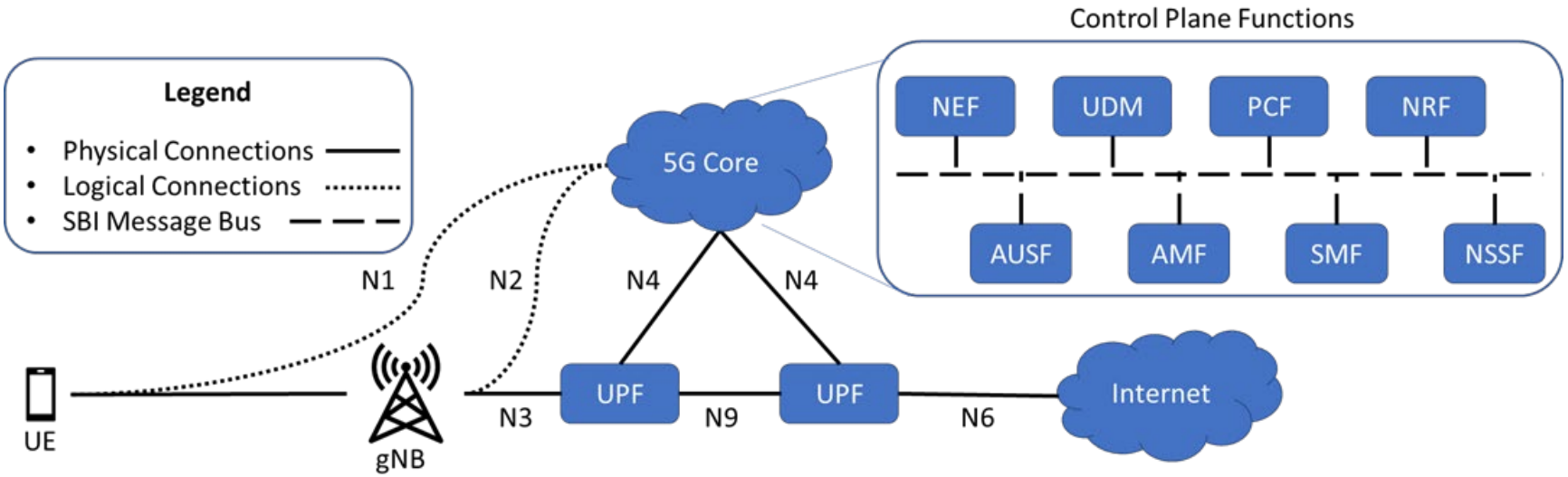

Figure 3. 5G Network Architecture. Source: [12].

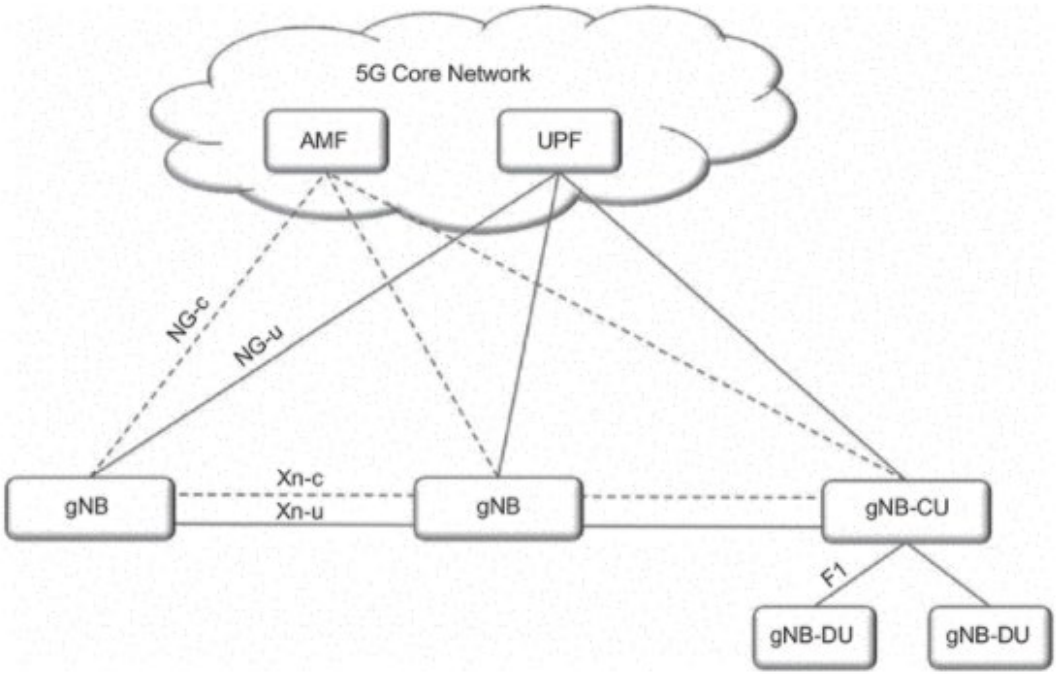

Figure 4. 5G Radio Access Network. Source: [12].

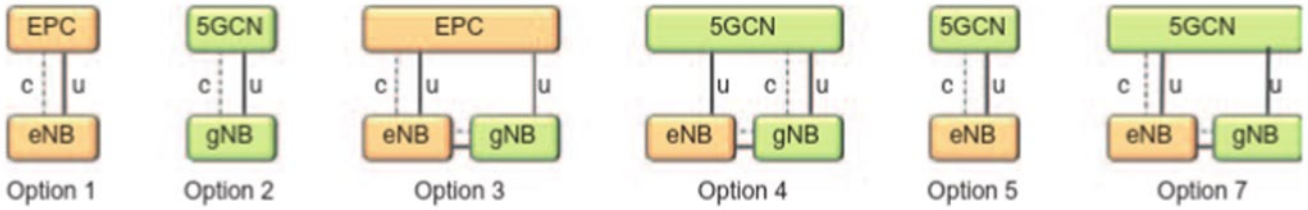

Figure 5. 4G and 5G RAN Configurations. Source: [12].

4. User Equipment and Base Station Link

The link between the UE and BS, whether it is an eNB or gNB is called the UU link. Starting in the 3G standard, this link has been encrypted. For each generation of standards, the UU link has a series of technical specifications that cover the acceptable frequency deviation and synchronization times. The frequency deviation specification shown in Table 1 covers 3G, 4G, and 5G.

Table 1. Base Station Frequency Deviation Specifications

Base Station Class	Required Accuracy
Wide Area BS	+/- 0.05 ppm
Medium Range BS	+/- 0.1 ppm
Local Area BS	+/- 0.1 ppm
Home BS (not in 5G standard)	+/- 0.25 ppm

Adapted from [14], [15], and [16].

In the 4G and 5G standards, a time alignment error (TAE) is defined as the average frame timing difference between any two transmitters on different transmit antenna connections [15], [16]. The maximum permitted TAE for 4G and 5G is given in Table 2.

Table 2. Time Alignment Error Specification

Operating Mode	4G LTE	5G NR
MIMO or Transmitter Diversity	≤ 65 nsec	≤ 65 nsec
Intra-band contiguous carrier aggregation	≤ 130 nsec	≤ 260 nsec
Intra-band non-contiguous carrier aggregation	≤ 260 nsec	≤ 3 μsec
Inter-band carrier aggregation	≤ 260 nsec	≤ 3 μsec

Adapted from [15] and [16].

Additionally, some services provided by the base station can utilize GNSS or provide services that complement GNSS. For example, the 3G standard employs TDD Node B synchronization ports, where at least one Node B in each cell is synchronized by an external reference such as GPS [17]. In 4G, the LTE Positioning Protocol (LPP) is used by a UE to

determine its location through position-related measurements on one or more reference sensors [18], [19]. Finally, 5G specifies requirements for Assisted-GNSS (A-GNSS), which reduce the time to first fix by providing the UE with satellite ephemeris data [20].

B. TIMING BASICS

1. Synchronization Methods

a. Network Time Protocol

Network Time Protocol (NTP) is a one-way clock synchronization protocol capable of maintaining time within a few milliseconds. It has numerous software implementations such as Simple Network Time Protocol (SNTP), Windows Time, and chrony.

b. Precision Time Protocol

Precision Time Protocol (PTP) is a two-way clock synchronization protocol capable of maintaining timing accuracy less than 15 nanoseconds. It employs the best master clock (BMC) algorithm to determine which clock source to select for a particular domain and network segment. PTP is widely implemented in routers, switches, integrated circuits, standalone solutions, and software stacks.

c. Global Navigation Satellite Systems

GNSS broadcast timing messages over the air that can be used to maintain timing accuracy within 15 nanoseconds. Additionally, GNSS can be used to determine the location of a receiver. GNSSs are discussed further in Section II.C.

2. Types of Synchronization

Synchronization is a necessary part of any communications system. The receiver must be able to determine and compensate for differences between design and reality. These include frequency offsets, phase offsets, and time offsets. Figure 6 illustrates the different types of offsets, and the possible effects. Different communications systems require different levels of synchronization to function properly.

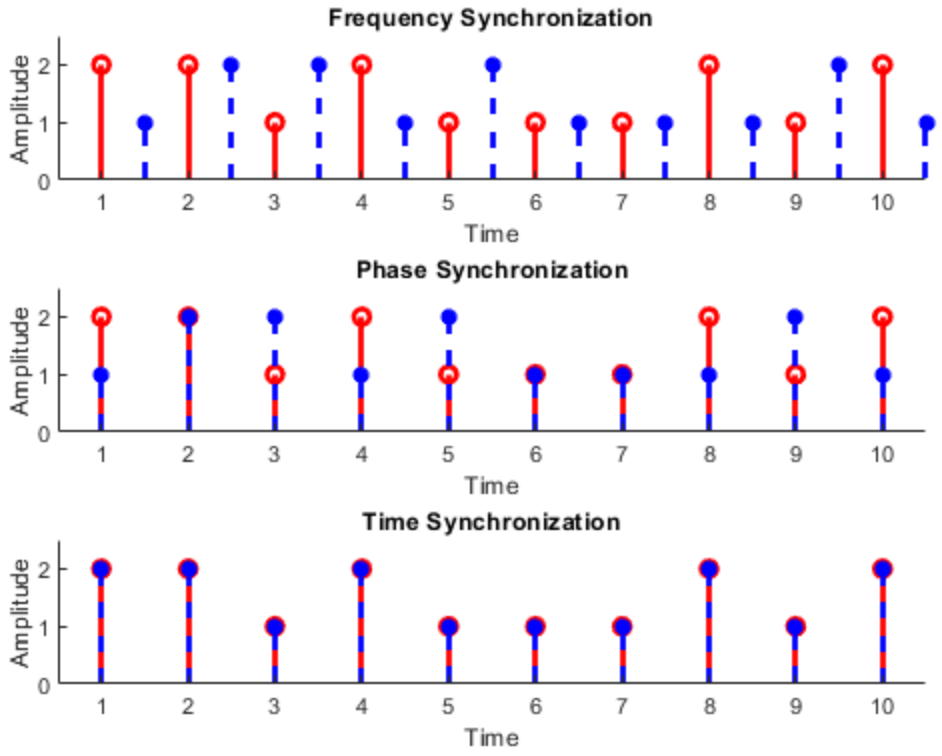

Figure 6. Types of Synchronization

Two nodes that have achieved frequency synchronization exhibit the same time delay between matching points of the signal. Frequency synchronization compensates for Doppler shift between nodes and any instabilities or inaccuracies in oscillators. Although the nodes are synchronized in frequency, they are not necessarily synchronized in time or phase. Examples of standards that can provide frequency synchronization include E1/T1, SyncE, PTP, NTP, GNSS, 10 MHz, and 1PPS.

Phase synchronization is then established when the two nodes receive information at the same instant in time, but not referenced to a common epoch. Frequency synchronization must be established before phase synchronization can occur. PTP, NTP, 1PPS, and GNSS can also be used to establish phase synchronization.

Finally, time synchronization is achieved after frequency and phase synchronization when information is referenced to a common epoch. Standards that can achieve time synchronization are NTP, PTP, and GNSS.

3. Evolution to GNSS Functionality in Cellular Chips

Major chipmakers produce modems for 4G and 5G that can be purchased as an individual circuit chip, or part of an integrated system on a chip (SoC) solution. Figure 7 shows the components within a Qualcomm radio frequency front end (RFFE) solution used in the Snapdragon line of modems and processors. A SoC solution, such as the Snapdragon system-in-package solution shown in Figure 8, typically includes an RFFE, various filters, and interfaces to external devices and features such as WiFi or GNSS. A brief analysis of 5G products available in the marketplace, shown in Figure 8, reveals the prevalence of GNSS functionality in commercial SoC solutions.

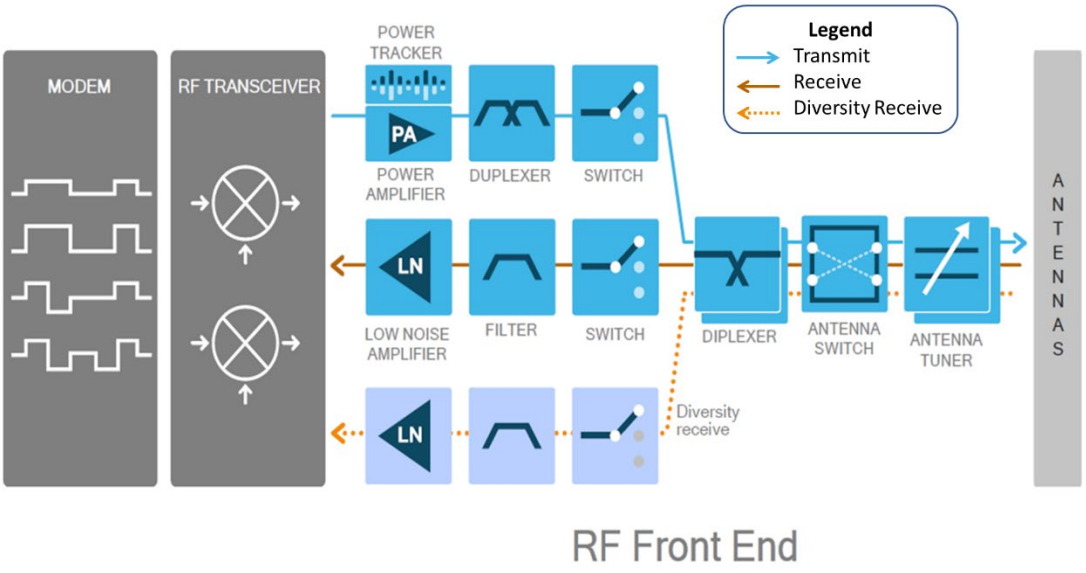

Figure 7. Qualcomm Radio Frequency Front-End Structure. Adapted from [21].

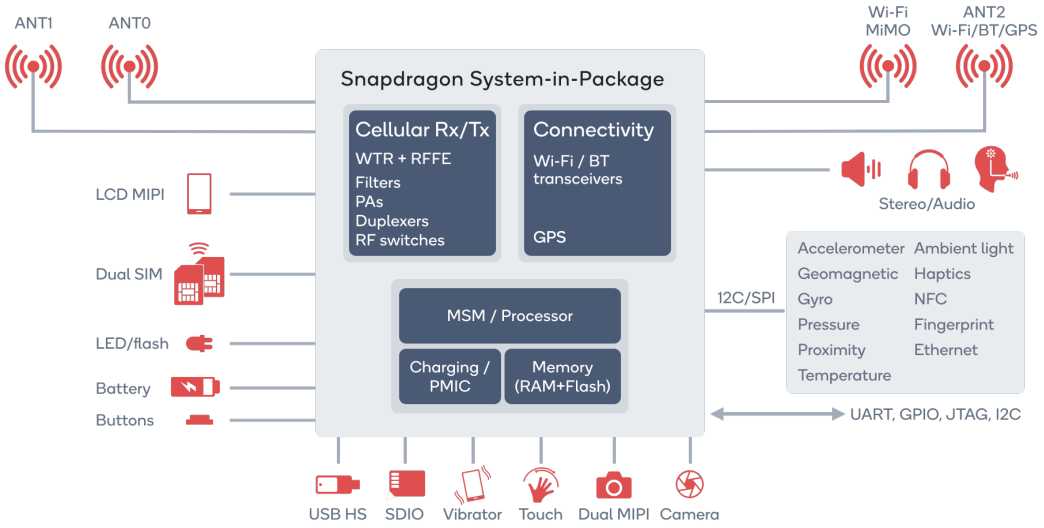

Figure 8. Qualcomm Snapdragon Functionality. Source: [22].

Table 3. Market Analysis of 5G SoC

Company	Product Line	GNSS?	Constellations & Freq Bands
Huawei	Kirin 980	Yes	GPS, Glonass, BeiDou
MediaTek	Dimensity 1000 series	Yes	not specified
MediaTek	Dimensity 820 series	Yes	GPS, Glonass, BeiDou, Galileo
Quectel	RG500Q	Yes	GPS, Glonass, BeiDou, Galileo, QZSS
Samsung	Exynos 980	Yes	GPS, Glonass, BeiDou, Galileo, QZSS, NavIC
Samsung	Exynos 1080	Yes	GPS, Glonass, BeiDou, Galileo
Samsung	Exynos 2100	No	N/A
Sierra Wireless	EM9190	Yes	GPS (band L1 and L5), Galileo, Glonass, Beidou
Sierra Wireless	EM9191	Yes	GPS (band L1 and L5), Galileo, Glonass, Beidou
Telit	FN980	Yes	GPS (band L1 and L5)
Telit	FN980m	Yes	GPS (band L1 and L5)

C. GNSS AND PNT

1. History and Overview

The discovery of electromagnetic waves by Maxwell and subsequent demonstration of wireless communications by Marconi have fascinated scientists and scholars alike. By the end of World War II, the United Kingdom and United States had both developed radio navigation systems. Gee, the U.K. system, was accurate to a few hundred meters and covered a range of approximately 350 miles. LORAN (an abbreviation for long range navigation) was designed by the U.S. to cover a large portion of the Atlantic Ocean. It was accurate to tens of miles and covered a range of approximately 1500 miles.

In the 1960s, the U.S. deployed the first satellite navigation system, known as Transit. The primary purpose of this system, also referred to as NAVSAT, was to provide location services for ballistic missile submarines during the Cold War. Satellite coverage was not complete, however, and the quality of the positioning afforded by the system depended on the ability of the receiver to accurately measure the Doppler shift of the satellite.

The GNSS were launched in 1978 and 1982 by the United States and the U.S.S.R. respectively. The Global Positioning System (GPS), created by the U.S., achieved global coverage in 1994. Global'naya Navigatsionnaya Sputnikovaya Sistema (GLONASS) achieved global coverage for the Russian Federation in 1995. Since then, two other GNSS systems have been launched: BeiDou by the People's Republic of China in 2000, and Galileo by the European Union in 2011. BeiDou achieved continuous global coverage on 23 June 2020. Galileo is scheduled to achieve continuous global coverage in 2021 after an additional two satellite vehicles complete on-orbit testing [23].

In addition to global navigation satellite systems, regional navigation satellite systems (RNSS) exist to provide a localized PNT solution. Two RNSS are currently deployed, Navigation with Indian Constellation (NavIC) covers India and Quasi-Zenith Satellite System (QZSS) covers the Japanese islands. Further augmentation of GNSS constellations can be accomplished via satellite-based augmentation systems (SBAS), ground-based augmentation systems (GBAS), or aircraft-based augmentation systems

(ABAS). Some examples of SBAS include Wide Area Augmentation System, Differential GPS, and Multifunctional Satellite Augmentation System. A map of SBAS coverage for various parts of the world is shown in Figure 9.

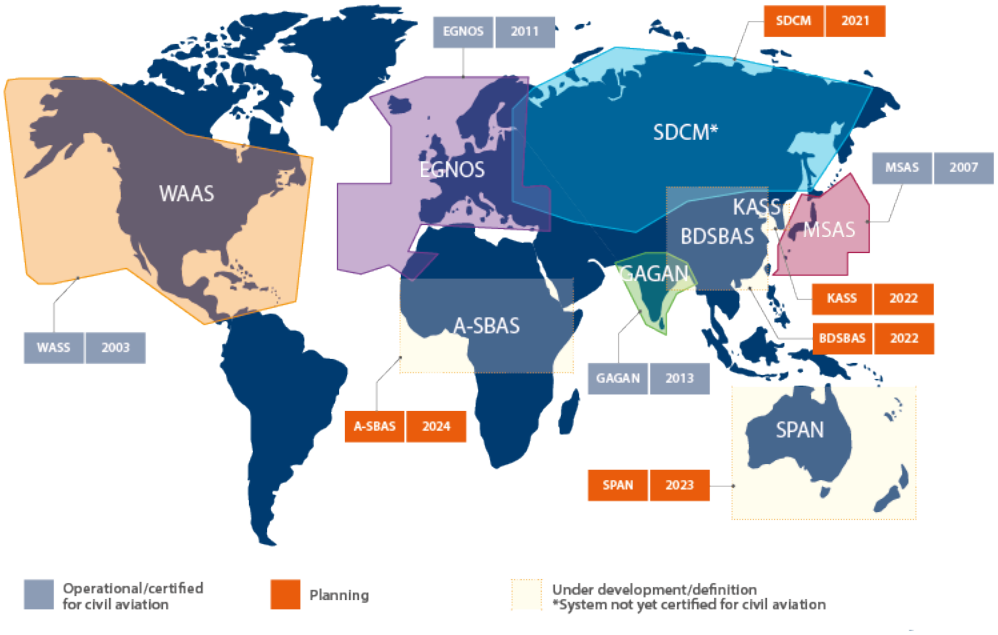

Figure 9. Global Coverage of Satellite-Based Augmentation Systems (SBAS). Source: [24].

GNSS offers three major services to users: positioning, navigation, and timing (PNT). The timing accuracy of the GNSS constellations is particularly useful in modern applications such as banking, networking, manufacturing, and communications. Augmentations to the GNSS are designed to improve positioning accuracy, reduce the amount of time required to obtain the first satellite fix, or make the PNT services more resilient by incorporating multiple GNSS constellations.

2. Theory of Operations

GNSS receivers determine their three-dimensional location through trilateration. To unambiguously determine the location of the receiver, the receiver must also compensate for clock offset at the receiver. The receiver, therefore, must solve a set of four

equations with four unknown variables (one for each of the three-dimensions of position, and another for local clock offset). Thus, one of the requirements for a GNSS constellation is that the satellite orbits are designed such that at least four of the satellites are observable by any receiver at any time. The receiver calculates the distance, d, to each of four satellites in the GNSS constellation using the speed of light and the amount of time it took for the signal to travel from the satellite to the receiver, τ, as shown in (1).

$$d = c \times \tau \tag{1}$$

The sequence transmitted by each satellite must be known so that it can be compared to a local copy at the receiver. The sequence also must be sufficiently long so that it does not repeat within the amount of time the signal propagates from the satellite to Earth.

3. Global Positioning System

The United States began launching satellites for the Global Positioning System (GPS) in February 1978. The constellation of 24 satellites was completed in 1993. Since then, additional satellites have been launched to replace failing satellites and improve the precision of the constellation by adding transmission capability in the L5 band. Today, the GPS constellation contains on-orbit spares in addition to the 24 satellites required for PNT.

The satellites are divided into six orbital planes with an inclination of 55° and are offset by 60° in the right ascension of the ascending node (RAAN) from one another. Each orbital plane contains four satellites in medium Earth orbit (MEO) at approximately 20,200 km altitude. The satellites are not evenly spaced within each orbital plane but organized so that at least six satellites are in the line of sight to any receiver on Earth at any given time.

Each satellite transmits a unique pseudo random noise (PRN) code, which allows the receiver to distinguish between satellites. For the L1 coarse acquisition (C/A) code, the PRN sequence is 1023 bits long. The atomic clock onboard each satellite creates a reliable signal at 10.23 MHz which is used to generate the C/A code at 1.023 Mbps. As a result, the C/A code repeats every 1 millisecond.

The PRN code is generated by summing two generator polynomials, one of which is delayed by a number of chips unique to each satellite, to create a Gold code (after Robert Gold), as shown in Figure 10. The delay value, n, used for each satellite is given in Appendix A. Alternatively, taps from the linear feedback shift register can be used instead of a delay block. The required taps for each satellite are also given in the appendix. The Gold codes are designed so that any two codes have approximately no cross-correlation and that each code is almost uncorrelated with itself except at zero lag.

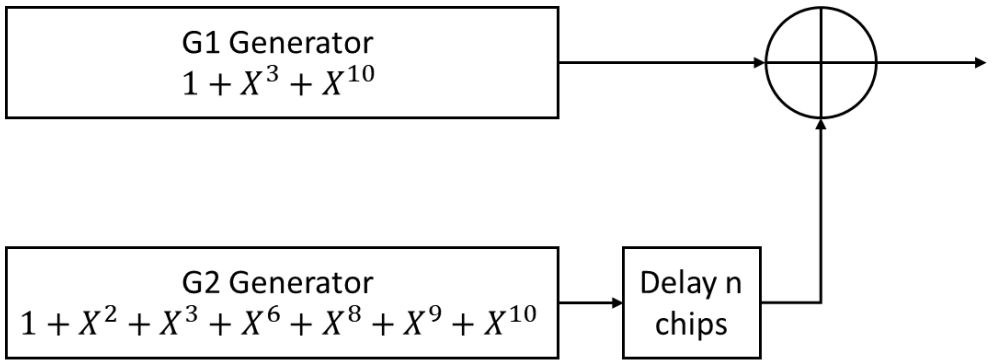

Figure 10. PRN Code Generation for GPS Satellites

Once generated, the Gold code is XORed with navigation data and used to modulate the L1 carrier frequency via binary phase shift keying (BPSK). Navigation data is transmitted on the L1 frequency at 50 bits per second and consists of five 300-bit subframes that each last 6 seconds. The last two subframes contain satellite almanac data that is cycled over 25 frames before it is repeated. Altogether, the navigation data repeats every 12.5 minutes. The navigation data structure is shown in Figure 11.

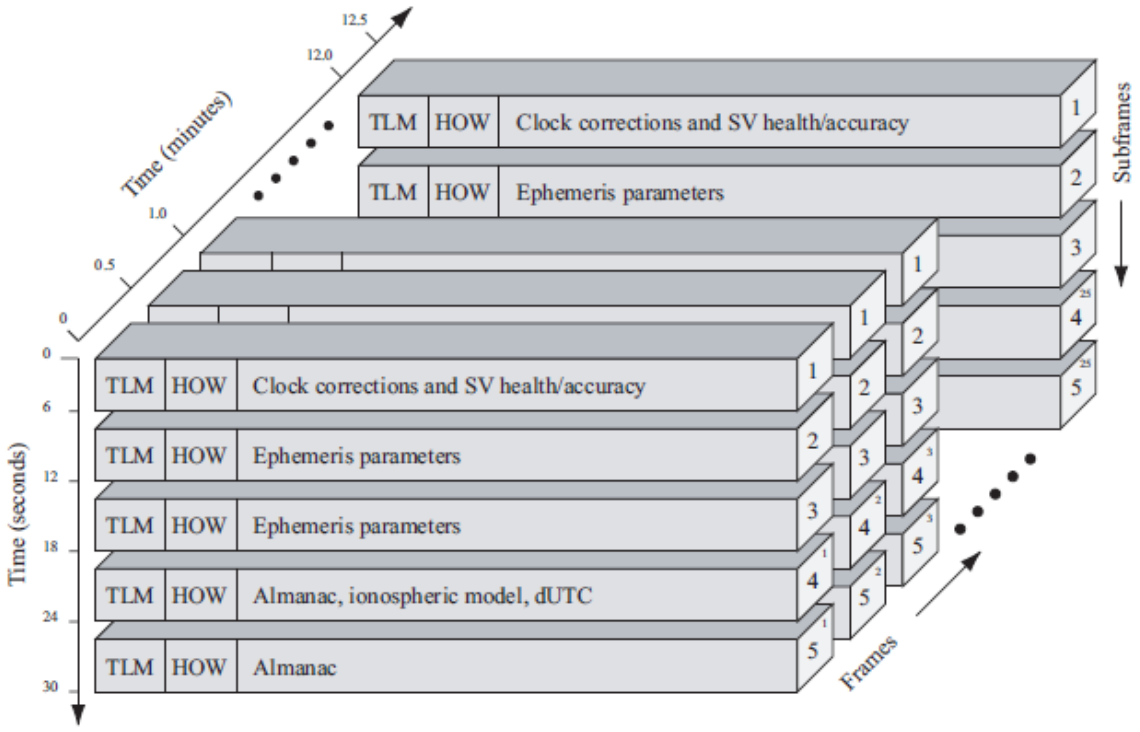

Figure 11. GPS Navigation Data Frame Structure. Source: [25].

Each subframe contains telemetry data (TLM) and a handover word (HOW). Subframe one includes clock information used to determine when the navigation message was transmitted by the satellite. Subframe two and three include orbital information for the transmitting satellite that is used in position calculation. Subframes four and five contain ephemeris data for all the GPS satellites in service, but at reduced precision.

Another code, the encrypted P(Y) code is generated and XORed with the navigation data. The result is also used to modulate the L1 carrier via BPSK, however, this resultant constellation is orthogonal to the C/A code constellation. The two signals are transmitted together on the L1 carrier frequency. All GPS satellites transmit the C/A and P(Y) codes on L1 and P(Y) again on L2. Block IIF and newer GPS satellites transmit a new signal on L5 consisting of an in-phase and quadrature-phase code, I5 and Q5, respectively. Block III satellites transmit the L1C signal, which is designed to be backward compatible with the C/A code, as well as a military (M) code on L1 and L2.

4. Major Systems Comparison

Table 4 compares the capabilities of the various GNSS and RNSS constellations currently in service. The precision of each system depends on a variety of factors, including which of the frequency bands are being utilized (public or encrypted), receiver tolerances, atmospheric conditions, and the use of multiple constellations or external timing corrections.

Table 4. GNSS and RNSS Comparison

System	Host(s)	Satellite Parameters	Frequency Bands	Channel Access	Altitude	Revolutions / Sidereal Day
GPS	United States	24 MEO 4 per 6 orbital planes 6 additional orbiting spares	L1 1.57542 GHz L2 1.22760 GHz L5 1.17645 GHz	CDMA	20,180 km	2
GLONASS	Russia	24 MEO 8 per 3 orbital planes	G1 1.593–1.610 GHz G2 1.237–1.254 GHz G3 1.189–1.214 GHz	CDMA & FDMA	19,130 km	17/8
BeiDou	China	24 MEO, 3 IGSO, 1 GSO	B1/B1-2 1.561098/1.589742 GHz B2 1.20714 GHz B3 1.26852 GHz	CDMA	21,150 km	17/9
Galileo	European Union	24 MEO 8 per 3 orbital planes 6 additional orbiting spares	E1 1.57542 GHz E5 1.191795 GHz E6 1.27875 GHz	CDMA	23,222 km	17/10
QZSS	Japan	3 GSO, 1 GEO	L1C/A,L1C,L1S 1.57542 GHz L2C 1.22760 GHz L5,L5S 1.17645 GHz	CDMA	32,600 - 39,000 km	1
NavIC	India	3 GEO, 5 GSO MEO	L5 1.17645 GHz S 2.492028 GHz	CDMA	36,000 km	1

5. Multi-constellation Receivers and Real-Time Kinematics

Using a single frequency from a GNSS constellation can have long acquisition times. To solve this problem, many receivers implement additional features, such as pulling almanac data from other sources, using two or more frequency bands to improve location resolution, or using multiple GNSS constellations. As discussed in Section II.A.4, GNSS receivers are used in conjunction with mobile telephony services such as A-GNSS or LPP.

Another way to improve the resolution of GNSS constellations is to use carrier-phase enhancement to perform real-time kinematic positioning (RTK). By tracking changes in the phase of the carrier signal (at roughly 1 GHz) rather than the overlayed data signal (at roughly 1 MHz), centimeter-level precision can be obtained by specially equipped receivers. This technology will play an important part in autonomous vehicles and their communications via vehicle-to-vehicle or vehicle-to-everything protocols.

D. UNIVERSAL INTEGRATED CIRCUIT CARDS

The subscriber information for mobile devices is stored on a universal integrated circuit card (UICC). The UICC contains an integrated circuit that runs applications programmed using the Java Smart Card toolkit. Subscriber information is stored within the SIM application for 2G networks or the USIM application for networks starting with 3G. Additional applications include billing, mobile banking, and additional storage for phone numbers or text messages.

Within the USIM application, there are several dedicated files (DF) stored as offsets from the master file (MF). A DF may contain sub-directories, elementary files (EF) where information is stored, and application dedicated files (ADF) which are DFs related to a specific application. An EF can take several forms:

- Transparent: a fixed sequence of bytes
- Linear fixed: a sequence of up to 255 records
- Cyclic: a sequence of records where the last record is linked back to the first record

- Tag length value: a record containing a tag, length, and value

Certain DFs and EFs are mandatory within the USIM application and have specified memory locations, while other locations are reserved for usage by the UICC manufacturer or MNO. Section 4.7 of [26] contains two figures that depict the file structure of the UICC and USIM ADF (ADF_{USIM}), respectively. Figure 12 shows the UICC MF, EFs at the MF level, and some of the DFs. A list of applications loaded onto the UICC can be obtained by reading the records stored in EF_{DIR}. The UICC integrated circuit card identification number (ICCID) is stored in EF_{ICCID}. The ADF_{USIM} file system (not shown) contains EFs with the information such as the subscriber's IMSI, cryptographic keys, and mobile subscriber integrated services digital network (ISDN) number (MSISDN).

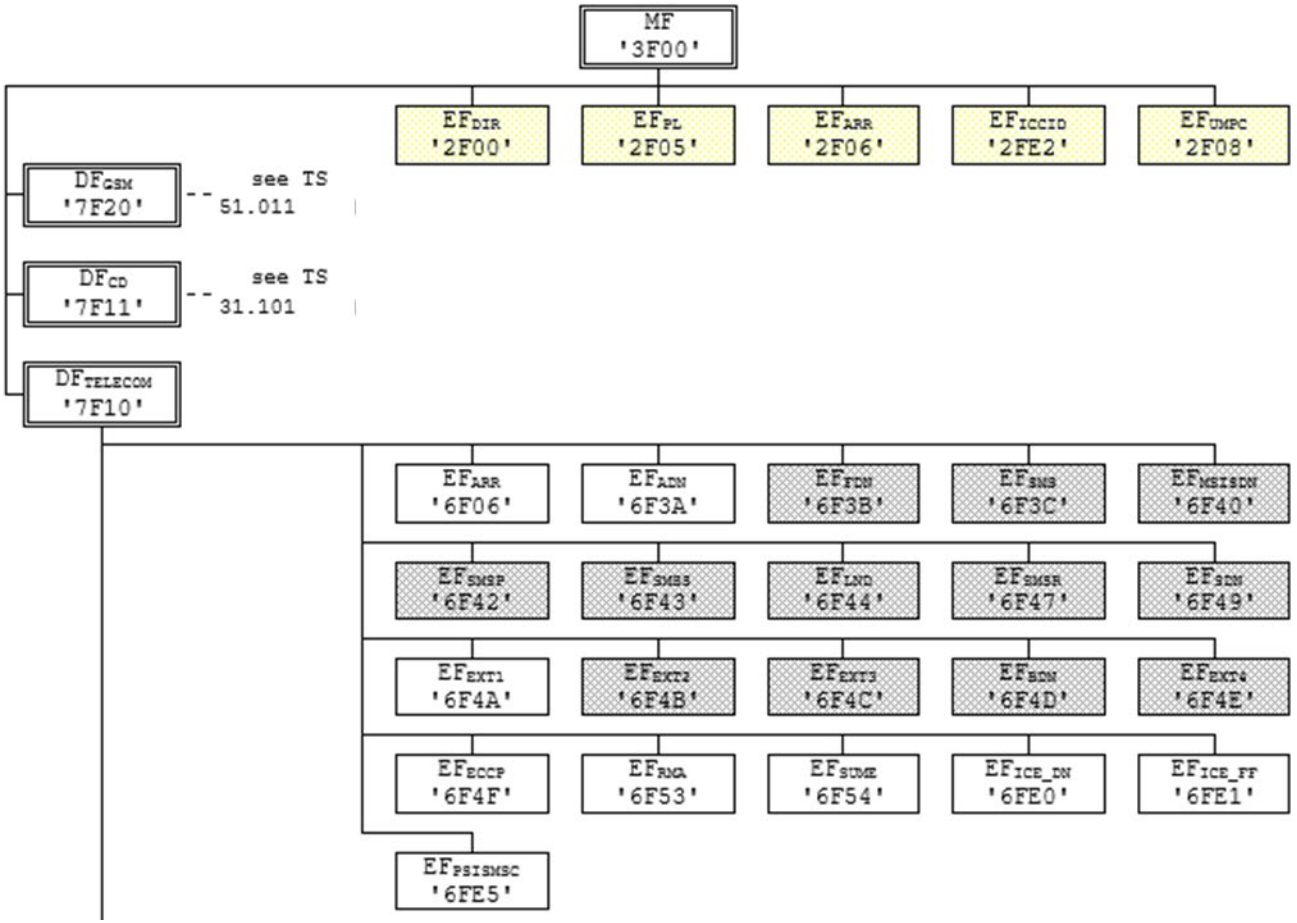

Figure 12. Overview of UICC MF Structure. Source: [26].

Each file in ADF_{USIM} has a set of access conditions for reading, writing, activating, or deactivating. For a given EF, the access conditions for each action tend to differ. The access conditions have levels zero through seven as described in Table 5. An example of the access conditions and structure listed in [26] for EFs is provided in Table 6 for the Access Point Name Control List EF (EF_{ACL}).

Table 5. USIM Access Conditions. Adapted from [27].

Level	Access Condition	Description
0	ALW	The associated action is always allowed
1	PIN	Pin level 1 must be verified to perform the action
2	PIN2	Pin level 2 must be verified to perform the action
3,4	RFU	Permission levels set aside for future use
5,6	ADM	Administrator pin required
7	NEV	The associated action is never allowed

Table 6. EF for Access Point Name Control List. Adapted from [26].

<table>
<tr><td colspan="2">Identifier: ‘6F57’</td><td colspan="2">Structure: transparent</td><td>Optional</td></tr>
<tr><td colspan="3">File size: X bytes (X>1)</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ PIN
UPDATE PIN2
DEACTIVATE ADM
ACTIVATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td colspan="2">Number of APNs/DNNs</td><td>M</td><td>1 byte</td></tr>
<tr><td>2 to X</td><td colspan="2">APN/DNN TLVs</td><td>M</td><td>X-1 byte</td></tr>
</table>

III. EQUIPMENT AND NETWORK SETUP

This chapter details the hardware and software utilized, and the various network topologies employed, during the phased implementation of an open stack network. First, the UE rationale and capabilities are discussed. Then, the open stack network is introduced, to include selection of software and hardware, available software utilities, and network topologies employed. A detailed description of the steps required to attach the UE to the open stack network is given, which includes subscriber credential provisioning, network provisioning, and execution of modem-level commands. Finally, an overview of some additional software used throughout the work is provided.

A. QUECTEL EC20

The COTS modem selected for this thesis was the EC20 model from Chinese telecommunications company Quectel. This modem product line is LTE capable and is advertised for machine-to-machine and Internet of Things applications. Figure 13 shows a Quectel EC20 modem without the protective metal covering over the internal modem components. At the top of the image are three connectors for the main, GNSS, and receiver diversity antennas. Figure 14 displays the EC20 modem used for this thesis housed in a developer board with adapters connected to the antenna ports. The developer board facilitates communications between the modem and a host computer via USB. Additionally, the developer board provides interfaces for microphones and headphones to operate with the modem.

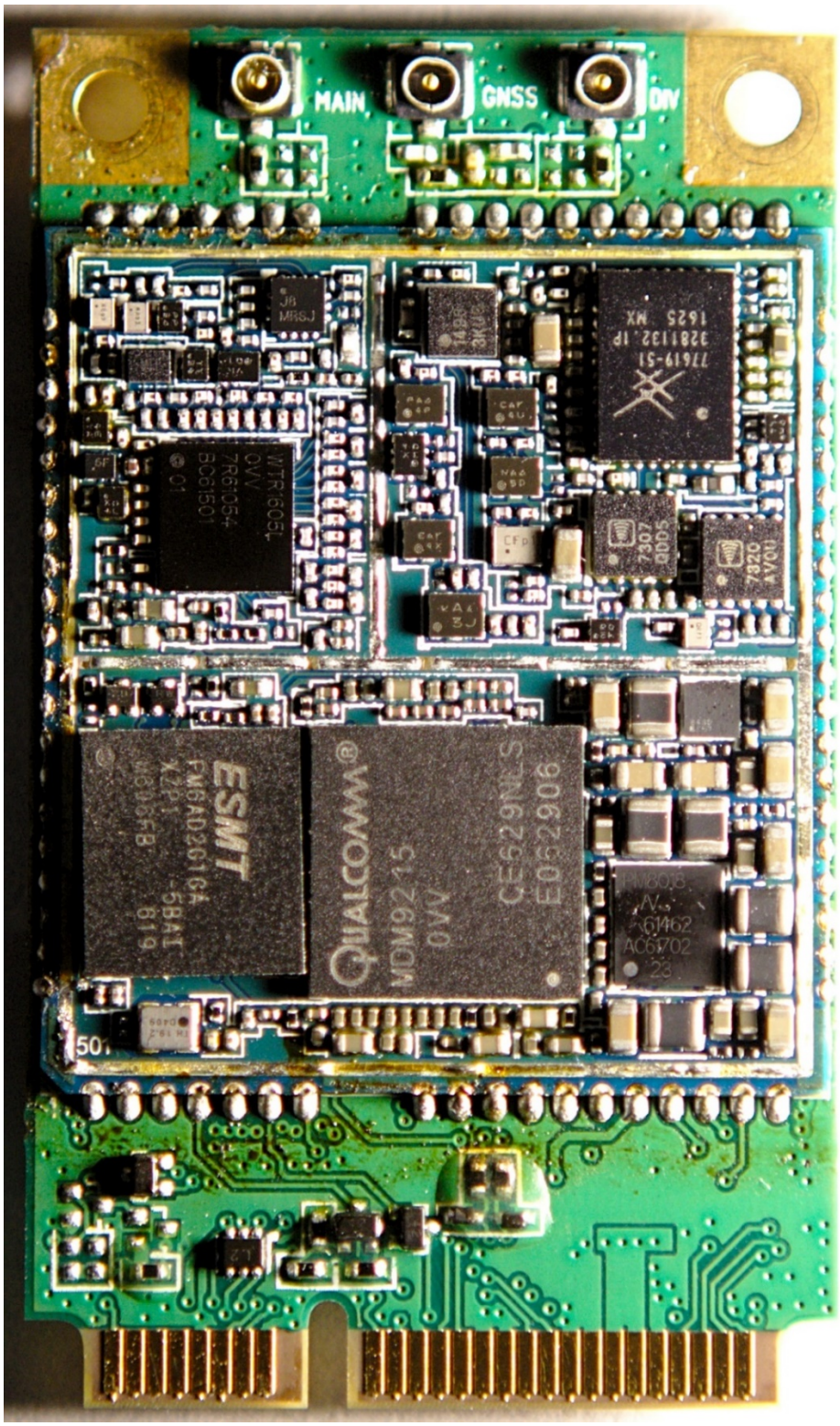

Figure 13. Quectel EC20 Modem without Protective Covering. Source: [28].

Figure 14. Quectel EC20 Modem in Developer Board

1. Modem Selection Rationale

The EC20-CE modem was chosen as our COTS UE for several reasons. Most importantly, the EC20 product line contains Qualcomm chips identified in the GNSS and component-level vulnerabilities discussed in [1]–[4]. While EC20 regional variants and revisions contain different Qualcomm chips, including the MDM9215, MDM9615, or MDM9607, these chips are all vulnerable to multiple component-level CVEs. The variant chosen, the EC20-CE, contains additional provisions in the firmware to region lock the device to China and India. The use of international bands not deployed in the United States enabled wireless testing of the device without interfering with commercial networks.

Secondarily, the EC20 modem is low-cost at $50. Several were already available from previous projects. This ensured that the project would not stall if a few modems were destroyed during the component identification or configuration. Additionally, with several modems available, a network could be configured with multiple UEs to test functionality such as phone calls over voice or data circuits, text messaging, and data transfer.

2. Device Specifications

The variant used in this thesis was the EC20-CE in a miniPCIe form factor inserted into a Quectel developer board. The LTE frequency bands in which the modem can operate differed from those provided in several manufacturer brochures, likely due to changes in regional variant and device revisions. The actual capabilities of the modem were found by executing the command:

AT+QENG="band"

The result was then parsed to determine the LTE frequency bands in which the modem can operate as shown in Table 7.

Table 7. Quectel EC20-CE 4G LTE Frequency Band Capability

LTE Band	Mode	f [MHz]
1	FDD	2100
3	FDD	1800
5	FDD	850
7	FDD	2600
8	FDD	900
20	FDD	800
38	TDD	2600
39	TDD	1900
40	TDD	2300
41	TDD	2500

Additional device specifications of note include capability for 2x2 multiple-input and multiple-output (MIMO) on the downlink, GNSS services, and compliance with the 3GPP release 9 of the LTE standard. Due to the single transmit antenna and the 2x2 MIMO receiver capability, the EC20 is capable of up to 50 Mbps on the uplink and 100 Mbps on the downlink [29]. A more detailed listing of the EC20 revision 1 and revision 2.1 modem capabilities is provided in Appendix C. An exhaustive list of EC20 revision 2.1 hardware specifications, including supported protocols, power transmission capabilities, and functional and circuit diagrams is given in [30].

3. Hardware Teardown

Several sources have performed a hardware teardown of the EC20 product to identify the internal components used by Quectel for any additional vulnerabilities, including [28] and [31]. Although differences exist in the Qualcomm chip used for the modem, as discussed in section III.A.1, the chips all come from similar product lines with similar vulnerabilities. These differences are likely the result of regional variants and hardware revisions to the EC20 modem. Figure 15 labels the major components within an EC20 modem and gives their functional role. The EC20 modem shown in Figure 13 is likely a revision 1 device, while the modem in Figure 15 is likely a revision 2.0 or 2.1 device.

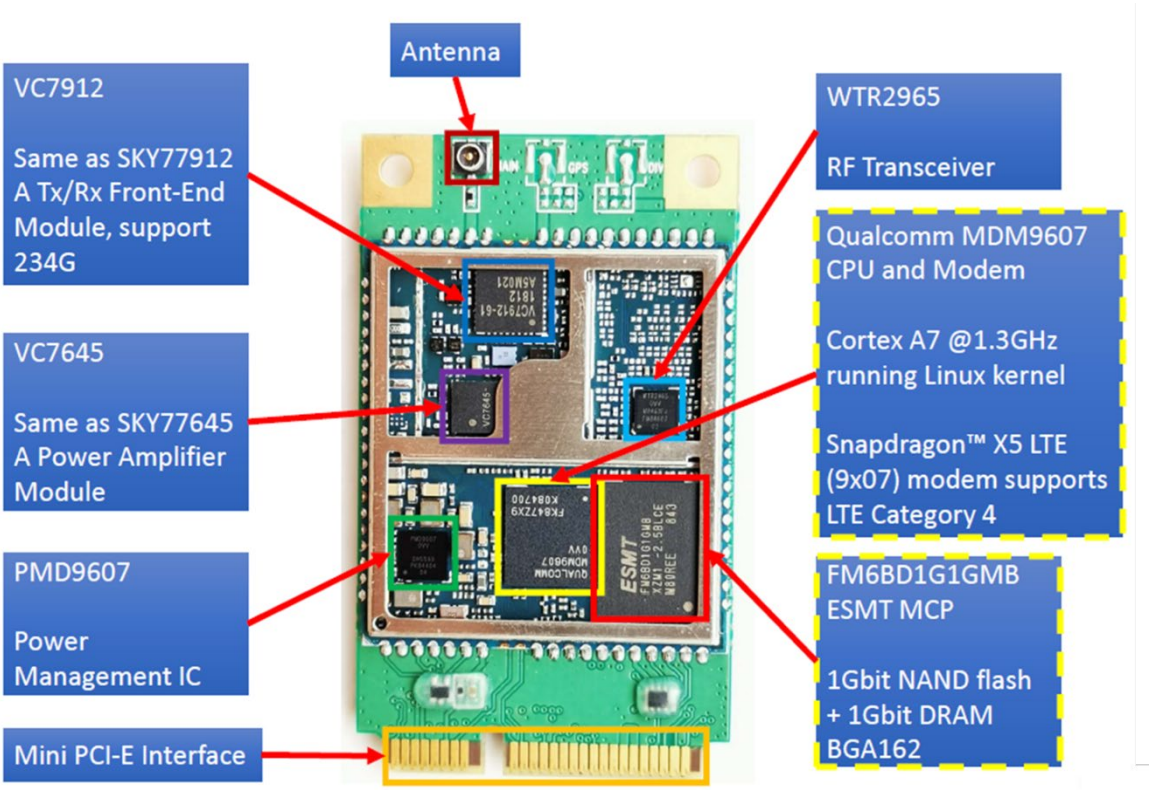

Figure 15. Hardware Teardown of EC20 Modem. Source: [31].

Figure 16 adapts a functional block diagram for the EC20 revision 2.1 modem to highlight how the integrated circuits identified during the hardware teardown interoperate.

The colors match between the two figures to facilitate comparison. Notably, the transmitter and receiver radio frequency front-end (RFFE) shown in a dark blue box in Figure 15 is indicated by a dashed dark blue box in Figure 16.

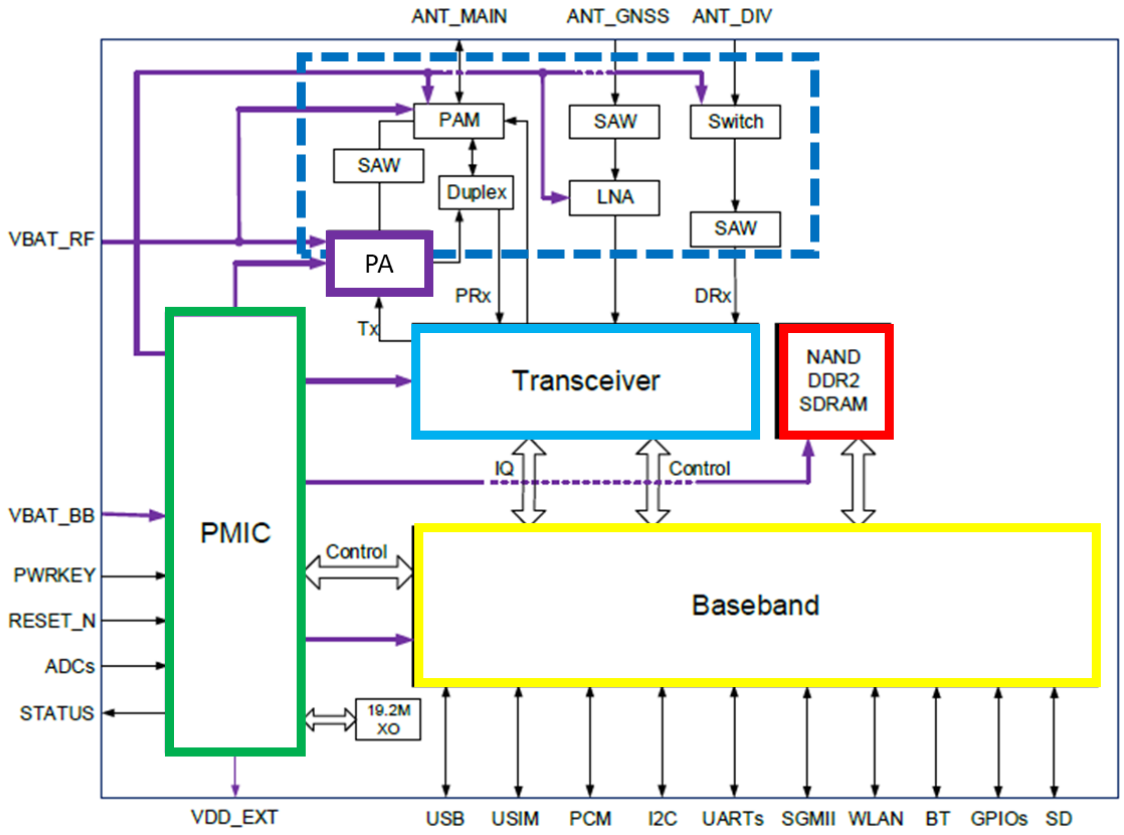

Figure 16. EC20 Modem Functional Diagram. Adapted from [30].

4. Software Teardown

The EC20 modem runs the factory firmware and a minimal Linux distribution based on the 3.18.20 kernel. Previous collaborative efforts between the NPS Electrical and Computer Engineering and Computer Science departments succeeded in pulling some of the system firmware. While most of these files are binaries that include Qualcomm proprietary code for the modem chip, the “strings” utility was able to pull some human-readable information from the files. This includes portions of the software used to load the

Qualcomm drivers and some additional modem-level commands not listed in any of the Quectel documentation.

5. Vulnerability Analysis

As previously discussed, multiple component-level vulnerabilities have been identified that could affect the EC20 modem. Each of the CVE reports for these component-level vulnerabilities were analyzed to determine the severity of the vulnerability and whether the EC20 modem was affected. Of the 21 vulnerabilities analyzed, seven had a criticality rating of 9.8, which corresponds to "critical." Each of these "critical" vulnerabilities affected both the MDM9207 and MDM9607 series chips except that CVE-2019-2276 did not affect the MDM9207 product line. Appendix B contains information for all 21 CVEs. Further research into Qualcomm component-level vulnerabilities related to GNSS functionality could reveal additional vulnerabilities that remain unpatched in the software installed on the EC20.

The high severity rating for CVE-2019-2254 provides reason to believe that "position accuracy may be degraded" is an understatement of the possible effects of exploiting the vulnerability. The other part of the vulnerability description, "wrongly decoded information," could refer to either malformed information from spoofed GNSS signals or external almanac files.

B. OPEN STACK NETWORK

An open stack network leverages commodity hardware and open-source software to provide a tool or testbed that is low-cost, easy to upgrade, and flexible. As mobile telephony standards progress, open stack networks can be upgraded by patching the software. Additionally, the network can be reprogrammed to offer more or fewer of the services described in the 3GPP standards. If the open stack network were to be operated outside of a Faraday cage, spectrum access would be required.

1. Hardware Selection Rationale

Since the open stack network concept should not require any specialized hardware, three modern laptops were obtained to run the CN, RAN, and UE. The laptops were

identical HP ProBook 640 G1s with Intel Core i7 processors and 16 GB of random-access memory running the Ubuntu 18.04 operating system. The processor in each laptop contains four physical cores divided into two logical cores each via hyperthreading for a total of eight cores.

For radio frequency communications, two Universal Software Radio Peripheral (USRP) B200 devices from Ettus Research were utilized. These software-defined radios (SDR) can be configured to operate from 70 MHz up to 6 GHz and cover 56 MHz of bandwidth. USB3 connectivity allows high data transfer rates to and from the device. Furthermore, the USRP B200 contains a reprogrammable Spartan6 FPGA that can be used to configure parts of the digital signal processing chain on the SDR. The USRP B200s utilized contain an add-on GPSDO module to provide precision timing, if needed. Full technical specifications for the USRP B200 are provided in Appendix D.

2. Software Selection Rationale

Various open source 4G networking projects were considered for use in this thesis, including FreedomFi, Open5Gs, OpenAirInterface, and SysLTE. While each project had its own merits, OpenAirInterface was chosen for its compliance with 3GPP standards and robust development practices. Additionally, the OpenAirInterface software provided the largest set of features. Each of the projects is briefly described.

a. FreedomFi

This project leverages an open-source distributed mobile packet project called Magma, created by the Connectivity sub-division of Facebook, to create a private 4G or 5G network. FreedomFi offers a gateway loaded with the Magma software for $300 [32]. This gateway interfaces between the private network and larger mobile network core that holds the billing and subscriber information. A complete open stack network using FreedomFi would still require an open-source RAN and likely some additional CN features. Customer support requires an account with FreedomFi and was not evaluated.

b. Open5Gs

This project implements a 4G and 5G CN in the C programming language that claims to be compliant with release 16 of 3GPP mobile telephony standards [33]. While the tutorials are incredibly detailed and span a variety of configuration options, closer analysis of the project source code on GitHub reveals that an overwhelming percentage of commits are by a single person. Furthermore, the project does not provide any software for the RAN.

c. OpenAirInterface

The OpenAirInterface (OAI) Software Alliance (OSA) focuses on three projects: a RAN, a CN, and a continuous integration and continuous deployment (CI/CD) system. These projects started as 4G LTE software and are progressing towards 5G NR solutions. For example, the EPC software utilizes containers to host each of the network functions as a steppingstone to a 5G CN. The OSA CI/CD project highlights the dedication of the team to delivering robust software. Compliance with 3GPP standards is provided for each project, along with timelines for future updates. OSA members include major industry players such as Qualcomm, Xilinx, Nokia, Inmarsat, Fujitsu, and Facebook Connectivity, as well as numerous universities and research institutions [34]. An active community surrounds the OAI team, but the separation of software projects and rapid development hampers documentation. While plenty of third-party resources exist, the information tends to be significantly outdated when compared to the latest stable branch on GitHub.

d. srsLTE

This project, recently renamed srsRAN to reflect a transition past 4G solutions, provides software for both the RAN and CN. Additionally, an application is provided to test UE functionality with an SDR instead of a COTS UE. The project is supported by organizations such as Analog Devices, National Instruments, Nokia, and the Massachusetts Institute of Technology, among others [35]. Getting started guides are available and easy to navigate, but do not reflect the same degree of adaptability to various usage cases exhibited by the OAI software. The CN software is not emphasized as much as the RAN

or UE software, suggesting a minimal level of functionality. Lastly, the CN software does not include any information on compliance with 3GPP standards.

3. OAI Software Projects and Utilities

The software provided by the OSA is divided into two projects: OpenAir5G and OpenAirEPC-FED. The first project, OpenAir5G, includes tools and utilities to test the RAN and network interfaces for both 4G and 5G. The second project, OpenAirEPC-FED, contains software used to build the 4G Evolved Packet Core (EPC).

a. OpenAir5G

This project focuses on utilities for simulating, testing, and implementing the RAN of 4G and 5G networks via SDR. Some of the major utilities packaged with this project include:

- Softmodem: an application used to simulate or run a 4G eNB or 5G gNB using an SDR as the radio front-end and parameters specified in a configuration file
- UE-softmodem: an application used to simulate a UE via SDR and a virtual SIM card with command line tuning parameters
- Virtual SIM programmer: a script used to generate a select subset of the parameters typically stored in the USIM application. This script must be used to generate valid credentials prior to attaching the UE-softmodem to an open stack network complete with EPC
- Virtual Oscilloscopes: a run-time utility that executes code programmed into the SDR FPGA to display real-time graphics for the wireless interface. The display includes time and frequency plots of the received signal, channel throughput, and a received constellation diagram as shown in and Figure 17 and Figure 18.

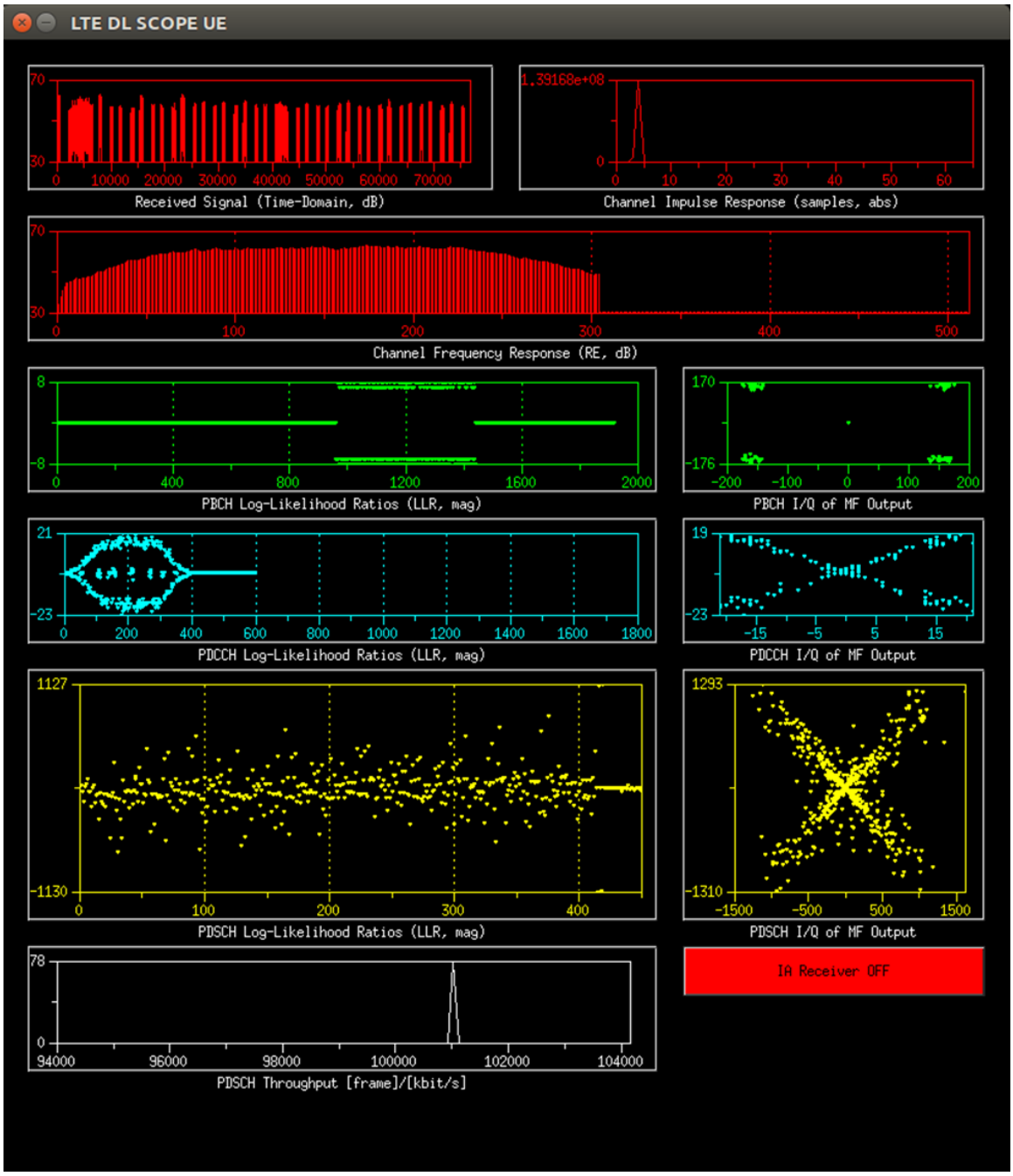

Figure 17. OAI UE Oscilloscope. Source: [34].

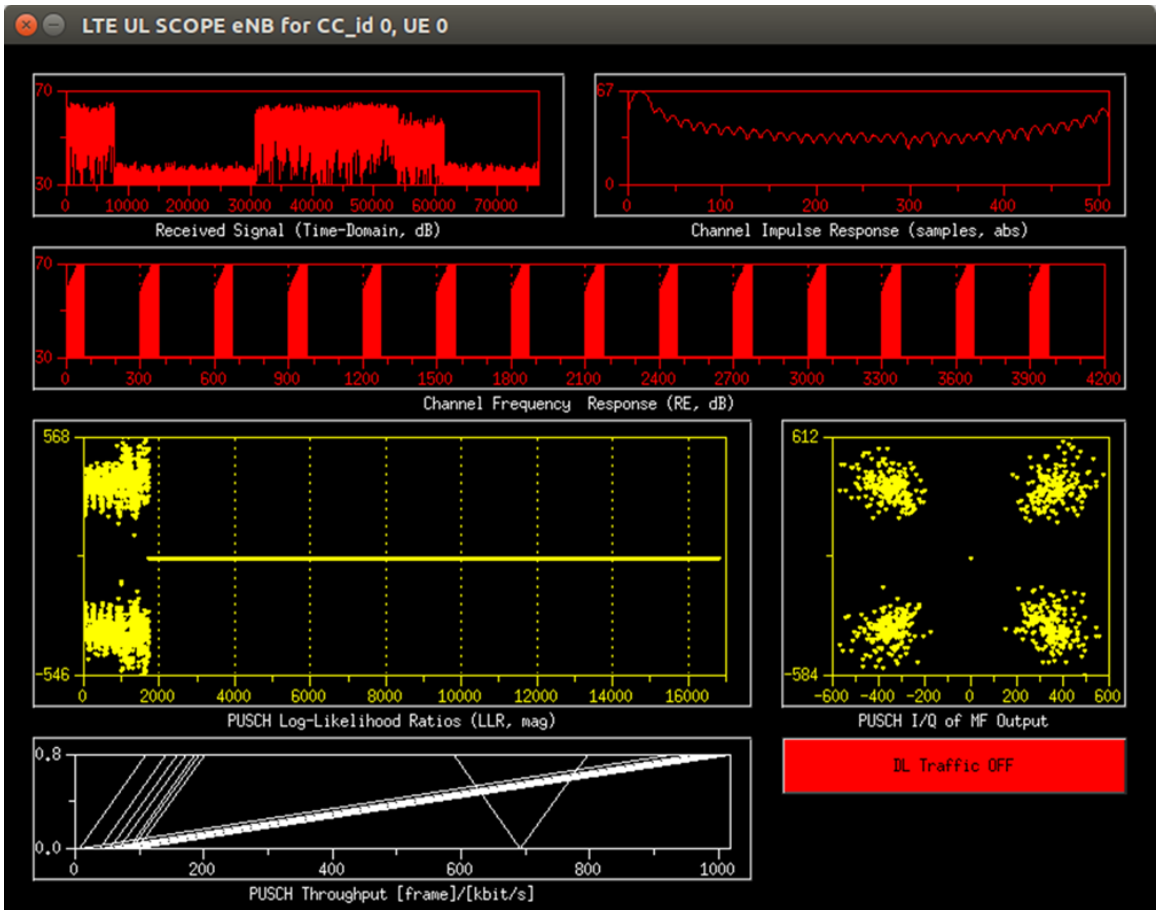

Figure 18. OAI eNB Oscilloscope. Source: [34].

- Simulators: numerous simulators are included so that almost every aspect of the network can be simulated. The core network, wireless link, and RF channel can all be defined virtually.
- T tracer utility: a utility that allows for debugging of selected packetized information at various levels of the wireless stack. This utility can also be used to monitor both physical layer parameters (like those provided in the oscilloscope) and traces at each of the higher levels in the wireless stack as shown in Figure 19.

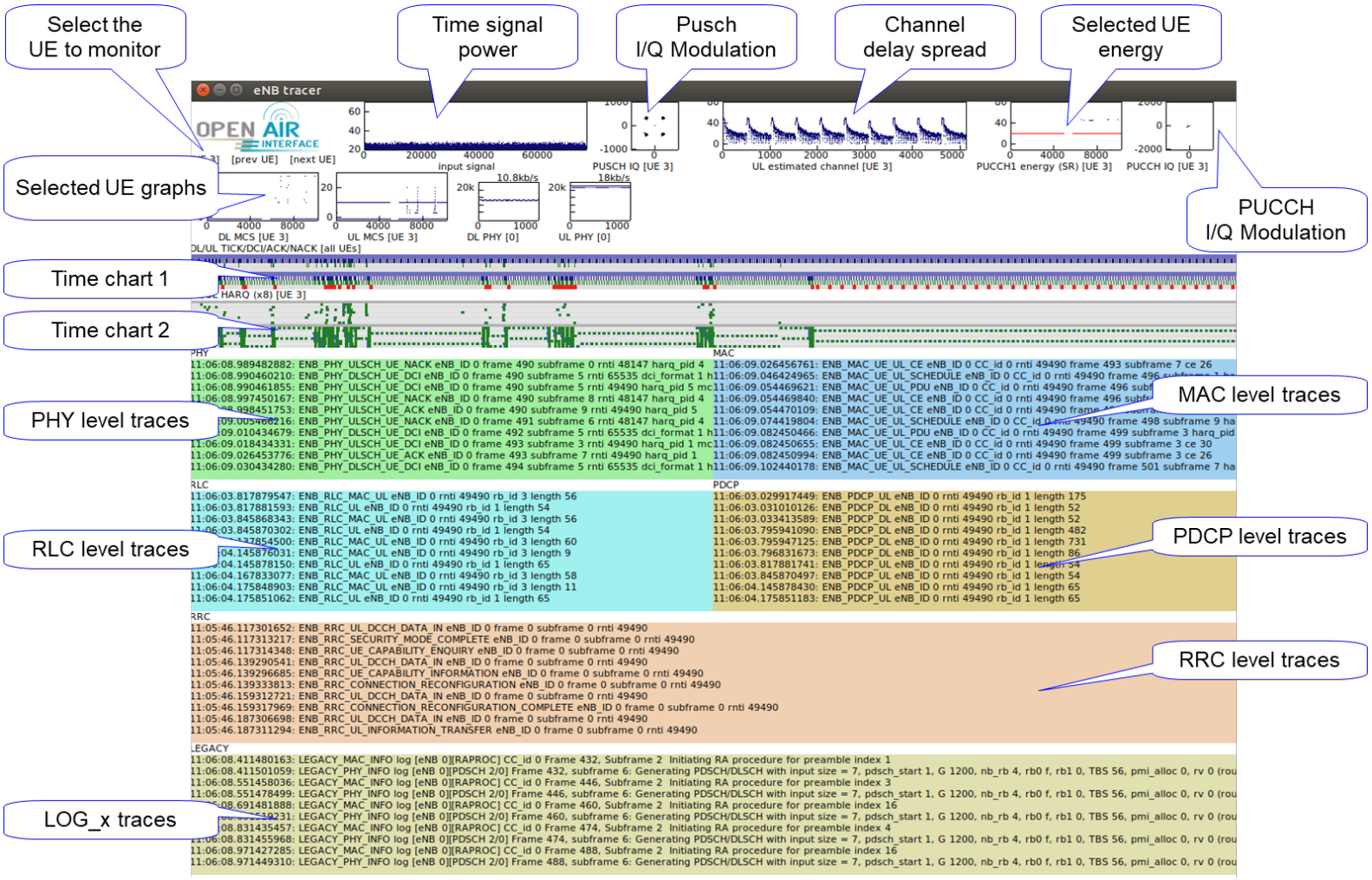

Figure 19. OAI eNB T Tracer Utility. Source: [34].

b. OpenAirEPC-FED

A separate project was created by the OSA to facilitate development of a 4G and 5G CN. Design choices for the 4G EPC were made to ease the transition to a 5G CN. For example, 4G EPC network functions are instantiated as separate docker containers connected via virtual interfaces.

4. Kernel Configurations

While the OAI RAN and CN are designed to run on modern computers without needing accelerators or high-end graphics cards for data processing, they do require some configuration of the computer kernel for optimal performance. Especially for the computer running the RAN, this configuration is necessary to prevent data overflows and dropped frames.

a. RAN Kernel Configuration

The RAN Kernel was configured using the recommendations on the OAI wiki for Ubuntu 16.04 as a guide, since the computers used for this work were configured with Ubuntu 18.04. The major steps required were to install the low-latency kernel, remove all power management features such as P-states and C-states, disable CPU frequency scaling, and disable hyperthreading. The commands to accomplish each step and verify the step was completed are now presented. While ultimately unnecessary, the BIOS for the RAN computer was also updated to the latest version using a removable thumb-drive configured for the HP laptops.

1. Install low-latency kernel

The low-latency kernel is installed using (2) and can be verified with (3). If installed correctly, then the system should display “low-latency” instead of “generic” attached to the kernel version number. If the system fails to boot into the low-latency kernel, then hold “shift” after the computer reboots and before the operating system (OS) loads the log-in page to enter the grub menu. In this menu, the low-latency kernel can be selected as the preferred kernel.

```
sudo apt-get install linux-image-lowlatency linux-headers-lowlatency          (2)
```

uname -a (3)

2. Remove all power management features

To disable p-states and c-states, the following command is added to the file /etc/default/grub before updating that boot options with “update-grub.”

GRUB_CMDLINE_LINUX_DEFAULT=“quiet intel_pstate=disable processor.max_cstate=1 intel_idle.max_cstate=0 idle=poll”

Then, to check that p-states and c-states are disabled, install the i7z utility with “sudo apt-get install i7z” and verify that only c-state c0 is utilized when “sudo i7z” is run.

Next, the Intel powerclamp module is disabled by adding “blacklist intel_powerclamp” to the end of the file /etc/modprobe.d/blacklist.conf. If the file does not exist, then an empty file with the same name should be made before adding the blacklist command.

3. Disable frequency scaling

The following commands are used to disable frequency scaling. First, install cpufrequtils with “sudo apt-get install cpufrequtils.” Then run “sudo systemctl disable ondemand” to prevent the CPU governor from switching modes. Next, set the governor with “sudo cpufreq-set -g performance” and restart the process with “sudo /etc/init.d/cpufrequtils restart.” If done successfully, then when “cpufreq-info” is run, all cores should show the governor as set to “performance.” Another check is to verify with “sudo i7z” that all cores are operating at the maximum frequency.

4. Disable hyperthreading

On the HP laptops, hyperthreading is disabled in the BIOS menu. During computer booting, press “F10” to enter the BIOS menu. Log in to the BIOS as an admin. This may require entering or making a password. For the RAN laptop, the BIOS password was set as “1qaz2wsx!QAZ@WSX.” Next, deep sleep was disabled so that the Intel HT (hyperthreading) technology could be toggled off.

b. CN Kernel Configuration

The CN Kernel configuration guide recommends installing the GTP kernel modules. However, this step is likely a relic of the EPC project before it shifted to using containers. If necessary, the GTP kernel modules can be installed with:

```
wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v4.8/linux-headers-4.8.0-040800-generic_4.8.0-040800.201610022031_amd64.deb

wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v4.8/linux-image-4.8.0-040800-generic_4.8.0-040800.201610022031_amd64.deb

sudo dpkg -i linux-headers-4.8.0-040800-generic_4.8.0-040800.201610022031_amd64.deb

sudo dpkg -i linux-image-4.8.0-040800-generic_4.8.0-040800.201610022031_amd64.deb
```

After rebooting, the kernel module and version can be checked with "uname -a," "sudo modprobe gtp," and "dmesg | tail."

5. Software Configuration

Each OAI project has a different method to configure and run the software. The RAN project is downloaded as a cloned repository from GitHub in which build scripts compile shared libraries to create executable files. The CN project is also downloaded as a cloned repository from GitHub, but it contains scripts for configuring containers built using the Docker platform. The steps to build each project are described in this section. Although the latest master branch was originally used for the open stack network, ultimately a recent developer branch was required in most cases.

a. RAN Software Build

Despite requiring a more complex kernel configuration, the RAN software build is simple and straightforward. A repository containing the 4G and 5G RAN is available on GitLab. If not already installed on the computer, git can be installed from the Ubuntu repositories via package manager with "sudo apt-get install subversion git cmake." Once git is installed, the repository is cloned with "git clone https://gitlab.eurecom.fr/oai/openairinterface5g.git." For this work, the 2020.w47 tag from the developer branch was

utilized. The user then navigates to the newly created directory with "cd openairinterface5g" and initializes some environment variables used by the software with "source oaienv." Next, the user navigates to the cmake directory with "cd cmake_targets." Finally, the user builds the desired RAN software with "./build_oai -I -w USRP --eNB --UE --buildlib enbscope --build-lib uescope." A summary of the flags used in the build script is provided in Table 8.

Table 8. OAI RAN Build Script Flag Summary

Flag	**Purpose**
-I	Install external dependencies using the package manager
-w	Select desired radio front-end (USRP, bladeRF, hackRF, etc)
--eNB	Build the eNB executable
--UE	Build the UE executable and associated utilities
--build-lib	Used to build additional supported libraries such as the oscilloscopes for the eNB and UE
-c	Erase previously built files for target (UE or eNB) before starting new build
-C	Erase previously built files for all targets before starting new build
-g	Build binaries with support for gdb
-h	Display help information for script and list complete flag summary

Analysis of the build_oai script reveals that all instances of the USRP hardware driver (UHD) are first removed and then the latest version from the Ettus Research repositories is installed in the function "check_install_usrp_uhd_driver." This can cause software mismatch errors that prevent the OAI RAN software from executing properly if a different version of UHD is installed from source.

The RAN software is executed from the "~/openairinterface5g/cmake_targets/ran_build/build" directory after first executing "source oaienv" from the "~/openairinterface5g" directory. A log file containing the information the eNB or UE script prints to the terminal window can be created by appending "| tee eNB.log" or "| tee UE.log" to the command used to execute the software.

b. CN Software Build

In contrast to the CN kernel configuration, the steps required to build the software are more complex. The OAI 4G LTE EPC uses the Docker platform to create individual containers for each of the EPC network functions. Docker containers are similar to virtual machines running on a host device, except they typically have less of the overhead and additional services provided by an OS. In the Docker environment, images are used to build containers, just like how OS image files (.iso) are required to build virtual machines.

The OAI EPC implementation differs slightly from the textbook EPC architecture by combining the S-GW and P-GW into a single entity labelled the SP-GW that is then separated into a control plane and user plane container. Figure 20 depicts a network diagram for the OAI EPC with labels for the interfaces between network functions. The middle box containing the control-plane aspects of the S-GW and P-GW is instantiated as a singled container called the SPGW-C in the software. The bottom box containing the user-plane aspects of the S-GW and P-GW is called the SPGW-U in the software.

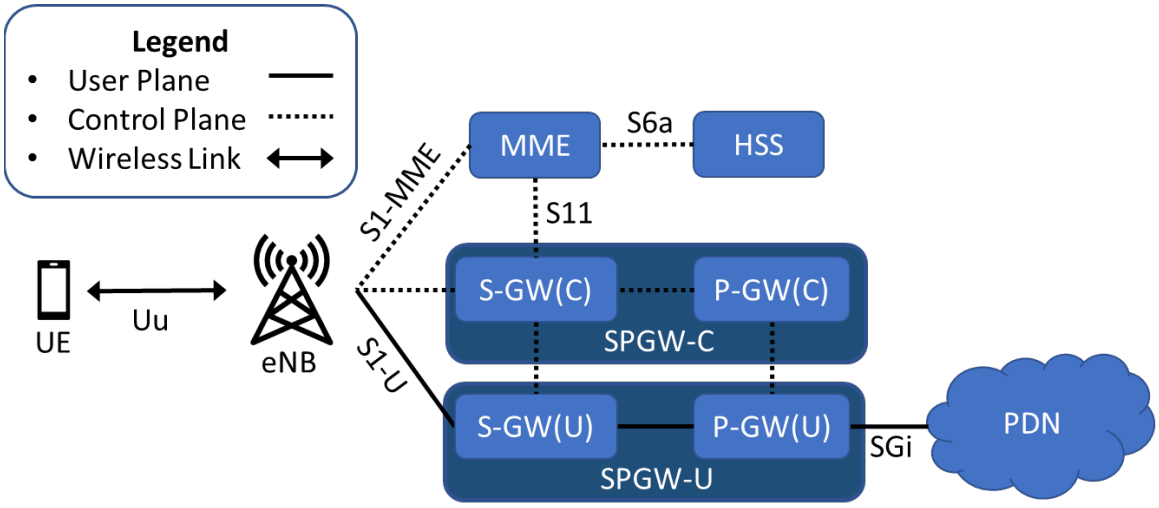

Figure 20. EPC Architecture in OAI 4G CN Software

After installing the Docker software, base images are pulled from the Docker repository for Ubuntu 18.04 and the Cassandra database management system. The Ubuntu images are used as the basis for creating the EPC network function images and the

Cassandra image is used to create a database that stores subscriber information. The Cassandra container operates exclusively with the HSS container. Additionally, the CN computer is configured to enable IP forwarding and the Docker network bridge is assigned an IP address on the desired network. The Docker network bridge is used to allow containers on one host machine to interact with other devices on the network.

The remaining steps required to configure the CN are covered in this section and explicitly detailed in Appendix E. First, additional docker images are built for each EPC network function using software downloaded from the OAI GitHub repository at https://github.com/OPENAIRINTERFACE/openair-epc-fed.git. Table 9 provides the software versions used for each of the OAI EPC network-function images. The SPGW-U image is referred to as SPGW-U-tiny to reflect the fact that a limited number of users can be supported. No other size SPGW-U image is offered on the public-facing OAI websites.

Table 9. OAI EPC Network Function Software Branch and Tags

Network Function	**Branch**	**Tag**
HSS	Master	V1.1.1
MME	Developer	2020.w47
SPGW-C	Master	V1.1.0
SPGW-U-tiny	Master	V1.1.0

Next, Python scripts are executed to generate configuration files for each of the containers. The configuration files are used with the images to create each of the four network function containers and the Cassandra database container. The containers are then connected to a private network so that the network functions can communicate with one another. During the configuration process, the MME, SPGW-C, and SPGW-U-tiny functions are made reachable through the Docker network bridge for devices external to the CN host machine. Also, the configuration process creates subscriber information for a specified number of users. Finally, the network functions can be started and stopped as desired to test the EPC. When the containers are started or secured, a specific order must be followed: HSS, MME, SPGW-C, and SPGW-U-tiny.

Scripts were created to facilitate each of the steps required to configure the EPC so that changes to network parameters could be made without manually reworking the software installation and configuration process. These scripts were divided between each of the steps described (i.e., image creation, container configuration, container launching, and container stopping) and an additional script to facilitate restarting each container after the host machine was rebooted. These scripts form each of the sections in Appendix E and are provided with the electronic source material for this work.

A final note about the OAI EPC software is that there are multiple ways to deploy the Docker containers. Instead of manually launching and stopping the containers after configuration as described, a separate process can be followed to configure and permanently launch the containers. This process is more appropriate for a permanently deployed EPC instead of one used for research. When the permanent deployment was used, the resulting EPC log files overflowed the available memory on the host machine after 72 hours of continuous operation. A method of maintaining and storing large volumes of EPC logs generated by the software is not provided. Additionally, a recent developer branch has shifted towards integration of the OAI EPC software with Magma MME software from Facebook Connectivity. This branch utilizes the "docker-compose" set of commands to replace the scripts from the described approach.

C. OPEN STACK NETWORK CONFIGURATION

After configuring the open stack network software, additional steps are still required to connect a COTS UE to the network. These include programming the USIM application with correctly provisioned data, testing the network interfaces, and obtaining access to the EC20 modem commands.

1. Subscriber Information Provisioning

For simplicity, the information for a subscriber was copied from the HSS configuration file to provision the USIM instead of generating a completely new subscriber. Three applications were evaluated to read and write to the UICC: pySIM, SIM manager from Dekart, and GRSIMwrite.exe from OYEI TIMES in Sehnzhen, China.

pySIM is an open-source project for reading and writing subscriber information. This project uses the Python programming language and was excellent for reading data from the UICC. Programming subscriber information onto the UICC, however, was limited to basic parameters such as the IMSI, MSISDN, ICCID, and cryptographic keys. While additional functionality could be programmed into pySIM, this is left as an item for future work.

Dekart SIM manager was trial software best suited for reading the UICC. It displayed the results of the read command in a detailed tree structure with each the hexadecimal offset for each file. The software was unable to program the blank UICCs available, which might be a limitation of the trial period.

OYEI TIME GRSIMwrite.exe was the most capable application evaluated for writing information to the UICC. The software displays the answer to reset (ATR) code used to identify the UICC and its capabilities. Within the application, separate tabs exist for most cellular technologies. In addition to the basic subscriber information, various types of PLMN (e.g., home, operator, forbidden), the number of the SMS provider, and the service provider name can be programmed. The only drawback to the software is that it does not come with any support or user guide for the other available features.

2. Network Diagram and Software Commands

Multiple partial network set-ups were utilized to test each aspect of the open stack network before attempting a full network set-up. These set-ups are described here to provide a variety of methods to troubleshoot the open stack network. Section V.A discusses how each set-up was used to achieve UE operation over the complete open stack network. The soft UE refers to the use of the OAI UE application running on an additional computer.

1. **eNB and Soft UE over Ethernet.** This operation mode ensures that the OAI RAN software has been properly installed by transmitting LTE packets over an Ethernet interface without using any RF devices. No EPC is required for this set-up.

2. **eNB and Soft UE over RF.** This operation mode tests the wireless interface between the eNB and UE without the EPC. To avoid the need for a Faraday cage, this set-up was used with SMA cables. The transmitter of one SDR was connected to the receiver of the other SDR with 30 dB of in-line attenuation to prevent inadvertent damage to the radio interfaces.

3. **eNB, EPC, and Soft UE over Ethernet**. This operation mode tests the network interfaces between the eNB and EPC as well as checking that the soft UE and EPC have matching subscriber information. All communications are passed over Ethernet.

4. **eNB, EPC, and Soft UE with Wireless Link**. This mode tests the network interfaces between all devices and includes the wireless link between the UE and eNB. In this set-up, the entire open stack network is tested using the soft UE instead of a commercial UE.

5. **eNB, EPC, and COTS UE**. This mode tests the entire open stack network with a fully functional UE. For UEs without displays, a computer must be used to pass commands to the modem and attach it to the network. This process is discussed in the following section.

3. EC20 Modem Commands

The EC20 modem can be controlled by an externally connected computer through two methods. The first method involves using the QNavigator software provided by Quectel. The second is to use the DiagParser and minicom utilities to remotely run commands on the modem. While QNavigator includes a library of modem commands and their usage, the second method can also be used to obtain root access to the modem OS. Each method was used to explore the EC20 modem. The process required to execute each of these methods are presented in the remainder of this section. Modem commands are also called AT commands, where “AT” is meant to grab the attention of the modem.

a. QNavigator

Version 1.6.9.1 of the QNavigator software is the most recent update to the modem interfacing utility from Quectel. This version includes support for 4G LTE devices, which was necessary for testing the modem in the 4G LTE open stack implementation. As a graphical, Windows-based program, minimal configuration was required. The program and Quectel modem drivers were installed using default parameters. Then, the EC20 was connected to the computer via USB interface on the developer board. After opening the software, selecting LTE and following the on-screen prompts, the program automatically attempts to register the modem to an available network.

The left-hand side of Figure 21 contains tabs used for common tasks such as sending text messages, initiating voice calls, and managing TCP/UDP connections. The center pane displays information for the currently selected task. In this case, the home tab displays information about the modem, network registration, and subscriber. Other tabs contain macros for their designated tasks to minimize the need for memorizing AT commands. The right-hand pane displays a log of the previously executed AT commands. Single commands can be sent using the text input box below the log.

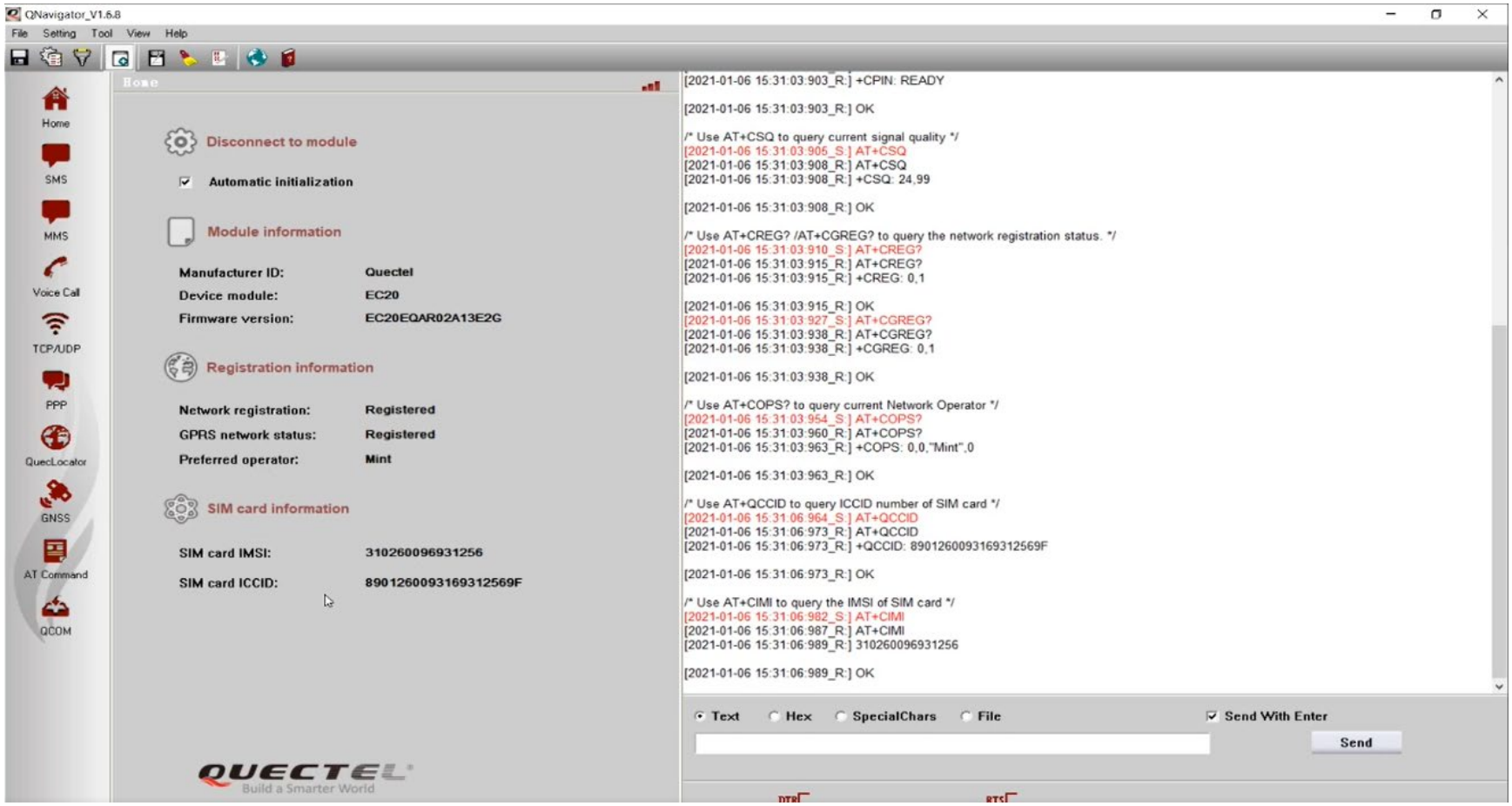

Figure 21. QNavigator Screenshot

b. DiagParser and Minicom Utilities

The EC20 can also be controlled through Linux command line utilities diag_parser and minicom. DiagParser decodes the byte-level messages output by the modem in the Qualcomm DIAG format. Three terminal windows are required: one to read the DIAG code, one to decode the DIAG message format, and one to send commands to the modem. Appendix F contains detailed steps for this method, including additional steps to capture mobile telephony packets in Wireshark and access the modem Linux OS. The process for executing modem-level commands is provided in the remainder of this section.

1. Open a terminal window and launch the minicom utility with "sudo minicom -s." Within the utility, select "setup" and change the path to "/dev/ttyUSB0." Escape and exit the menu. The raw byte-level messages from the modem will now display on the screen.
2. In a second terminal window, change directory to the DiagParser utility with "cd ~/diag_parser." Run the utility with "sudo ./diag_parser -g 127.0.0.1 -i /dev/ttyUSB0 -v" to decode the packets received by the modem.
3. In the third terminal window, run "sudo minicom -s." Select "setup" and change the device path to "/dev/ttyUSB2." Escape and exit the menu. The modem will attempt to execute its initialization AT commands. To make the user input visible, type "ATE1" and hit the return key.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. SPOOFED NETWORK ATTACK THEORY

After successful implementation of the open stack 4G LTE network, a few findings stood out from an attack perspective. First, the user programs all the parameters used for network identification without any external forcing function to prevent copying parameters specific to the MNO. Second, the UE and network mutually authenticate via a shared set of parameters, most of which are available in the USIM application. There is no verification of either entity through a trusted third party. Finally, most subscriber parameters can be read using modem-level commands without requiring privilege escalation. Combined, these weaknesses form the basis for elevating the legacy false base station attack into a spoofed network attack.

A. REQUIRED NETWORK PARAMETERS

When configuring the network core containers, the user programs network parameters such as the MCC, MNC, TAC, APN, and the Operator Variant Algorithm Configuration Field (OP). Except for OP, these network parameters are either broadcast by cell towers or easily found on UEs attached to the commercial network. Figure 22 is a screenshot from the *LTE Discovery* application depicting the MCC, MNC, and TAC for a cellular tower near the NPS campus. The APN and OP are configured in the HSS container, while the MCC, MNC, and TAC are configured in the MME container. In the OAI EPC software, the APN is required to configure the hybrid SPGW-C container that combines the control plane functionality of the SGW and the PGW.

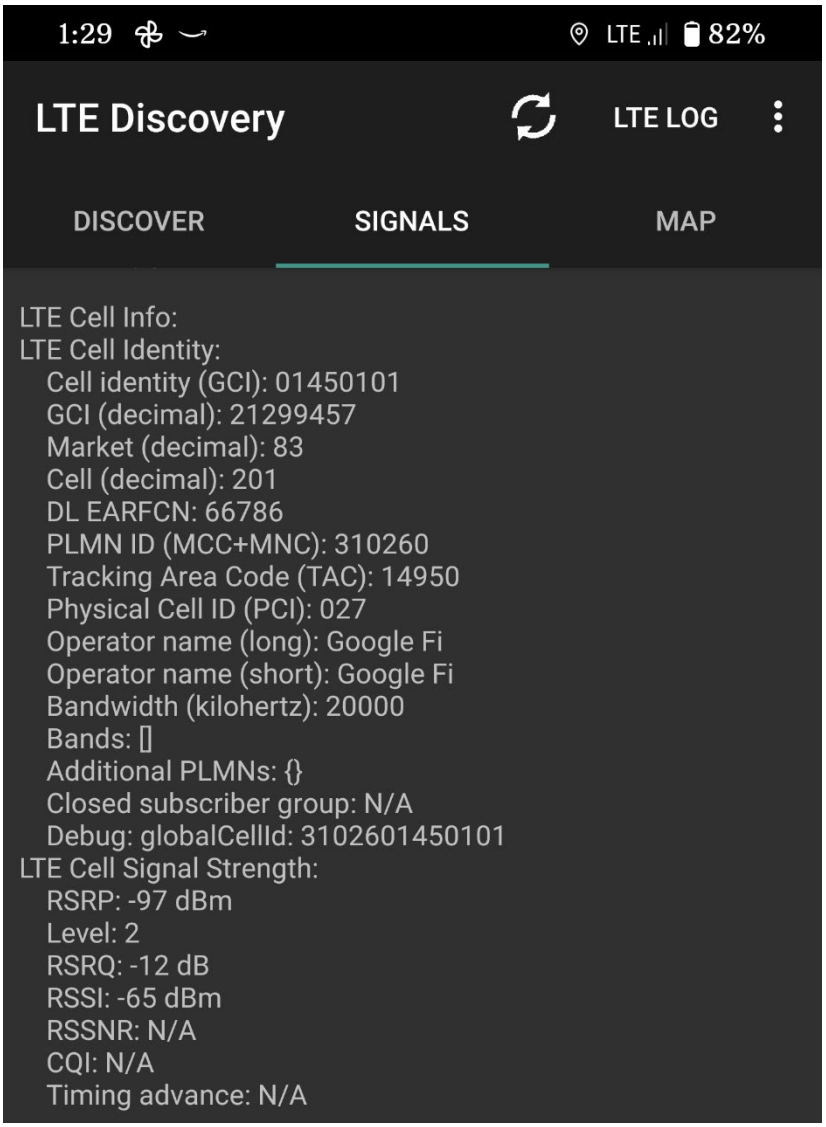

Figure 22. LTE Discovery Screenshot of an LTE Cell Parameters

OP is a 128-bit key, specific to the MNO or MVNO, used to provide an additional layer of protection to the subscriber's secret key, K_i. The OP_C is derived from OP using (4) where the encryption algorithm, E, could be any 128-bit keyed function that uses a 128-bit key [36]. 3GPP analysis and commercial practice use the Rijndael algorithm specified in the 128-bit Advanced Encryption Standard (AES) with the encryption key, K, being the subscriber secret key, K_i. 3GPP recommendations in §8.3 of [36] are to store OP and calculate OP_C off the USIM to prevent discovery and inhibit reverse engineering of the OP.

$$OP_C = OP \oplus E(OP)_K \tag{4}$$

Although many of the network parameters are known, the protection against spoofed networks is threefold: subscriber information must be known for each target; certain MNO-specific secret parameters must also be known; and large-scale network deployment is unlikely to remain covert. The spoofed network attack is best suited for a small number of target devices with modem-level vulnerabilities that operate within a suitable cell size.

B. THEORETICAL STEPS

The following keys steps form a theoretical exploitation chain that utilizes the component-level vulnerabilities in Qualcomm GNSS chipsets to force a target device to attach to a spoofed network.

1. The attacker uses vulnerabilities in the GNSS chipset to gain access to the cellular modem.
2. The attacker then uses AT commands to formulate legitimate queries for subscriber information on the phone such as IMSI, keys, or cryptography capabilities and stores the results in a temporary file on the embedded Linux operating system of the modem.
3. At a desired time, the target UE transmits the collected information to another UE owned by the attacker via SMS messages created with AT commands. These messages are transparent to the user-application level.
4. Subscriber information is post-processed to recover K_i from CK and IK.
5. The attacker adds the valid user credentials of the target to the subscriber information database: the HSS or UDM.
6. When desired, the attacker utilizes the exploit from step 1 to regain access to the UE and manually attach the device to the spoofed network. Alternatively, the list of preferred PLMNs can be updated during initial

access so that the target UE automatically reconnects to the spoofed network whenever it is available.

7. The target network traffic is now visible to the attacker. As the owner and maintainer of the spoofed network, the attacker can filter or alter any traffic to or from the UE.

This exploitation chain, while theoretical, includes several steps that were reproduced using commercial hardware and the open stack network. These steps will be further discussed in the Results chapter of this thesis.

C. GNSS COMPONENT VULNERABILITIES

In [1], multiple vulnerabilities are listed which, when combined, could allow for execution of arbitrary code in Google Android operating systems (OS). Some of the vulnerabilities are specific to the OS, but the majority are related to Qualcomm components. An additional vulnerability is posted in CVE-2018-13911 in which the GNSS XTRA parser can be forced into reading and accessing out of bounds memory locations [4].

The prevalence of cellular modems that come bundled with GNSS functionality by semiconductor manufacturers makes GNSS vulnerabilities a tempting starting point for attackers. Furthermore, the numerous CVEs regarding Qualcomm components cover a variety of product lines. Since the cellular modem typically controls the GNSS functions, exploits that target the GNSS logic would not require additional permissions to execute commands on the modem. The complex manipulation of GNSS vulnerabilities required to execute arbitrary AT commands is not further explored in this thesis (but is an item of future work), but a comparison can be made to the SIMJacker attack [37].

Like the SIMJacker, the proposed GNSS exploitation chain seeks to extract subscriber information from the USIM application through manipulation of modem functionality. Both SIMJacker and the proposed GNSS exploitation chain require significant computational resources to uncover the key K_i. While the SIMJacker attack can be used to brute force 3DES encryption using rainbow tables, the proposed exploitation

only requires backtracking a single step from the Cipher Key (CK) and Integrity Key (IK) to recover K_i. New algorithms for recovering K_i could be devised to reduce the computational time by taking advantage of additional module information exfiltrated during initial access. Another difference is that SIMJacker utilizes specially crafted Over The Air (OTA) binary messages to target vulnerable S@T Browser technology on the target USIM, and the proposed exploitation chain would use vulnerabilities in the GNSS chipset to execute legitimate modem-level AT commands.

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESULTS

A. OPEN STACK NETWORK RESULTS

This chapter discusses each of the steps used to test the open stack network implementation with applicable lessons learned. The following subsections are organized to reflect each network configuration utilized in the testing process. While the process presented follows the logical network progression presented in Section III.C.2, the actual experimentation progression was not as straightforward due to the disjointed and incomplete documentation available on the OAI wiki pages. A key contribution of this thesis is a consolidation of logical and repeatable steps to implement an OAI 4G LTE open stack network. Appendix G contains a guide to each of the archived network tests and contains the logs, configuration files, and packet captures of a desired test in the supplemental materials.

1. Network Configuration 1: eNB and Soft UE over Ethernet

After installing and configuring the OAI RAN software on two computers, the machines were connected via Ethernet cable to test the software without the RF interface. The machine running the eNB application was configured with an IP address of 192.168.13.1 and the machine running the UE application was configured with an IP address of 192.168.22.1. The IP address of the EPC computer, 192.168.14.1, was used as the gateway address on all computers because it was configured to forward local traffic through the NPS enterprise network. Figure 23 depicts the network addressing scheme used throughout the open stack network implementation.

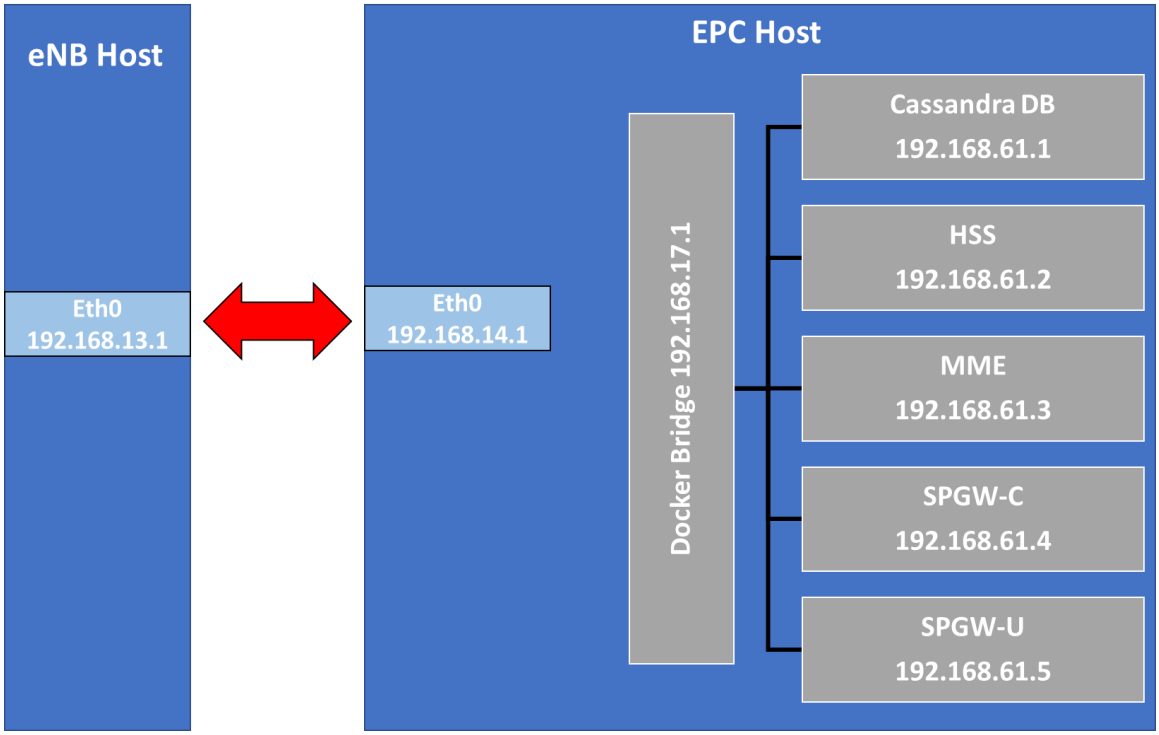

Figure 23. 4G LTE Open Stack Network IP Address Assignment

On the eNB machine, a terminal window was navigated to the "~/openairinterface5g/cmake_targets/ran_build/build" directory. Then, the eNB application was executed using a default configuration file provided with the RAN software as shown:

> ENODEB=1 sudo -E ./lte-softmodem -O ~/openairinterface5g/ci-scripts/conf_files/lte-fdd-basic-sim.conf --basicsim --noS1 --nokrnmod 1 | tee eNB.log"

The first argument, "ENODEB=1" is required in the basic simulator mode to tell the software to run in server mode and listen to all IP address on port 4043. The "-E" flag is used to preserve the OAI environmental variables while executing the eNB software. The "-O" flag is used to indicate the configuration file to be used with the eNB software. The "--basicsim" flag is used to indicate that all traffic will be passed over Ethernet without the RF interface. The "--nokrnmod 1" flag can be used when the "--noS1" flag is set to force the usage of the tunnel interface instead of using the nasmesh kernel module. Current tags of the RAN software utilize the tunnel interface by default, which makes the flag unnecessary. Finally, the "| tee eNB.log" is used to create a log file of the information the eNB software prints to the screen.

On the UE machine, a terminal window was navigated to the "~/openairinterface5g/cmake_targets/ran_build/build" directory. Then, the soft UE application was executed and configured to match the eNB parameters.

TCPBRIDGE=192.168.13.1 sudo -E ./lte-uesoftmodem -C 2680000000 -r 25 --ue-rxgain 140 --basicsim --noS1 | tee UE.log"

where the "TCPBRIDGE=192.168.13.1" specifies the IP address of the eNB machine. The "-C" flag is used to specify the carrier frequency of the eNB downlink signal in Hz and the "-r" flag specifies the number of radio bearers on the downlink. The "--ux-rxgain" flag is used to specify the gain used by the UE receiver. All flags are used for the same purpose as in the eNB software.

The UE machine connected to the eNB machine and created two additional network interfaces on each machine as shown in Table 10. Successful connectivity was verified by using the ping utility. The UE machine was pinged from the eNB machine with "ping -I oaitun_enb1 -c 20 10.0.1.2." Similarly, the eNB machine was pinged from the UE machine with "ping -I oaitun_ue1 -c 20 10.0.1.1." In each case, 100% of packets were received with round-trip-times (RTT) shown in Table 11.

Table 10. OAI Interfaces Created by Basic Simulator

Machine	**Interface Name**	**IP Address**
eNB	oaitun_enb1	10.0.1.1
eNB	oaitun_enm1	10.0.2.1
UE	oaitun_ue1	10.0.1.2
UE	oaitun_uem1	10.0.2.2

Table 11. RTT Comparison Between Ethernet and Network Configuration 1

	RTT		
Connection Type	**Minimum**	**Average**	**Maximum**
Network Configuration 1	6.994 ms	9.179 ms	13.692 ms
Direct Ethernet	0.154 ms	0.162 ms	0.177 ms

2. Network Configuration 2: eNB and Soft UE with RF Interface

Prior to testing in network configuration two, the operation of each USRP B200 was verified through a simple FM radio receiver application in GnuRadio. On one of the computers, the USRP was not recognized. The solution was to create a file at "/etc/udev/rules.d/10-usrp.rules," as discussed in Appendix H, and then reboot the computer. To test each USRP transmitter, a cosine wave was generated at 2.68 GHz and 0 dB gain in GnuRadio and coupled via SMA cable into an Agilent MXA Signal Analyzer. The signal was received at -75 dBm which showed that the USRP transmitters were functional and provided a coarse device power calibration.

The Ethernet cable was removed from the two computers. A USRP B200 was connected to each computer via USB3 port. The transmitter of each USRP was connected to the receiver of the other USRP with an additional 30 dB in-line attenuator to protect the SDR internals from high-powered signals. This setup is shown in Figure 24 with the UE computer on the left and the eNB computer on the right.

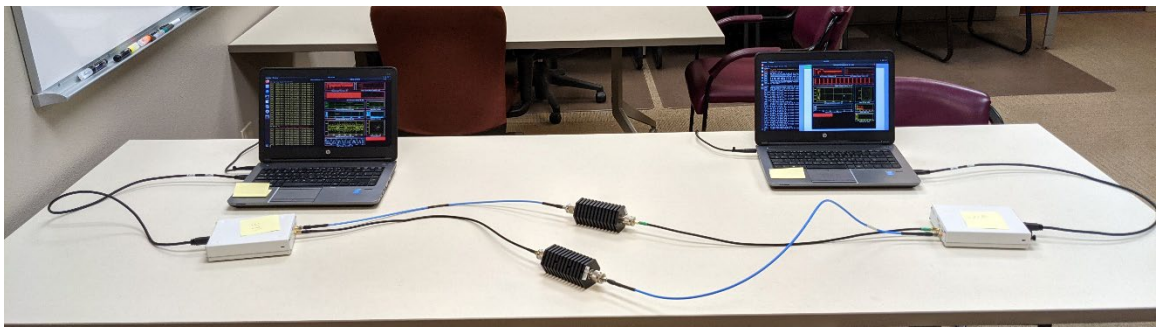

Figure 24. Implemented 4G LTE Open Stack Network in Configuration 2

On the eNB machine, a terminal window was navigated to the "~/openairinterface5g/cmake_targets/ran_build/build" directory. Then, the eNB application was executed using a configuration file provided with the RAN software for operation in LTE band 7 as shown:

```
sudo -E ./lte_build_oai/build/lte-softmodem -O ~/openairinterface5g/ci-
scripts/conf_files/my-enb.band7.tm1.25PRB.usrpb210.conf --nokrnmod 1
--noS1 --eNBs.[0].rrc_inactivity_threshold 30 | tee ENB.log
```

All arguments retain their functionality as described in network configuration 1. The configuration file used was provided with the RAN software and calibrated for operation of the B200 series USRP in LTE band 7 with 30 dB attenuators. For this test, no modifications to the configuration file were required. The "--eNBs" flag is used to alter parameters defined in the configuration file. In this command, the radio resource control (RRC) inactivity threshold was set to 30 seconds.

On the UE machine, a terminal window was navigated to the "~/openairinterface5g/cmake_targets/ran_build/build" directory. Then, the UE application was executed with the following command:

```
sudo ./lte-uesoftmodem -C 2680000000 -r 25 --ue-rxgain 120 --ue-txgain
0 --ue-max-power 0 --ue-scan-carrier --nokrnmod 1 --noS1 | tee UE.log
```

All arguments retain their functionality as previously described. Additional flags were utilized to set initial values for the transmitter gain, the maximum allowable transmitter power in dBm, and to ensure that the UE scanned for carrier frequencies offset from the nominal value specified.

Initially, the master branch of the RAN software was utilized for this test. Although the UE and eNB would attach, numerous problems arose including segmentation faults, buffer underflows, and buffer overflows. Troubleshooting efforts focused on altering several parameters in the eNB configuration file, including transmitter and receiver gain and different combinations of threading, but nothing solved the problems. Confident that the problems were not the result of errors in the eNB configuration, the RAN software was rebuilt using the developer branch tag 2020.w47 as described in Section III.B.5. This network configuration was then retested. With the developer branch of the software, none of the issues found with the master branch were observed over multiple hour-long tests.

The network connection was tested by pinging each interface as was done with network configuration 1. RTTs for a direct Ethernet connection, network configuration 1 and 2 are presented in Table 12. Network configuration 2 exhibited the longest RTT of all

three connections, including a factor of three increase over the RTT of network configuration 1. The most likely cause of this additional latency is the processing time required for the SDR to reconstruct the packets from received raw samples.

Table 12. Network Configuration 2 iPerf Testing Results

	RTT		
Connection Type	**Minimum**	**Average**	**Maximum**
Direct Ethernet	0.154 ms	0.162 ms	0.177 ms
Network Configuration 1	6.994 ms	9.179 ms	13.692 ms
Network Configuration 2	20.411 ms	30.785 ms	37.063 ms

3. Network Configuration 3: EPC, eNB, and Soft UE over Ethernet

Similar to network configuration one, this step connected all three components together via Ethernet cables and network switch. The machine running the eNB application was configured with an IP address of 192.168.13.1. The machine running the UE application was configured with an IP address of 192.168.22.1. The machine running the EPC application was configured with an IP address of 192.168.14.1. All computers used 192.168.14.1 as the gateway address. The network assignments for the eNB and EPC are depicted in Figure 23.

The EPC computer was also connected to the NPS campus WiFi, with the connection shareable to the other computers on the private subnetwork. Since the EPC software runs on a private subnetwork behind the Docker network bridge, a route was added on the eNB computer with the following command:

```
sudo ip route add 192.168.61.0/26 via 192.168.17.1 dev eth0
```

Prior to starting the test, connectivity between the eNB and EPC network functions was verified by pinging the MME, SPGW-C, and SPGW-U-tiny from the eNB. The Cassandra database and HSS containers were configured so that they are not accessible to networked devices outside the 192.168.61.0/26 subnetwork.

First, the EPC network functions were launched using the "Launch_4G_Containers.sh" script. After a few seconds, the MME log displays a line indicating that a listening socket has been created on port 36412 with the IP address associated with the MME container. Once this message has been displayed, the eNB application was run with:

> ENODEB=1 sudo -E ./lte-softmodem -O ~/openairinterface5g/ci-scripts/conf_files/net3_lte-fdd-basic-sim.conf --basicsim --nokrnmod 1 | tee eNB.log"

where the configuration file was an edited copy of the "lte-fdd-basic-sim.conf" file. In the configuration file, CI_MME_IP_ADDR was replaced with 192.168.61.3 and all instances of CI_ENB_IP_ADDR were replaced with 192.168.13.1. Successful connection of the eNB to the EPC was shown in the periodic MME updates printed to the terminal window. The number of connected eNBs increased from zero to one after displaying multiple lines of peering information as shown in Figure 25. Additionally, the eNB log displayed several lines of peering information that was similar what was contained in the MME log.

```
    ----------------------
    Peer addresses:
        - [192.168.13.1]
    ----------------------
    SCTP RETURNING!!
    Create eNB context for assoc_id: 2
    [2][48] Msg of length 59 received from port 36412, on stream 0, PPID 18
    SCTP RETURNING!!
    S1-Setup-Request macroENB_ID.size 3 (should be 20)
    New s1 setup request incoming from macro eNB id: 00e00[0m
Adding eNB to the list of served eNBs
    Adding eNB id 3584 to the list of served eNBs
    [48][2] Sending buffer 0x7ff258009f90 of 27 bytes on stream 0 with ppid 18
    Successfully sent 27 bytes on stream 0
    ======================================== STATISTICS ============================================

                    |   Current Status| Added since last display|  Removed since last display |
    Connected eNBs  |          1      |              1             |             0               |
    Attached UEs    |          0      |              0             |             0               |
    Connected UEs   |          0      |              0             |             0               |
    Default Bearers |          0      |              0             |             0               |
    S1-U Bearers    |          0      |              0             |             0               |

    ======================================== STATISTICS ============================================
```

Figure 25. Screenshot of eNB Pairing with EPC in MME Log

Before running the UE software, a soft SIM was programmed using additional tools from the OAI RAN software. A copy of the file "ue_eurecom_test_sfr.conf" at "~/openairinterface5g/openair3/NAS/TOOLS/" was saved as "ue.conf" in the file path for the eNB and UE executables. In the "ue.conf" file, several parameters were updated to reflect the values used in the EPC configuration as shown in Table 13.

Table 13. Network and Subscriber Parameters for Soft UE

Parameter	**Value**
MCC/MNC	404/92
Full Network Name	AirTel
Short Network Name	CMCC
MSIN	1000000081
K	0C0A34601D4F07677303652C0462535B
OPC	BA05688178E398BEDC100674071002CB

After editing the configuration file, a soft SIM was created for the soft UE with:

```
../../nas_sim_tools/build/conf2uedata -c ue.conf -o .
```

Finally, the UE software was executed with:

```
TCPBRIDGE=192.168.13.1 sudo -E ./lte-uesoftmodem -C 2680000000 -r
25 --ue-rxgain 140 --basicsim | tee UE.log"
```

When the UE completed connecting and attaching to the EPC, the statistics section of the MME log automatically updated itself to reflect that one UE was attached to the network. In a new terminal on the UE computer, the "ifconfig" utility was run and displayed two new interfaces. The first interface "oaitun_ue1" had an IP address of 12.1.12, which had been assigned by the EPC. The second interface "oaitun_uem1" had and IP address of 10.0.2.2 as seen for the previous two network configurations.

To test the network in this configuration, an additional terminal window on the UE computer was used to ping 8.8.8.8 via the "oaitun_ue1" interface. The first attempt was unsuccessful because network address translation was not enabled in the SPGW-U container. The container was rebuilt with an additional flag in the Python configuration file for "--network_ue_nat_option=yes." After restarting the EPC, the UE was able to ping 8.8.8.8 and other external webservers.

4. Network Configuration 4: EPC, eNB, and Soft UE with RF Interface

In this configuration, which is similar to that of network configuration three, the Ethernet connection to the UE computer was removed and replaced with the RF link as in configuration two. All network parameters and IP addresses remained the same from network configuration three. First, the EPC network functions were launched using the "Launch_4G_Containers.sh" script. Once the MME log indicated it was ready for peering, the eNB application was run with:

```
sudo -E ./lte-softmodem -~/openairinterface5g/ci-scripts/conf_files/my-
enb.band7.tm1.25PRB.usrpb210.conf | tee eNB.log"
```

where the configuration file utilized the base file for the USRP B200 series devices in LTE band 7 and replaced the default MME and eNB IP addresses with 192.168.61.3 and 192.168.13.1, respectively. The RRC inactivity threshold was set at 30 seconds in the configuration file and was sufficient for this test.

After the eNB completed peering with the EPC, the UE software was run with:

```
sudo ./lte-uesoftmodem -C 2680000000 -r 25 --ue-rxgain 120 --ue-txgain 0 --
ue-max-power 0 --ue-scan-carrier | tee UE.log
```

All arguments retain their functionality as previously described. Successful connection and attachment of the UE to the network was observed in the MME log. The UE then successfully pinged 8.8.8.8 and other external web servers.

To better understand the flow of packets in the open stack network, a reverse ping command was made to the IP address of the eNB computer. The reverse ping showed the packet transfer from the UE to the SPGW-U-tiny container to the EPC host network interface and then to the eNB computer. This confirmed that traffic between the UE and EPC is passed through tunnel interfaces. Additionally, the reverse ping test reinforced the separation between the EPC container subnetwork and the EPC host networking utilities.

5. Network Configuration 5: EPC, eNB, and COTS UE

In this network configuration, the soft UE from network configuration 4 was replaced with the EC20 UE. The "MAIN" antenna port was connected to the receiver port of the USRP

B200 and the “DIV” antenna port was connected to the transmitter port of the B200. Each of these connections included 30 dB of inline attenuation. An omnidirectional antenna was connected to the “GNSS” port on the EC20. This setup is displayed in Figure 26 with the laptop used to issue commands to the EC20 modem on the left and the eNB laptop and SDR on the right. The third laptop running the EPC software is not visible.

Figure 26. Implemented 4G LTE Open Stack Network in Configuration 5

Before the EC20 modem could be connected to the network, the subscriber information had to be programmed onto a UICC. First, a blank UICC was programmed with the pySIM application. In addition to the subscriber information in Table 13 for the soft SIM, an ICCID of 8991920123456789000 was provisioned. The format of the ICCID is 89 for mobile telephony, followed by the country calling code (91 for India), the MNC (92 for AirTel), and an account identification number (0123456789000). Once the UICC was programmed, it was verified with the pySIM reader and again with the Dekart SIM reader.

The EPC and eNB were launched as described for network configuration four, except with a different configuration file for the eNB. In this configuration file, the LTE band was

changed to 3, downlink frequency to 1.87 GHz, and uplink frequency to 1.775 GHz. These values were chosen to operate within the specifications for a set of RF multiplexers available in the lab. Ultimately, the multiplexers were not needed for this work, but showcased the simplicity of switching frequency bands at the eNB.

The EC20 was then powered on and communication was established with the UE computer. First, the command:

AT+COPS=?

was sent to query which mobile networks the EC20 could detect. While the UE was able to detect the open stack AirTel network, it would not connect to the network. Additionally, the UE was able to detect a commercial T-Mobile network at -110 dBm through poorly shielded segments of SMA cabling. Next, the command:

AT+QCFG="nwscanmode",3,1

was used to force the EC20 to only scan for LTE networks. Additionally, the command:

AT+QCFG="band",0,4,0,1

was used to restrict the EC20 to LTE band 3. After both commands were executed, the registration status of the modem was queried with:

AT+QNWINFO

No network information was displayed, which confirmed the lack of attached UEs in the MME log.

Since the entire network operation had been verified with the soft UE and soft SIM, the programmed UICC was identified as the problem. Further investigation into the UICC with the Dekart SIM reader revealed that the USIM application was not loaded onto the card. The UICC did have the SIM application loaded onto the card, which is where the subscriber information was programmed.

A new set of UICCs with the USIM application pre-loaded were then obtained. Additionally, the UICCs came bundled with access to the OYEI TIMES SIM writer software.

This software was used to program the subscriber information in the yellow boxes shown in Figure 27. An MSISDN was also programmed to facilitate testing voice calls over the network.

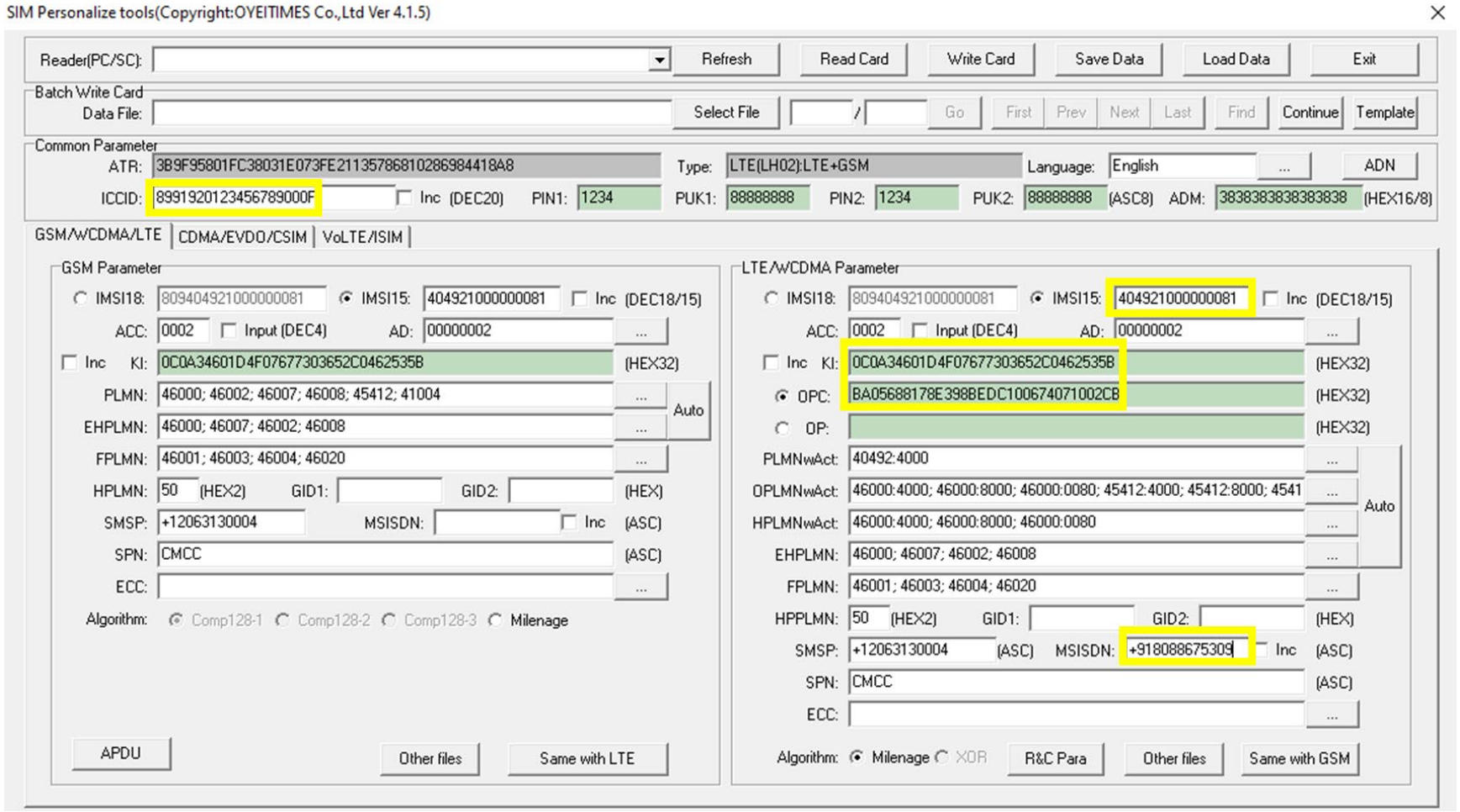

Figure 27. Subscriber Information Programmed via OYEI TIMES Software

The new UICC was inserted into the EC20 and access to the modem obtained on the UE computer. The modem was then connected to the network with the command:

AT+COPS=1,2,“40492”,7

Additionally, the address of the access point name (APN) was configured with the command:

AT+CGDCONT=1,“IP”,“apn1.carrier.com”,“12.1.1.0”

The MME log displayed completion of the authentication and key agreement process and attachment of a UE. On the EC20, a green LED turned on to indicate attachment to a network. Verification of the network was performed by executing the command:

AT+QPING=1,“8.8.8.8”,4,8

to ping the Google webserver. All eight pings were successful with RTT as shown in Figure 28 with the minicom utility. The QNavigator software was unable to identify the return ping messages which could be an issue with the Windows drivers for the EC20 modem.

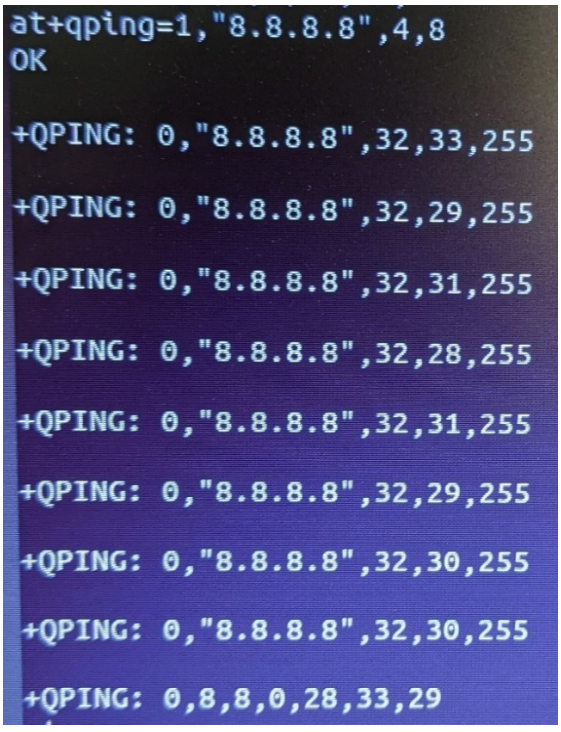

Figure 28. Successful Ping from EC20 to Internet Server

While the EC20 was connected to the open stack network, several AT commands were tested to determine what functionality was available through modem-level commands. First, the command:

AT+CLAC

was run to list all commands available to the modem as specified in §8.37 of [38]. The results, listed in Appendix I, included several commands not listed in the guides provided by Quectel or the QNavigator software. Next, the command:

AT+COPN

was run to list all the network operators stored in the UE as specified in §7.21 of [38]. The results of this command, listed in Appendix J, is not stored in the UICC. The location of this file within the EC20 OS was not determined.

Additionally, AT commands were attempted to make phone calls and send text messages. These network services failed because the open stack network was not configured to interoperate with other commercial networks, however, these services remain to be tested with a second COTS UE attached to the open stack network.

B. SIM CARD VULNERABILITY TESTING

A blacklisted UICC with subscriber credentials for a commercial mobile network (T-Mobile Straight Talk) was used as a comparison when programming UICCs. It was also used as a platform for testing methods to extract subscriber information, including information unavailable through modem-level commands.

The SIM Jacker tool was used to find vulnerable target application references (TARs) that could be susceptible to attack. TARs are part of a text message used to indicate the protocol or application to which the message should be delivered. Once identified, the vulnerable TARs were fuzzed to determine whether the TAR could be exploited.

A complete scan of all TARs was completed over five days of continuous testing. Of all the TARs analyzed, only five were identified as potentially vulnerable: 0x443231, 0x4F4241, 0XB00010, 0XB00011, and 0xB00012.

Each of the vulnerable TARs were then fuzzed to determine any undocumented sets of parameters that could be used to pass through spoofed over-the-air update (OTA) messages. Although 35 potentially nonstandard combinations of SMS parameters were identified by the fuzzer, none could be used with the SIM Jacker attack. The resistance of the UICC to the SIM Jacker attack suggests that manufactures have improved some of the security mechanisms on the UICC.

C. SPOOFED NETWORK ATTACK VECTOR TESTING

The spoofed network attack vector described in Chapter IV outlines seven steps to execute the attack. While a detailed description of the mechanism to obtain modem-level access through GNSS vulnerabilities is not provided in this thesis, at least one security research organization has discovered an exploitation chain through the UE OS [39]. The following subsections describe how to extract subscriber data from the UICC and the implications in terms of the mobile telephony key hierarchy.

1. Subscriber Data Harvesting

A commercial UICC and the programmed test UICC were probed with various cellular modem commands to verify that the desired subscriber information could be recovered. Specific commands such as AT+CIMI or AT+QCCID can be used to query single parameters such as the IMSI or ICCID, respectively. For 5G capable UEs, AT+CSUPI and AT+CNAI can be used to query the SUPI and network specific identifier, respectively [38]. However, not all subscriber information can be accessed in this manner. For subscriber information that does not have a dedicated command, the AT+CRSM command was used to perform a binary read of the data. Table 14 lists some required subscriber information, the type of record storage, the offset from the USIM master file location, and the length of the record (if known). These parameters, listed in [26], are all required to construct a valid AT+CRSM command. While the EPC was not programmed with the appropriate credentials to authenticate the commercial UICC, it did not impact the ability to recover subscriber information through modem-level AT commands.

Table 14. USIM Application File Identifiers

Parameter	Record Type	Offset (hex / decimal)	Record Length
IMSI	Transparent	6F07 / 28423	9B
Keys	Transparent	6F08 / 28424	33B
USIM Service Table	Transparent	6F38 / 28472	≥1B
MSISDN	Linear Fixed	6F40 / 28480	Variable
Location Information	Transparent	6F7E / 28542	11B

2. Master Key Recovery

Although backtracking of the K_i from the CK and IK values is difficult, these values provide a good starting point, as shown in Figure 29. A set of simplifying assumptions could significantly reduce the computational cost of brute forcing the K_i. First, MNOs likely use the same OP key across all of their UICCs or at least across a batch of UICCs. When the OP value is known, (4) can be rewritten as (5) where the only unknowns are the encryption algorithm and K value.

$$OP_C = OP \oplus E(OP)_K \tag{4}$$

$$OP \oplus OP_C = E(OP)_K \tag{5}$$

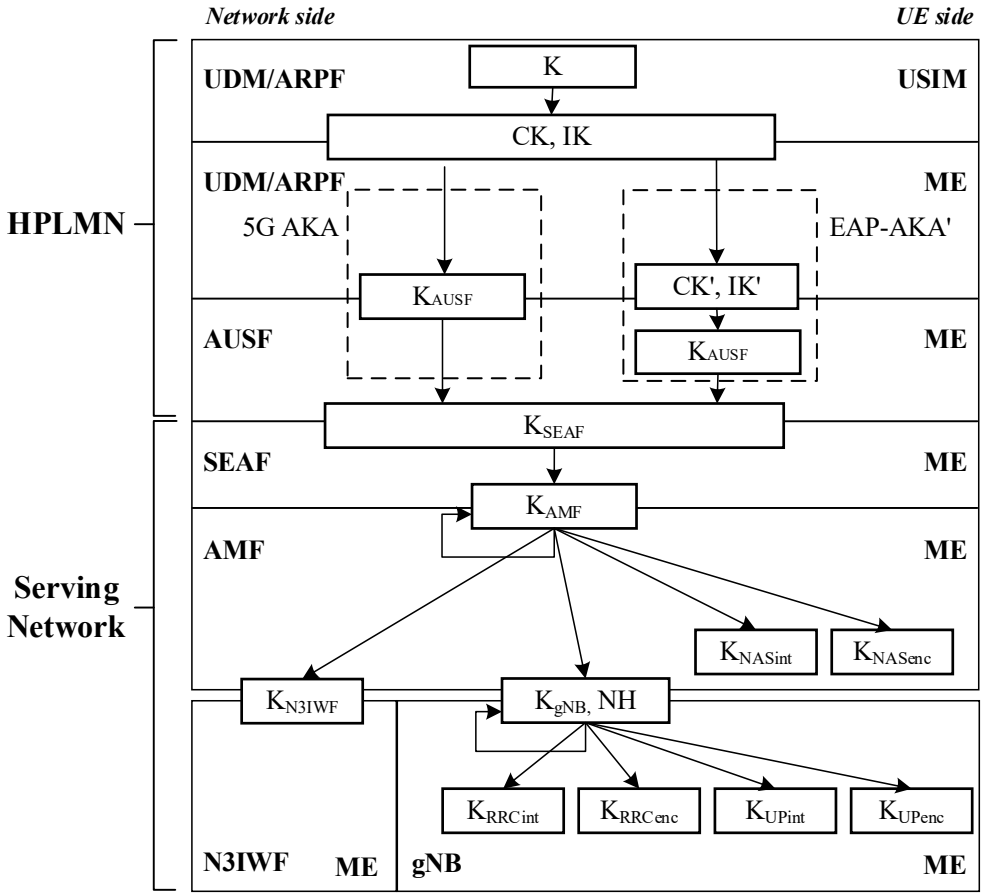

Figure 29. 5G Key Hierarchy. Source: [40].

The second assumption is that the encryption algorithms used by the MNO are common algorithms such as AES. This is because the UICC and CN must both utilize the same algorithm for AKA. Since the UICC should remain low-cost, it is unlikely to have specialized circuits for home-grown encryption algorithms.

With these two assumptions, the number of unknown variables is minimized. Since CK, IK, and OP are known, a cluster of computers should be able to brute force the K_i value. The amount of time required to obtain a solution could be reduced by configuring the spoofed network with a low RRC inactivity threshold to generate more sets of security vectors.

VI. CONCLUSIONS AND FUTURE WORK

A. CONCLUSIONS

This thesis provides a detailed guide to implementing an open stack 4G LTE network using the OAI software, an SDR, and a COTS UE. It also will assist others who would like to replicate the experimentation conducted as part of this thesis, or to use the open stack network to investigate other vulnerabilities and attack vectors. Instead of simply providing the commands to recreate a network with a specific set of parameters, this thesis also includes various network set-ups and utilities that can be used for troubleshooting, data collecting, or configuring a new network with different parameters. An archive of packet captures, configuration files, and logs is provided in the appendices and supplemental materials.

The open stack network is presented as a low-cost, small-footprint 4G/LTE network testbed that is extremely reconfigurable and scalable. The entire open stack network was implemented with approximately $3000 of equipment that could fit into a single Pelican 1510 case. As the 5G NR standards change to reflect new releases or even when 6G or 7G standards are developed, the same equipment can be reconfigured to create a modern cellular network. These qualities make using an open stack network ideal for integration into various backhaul infrastructures such as the Naval Operational Architecture. It is also beneficial for experimental and other purposes to have complete control over the security and operational aspects of the network, without heavy reliance on third-party support.

One of the most significant impediments to implementing the 4G LTE network was properly configuring subscriber information in the USIM application. As such, a major contribution of this work is providing updated guidance on SIM programming utilities and requirements for 4G/5G SIM cards. Certain applications, such as pySIM, are useful for reading the SIM or USIM application but have limited capabilities for writing data into the USIM application. On the other hand, the OYEITIMES SIM card programmer was capable of writing data into the USIM application but lacked some of the reader functionality seen in the pySIM program.

To better understand the mobile telephony security landscape, existing cyber-attack taxonomies were investigated and evaluated. After additional research into the 5G standards, a novel attack vector categorization scheme was developed and published in the IEEE Computer Society *Computer* magazine [5]. The framework is useful for analyzing attack vectors, rather than attack methods, and how they are enabled or defeated by mobile telephony security mechanisms. The authors believe that their novel categorization scheme, focusing on CUPS, is an effective method for reapplying legacy attack methods to modern mobile networks.

While implementing the open stack network, the concept of a spoofed-network attack vector was simultaneously developed and investigated. Such an attack vector is agnostic to the RAN and CN because it could be applied to any generation of mobile telephony standards. Key aspects of this attack vector were demonstrated, including the implementation of a 4G LTE network and the reading of critical subscriber information from the USIM application. Some limitations of this attack vector include the requirement that the target UE be within coverage range of the spoofed RAN and the limited function set provided in open-source network deployment. Additionally, the mechanism to recover the master key from the CK and IK stored on the USIM requires significant computational power and knowledge of the OP.

B. FUTURE WORK

The open stack network concept and spoofed network attack vector are presented only in a 4G LTE network due to 5G software unavailability. As the RAN and CN software finishes their testing stage, the open network deployment should be revisited for 5G NR networks. In addition to upgrading the utilities required for 5G open networks, several other items of future work are outlined.

One of the major hurdles encountered in the development of the 4G LTE open stack network was finding an existing capability to program the subscriber credentials in the USIM application. While multiple tools can be used to read the SIM or USIM application data, few were able to program the USIM. None of the SIM or USIM writer tools investigated can alter the service tables to configure 5G capabilities. Since the SIM and

USIM applications follow a rigid structure outlined in 3GPP technical documents, the open-source pySim tool could be upgraded with the additional functionality needed to generate 5G capable UICCs.

An additional goal for a new SIM card tool would be to automate the programming of the USIM application onto the UICC. Many of the UICCs available do not come with the USIM application installed which renders them useless for modern networks. By programming the USIM application on older UICCs, older resources can be repurposed. Furthermore, research into the process required to install the USIM application onto the UICC may reveal additional security vulnerabilities in the UICC environment.

Until the OSA software for 5G RAN and CN completes its testing stage, future work should focus on testing and implementing additional services to the 4G LTE open stack network. This would require attaching another COTS UE to attempt voice and data calls or messages within the network. The open stack network implemented in this thesis requires integration of a short message service center (SMSC) to support the short message service (SMS). Several open-source SMS gateways are available for this purpose, such as Jasmin SMS. With the aforementioned extension, the open stack network could be used in research on communications with unmanned vehicles. Such a demonstration would underscore the interdisciplinary research on unmanned vehicles at the Naval Postgraduate School and showcase the benefits to be accrued by integrating open stack networks into the Naval Operational Architecture.

Once the 5G network software has completed testing, the next task will be to implement an open stack 5G SA network. Although a 5G NSA open stack network could be implemented with the currently available software from OSA, doing so would require additional hardware modifications to support wired connections to both the eNB and gNB. The primary benefit of 5G NSA architectures is for commercial providers and not necessarily research institutions. A thorough investigation of utilities required to implement the 5G SA open stack network and detailed guidance is critical for other teachers and students eager to study modern mobile networks.

The spoofed network attack vector described in this thesis requires exploitation of GNSS chipsets. One avenue for future work is to continue investigating vulnerabilities in GNSS chipsets to determine how to implement the proposed attack vector. After the attack vector is thoroughly detailed, the method could be applied to various cellular modems to determine the scope of the attack vector against commercial hardware. While the spoofed network attack vector should work against both 4G and 5G networks, employing it against a vulnerable COTS UE on both types of networks may reveal practical differences in results.

Part of the benefit of the open stack network concept is that security researchers and academics can have low-cost access to an entire mobile network. Accessibility issues have previously precluded the CN from receiving the thorough security analysis afforded the RAN. With access to the entire network instead of just the RAN, future studies could investigate security vulnerabilities unique to the CN. Additionally, access to the RAN through weak security practices at the CN could be used to explore how legacy attack vectors could be applied to modern mobile networks.

Finally, 5G NR will push data processing closer to the network edge. Doing so effectively will require the utilization of artificial intelligence and machine learning (AI/ML) within the RAN. The open RAN (O-RAN) concept breaks down an eNB/gNB into three functional pieces to facilitate the introduction of AI/ML. Since the code for the OSA RAN is open source, it could be modified to include a layer for AI/ML. The RF and basic simulator could be used in conjunction with a cluster of low-cost computer nodes to test the response of the AI/ML algorithms to changing traffic and RF conditions.

APPENDIX A. GPS C/A CODE PHASE ASSIGNMENTS

This table, adapted from Table 5.3 of [41], provides the information necessary to identify or reproduce the C/A signal from each unique GPS satellite.

Satellite ID	GPS PRN Signal Number	Code Phase Selection	Code Delay Chips
1	1	$2 \oplus 6$	5
2	2	$3 \oplus 7$	6
3	3	$4 \oplus 8$	7
4	4	$5 \oplus 9$	8
5	5	$1 \oplus 9$	17
6	6	$2 \oplus 10$	18
7	7	$1 \oplus 8$	139
8	8	$2 \oplus 9$	140
9	9	$3 \oplus 10$	141
10	10	$2 \oplus 3$	251
11	11	$3 \oplus 4$	252
12	12	$5 \oplus 6$	254
13	13	$6 \oplus 7$	255
14	14	$7 \oplus 8$	256
15	15	$8 \oplus 9$	257
16	16	$9 \oplus 10$	258
17	17	$1 \oplus 4$	469
18	18	$2 \oplus 5$	470
19	19	$3 \oplus 6$	471
20	20	$4 \oplus 7$	472
21	21	$5 \oplus 8$	473
22	22	$6 \oplus 9$	474
23	23	$1 \oplus 3$	509
24	24	$4 \oplus 6$	512
25	25	$5 \oplus 7$	513
26	26	$6 \oplus 8$	514
27	27	$7 \oplus 9$	515
28	28	$8 \oplus 10$	516
29	29	$1 \oplus 6$	859
30	30	$2 \oplus 7$	860
31	31	$3 \oplus 8$	861
32	32	$4 \oplus 9$	862
Reserved	33	$5 \oplus 10$	863
Reserved	34	$4 \oplus 10$	950
Reserved	35	$1 \oplus 7$	947

Satellite ID	GPS PRN Signal Number	Code Phase Selection	Code Delay Chips
Reserved	36	$2 \oplus 8$	948
Reserved	37	$4 \oplus 10$	950

APPENDIX B. QUALCOMM COMPONENT-LEVEL VULNERABILITY CONSOLIDATED REPORTS

This table consolidates the CVE reports identified in [1] as affecting Qualcomm components using information provided from National Vulnerability Database hosted by the National Institute of Standards and Technology [42]. The two right-most columns indicate whether the vulnerability applies to the 9206 and 9607 product lines that are used in Quectel EC20 modems.

CVE #	Severity	Description	Affected Models	9206	9607
2019-2235	7.8	Buffer overflow occurs when emulated RPMB is used due to sector size assumptions in the TA rollback protection logic.	Snapdragon Auto, Snapdragon Compute, Snapdragon Connectivity, Snapdragon Consumer Electronics Connectivity, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wired Infrastructure and Networking in MDM9206, MDM9607, MDM9650, MDM9655, MSM8996AU, QCS404, QCS605, Qualcomm 215, SD 410/12, SD 425, SD 427, SD 430, SD 435, SD 439 / SD 429, SD 450, SD 625, SD 632, SD 636, SD 712 / SD 710 / SD 670, SD 820, SD 820A, SD 835, SD 845 / SD 850, SD 8CX, SDA660, SDM439, SDM630, SDM660, Snapdragon_High_Med_2016, SXR1130	x	x
2019-2236	5.5	Null pointer dereference during secure application termination using specific application identifiers.	Snapdragon Auto, Snapdragon Compute, Snapdragon Connectivity, Snapdragon Consumer Electronics Connectivity, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wired Infrastructure and Networking in IPQ8074, MDM9206, MDM9607, MDM9650, MDM9655, MSM8996AU, QCA8081, QCS605, Qualcomm 215, SD 410/12, SD 425, SD 427, SD 430, SD 435, SD 439 / SD 429, SD 450, SD 625, SD 632, SD 636, SD 650/52, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820, SD 820A, SD 835, SD 8CX, SDA660, SDM439, SDM630, SDM660, Snapdragon_High_Med_2016, SXR1130	x	x

CVE #	Severity	Description	Affected Models	9206	9607
2019-2237	5.5	Failure in taking appropriate action to handle the error case if keypad GPIO deactivation fails leads to silent failure scenario and subsequent logic gets executed every time.	Snapdragon Auto, Snapdragon Compute, Snapdragon Consumer Electronics Connectivity, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile in MDM9206, MDM9607, MDM9650, MDM9655, QCS605, SD 210/SD 212/SD 205, SD 410/12, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 8CX, SXR1130	x	x
2019-2238	7.8	Lack of check of data type can lead to subsequent loop-expression potentially go negative and the condition will still evaluate true, leading to buffer underflow.	Snapdragon Auto, Snapdragon Compute, Snapdragon Consumer Electronics Connectivity, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile in MDM9206, MDM9607, MDM9650, MDM9655, QCS605, SD 210/SD 212/SD 205, SD 410/12, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 8CX, SXR1130	x	x
2019-2239	5.5	Sanity checks are missing in layout which can lead to SUI corruption or can lead to Denial of Service.	Snapdragon Auto, Snapdragon Compute, Snapdragon Connectivity, Snapdragon Consumer Electronics Connectivity, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wired Infrastructure and Networking in MDM9150, MDM9206, MDM9607, MDM9635M, MDM9640, MDM9650, MDM9655, MSM8996AU, QCS404, QCS605, Qualcomm 215, SD 210/SD 212/SD 205, SD 410/12, SD 425, SD 427, SD 430, SD 435, SD 439 / SD 429, SD 450, SD 615/16/SD 415, SD 625, SD 632, SD 636, SD 650/52, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820, SD 820A, SD 835, SD 845 / SD 850, SD 8CX, SDA660, SDM439, SDM630, SDM660, SDX20, SDX24, Snapdragon_High_Med_2016, SXR1130	x	x

CVE #	Severity	Description	Affected Models	9206	9607
2019-2240	5.5	While sending the rendered surface content to the screen, error handling is not properly checked, resulting in an unpredictable behavior.	Snapdragon Auto, Snapdragon Compute, Snapdragon Connectivity, Snapdragon Consumer Electronics Connectivity, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon IoT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wired Infrastructure and Networking in IPQ4019, IPQ8064, IPQ8074, MDM9150, MDM9206, MDM9607, MDM9640, MDM9650, MSM8996AU, QCA6174A, QCA6564, QCA6574, QCA6574AU, QCA6584, QCA6584AU, QCA8081, QCA9377, QCA9379, QCA9531, QCA9880, QCA9886, QCA9980, QCN5502, QCS404, QCS605, SD 210/SD 212/SD 205, SD 425, SD 600, SD 625, SD 636, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820, SD 820A, SD 835, SD 845 / SD 850, SD 855, SD 8CX, SDA660, SDM630, SDM660, SDX20, SDX24, SXR1130	x	x
2019-2241	5.5	While rendering the layout background, error status check is not caught properly and also incorrect status handling is being done, leading to unintended SUI behavior.	Snapdragon Auto, Snapdragon Compute, Snapdragon Connectivity, Snapdragon Consumer Electronics Connectivity, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile, Snapdragon Wired Infrastructure and Networking in MDM9150, MDM9206, MDM9607, MDM9650, MDM9655, MSM8996AU, QCS404, QCS605, SD 210/SD 212/SD 205, SD 410/12, SD 636, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820, SD 820A, SD 835, SD 845 / SD 850, SD 855, SD 8CX, SDA660, SDM630, SDM660, SDX24, SXR1130	x	x

CVE #	Severity	Description	Affected Models	9206	9607
2019-2253	9.8	Buffer over-read can occur while parsing an ogg file with a corrupted comment block.	Snapdragon Auto, Snapdragon Connectivity, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon IoT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wearables in MDM9150, MDM9206, MDM9607, MDM9650, MSM8909W, MSM8996AU, QCS405, QCS605, Qualcomm 215, SD 210/SD 212/ SD 205, SD 425, SD 427, SD 430, SD 435, SD 439 / SD 429, SD 450, SD 600, SD 615/16/SD 415, SD 625, SD 632, SD 636, SD 665, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820, SD 820A, SD 835, SD 845 / SD 850, SD 855, SDA660, SDM439, SDM630, SDM660, SDX20	x	x
2019-2254	9.8	Position determination accuracy may be degraded due to wrongly decoded information.	Snapdragon Auto, Snapdragon Compute, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon IoT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wearables in MDM9150, MDM9206, MDM9607, MDM9615, MDM9625, MDM9635M, MDM9640, MDM9650, MDM9655, MSM8909W, MSM8996AU, QCS605, Qualcomm 215, SD 210/SD 212/SD 205, SD 425, SD 427, SD 430, SD 435, SD 439 / SD 429, SD 450, SD 615/16/SD 415, SD 625, SD 632, SD 636, SD 650/52, SD 665, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820, SD 820A, SD 835, SD 845 / SD 850, SD 855, SD 8CX, SDA660, SDM439, SDM630, SDM660, SDX20, Snapdragon_High_Med_2016, SXR1130	x	x
2019-2276	9.8	Possible out-of-bound read occurs while processing beaconing request due to lack of check on action frames received from user controlled space.	Snapdragon Auto, Snapdragon Consumer Electronics Connectivity, Snapdragon Consumer IOT, Snapdragon Mobile, Snapdragon Voice & Music in MDM9607, MSM8996AU, QCA6174A, QCA6574AU, QCA9377, QCA9379, QCS405, QCS605, SD 636, SD 665, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820A, SD 845 / SD 850, SD 855, SDM630, SDM660, SDX24		x

CVE #	Severity	Description	Affected Models	9206	9607
2019-2278	7.8	User keystore signature is ignored in boot and can lead to bypass boot image signature verification.	Snapdragon Auto, Snapdragon Consumer IOT, Snapdragon Mobile in MDM9607, MDM9640, SD 425, SD 427, SD 430, SD 435, SD 450, SD 625, SD 636, SD 712 / SD 710 / SD 670, SD 845 / SD 850, SDM660		x
2019-2305	9.8	Out-of-bound access when reason code is extracted from frame data without validating the frame length.	Snapdragon Auto, Snapdragon Consumer Electronics Connectivity, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile, Snapdragon Voice & Music in MDM9150, MDM9206, MDM9607, MDM9640, MDM9650, MSM8996AU, QCA6174A, QCA6574AU, QCA9377, QCA9379, QCS405, QCS605, SD 425, SD 427, SD 430, SD 435, SD 450, SD 625, SD 636, SD 665, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820, SD 820A, SD 835, SD 845 / SD 850, SD 855, SDA660, SDM630, SDM660, SDX20, SDX24	x	x
2019-2307	9.8	Possible integer underflow due to lack of validation before calculation of data length in 802.11 Rx management configuration.	Snapdragon Auto, Snapdragon Consumer Electronics Connectivity, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile, Snapdragon Voice & Music in MDM9150, MDM9206, MDM9607, MDM9640, MDM9650, MSM8996AU, QCA6174A, QCA6574AU, QCA9377, QCA9379, QCS405, QCS605, SD 210/SD 212/SD 205, SD 425, SD 427, SD 430, SD 435, SD 450, SD 600, SD 625, SD 636, SD 665, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820, SD 820A, SD 835, SD 845 / SD 850, SD 855, SDM630, SDM660, SDX20, SDX24	x	x
2019-2308	7.8	User application could potentially make RPC call to the fastrpc driver and the driver will allow the message to go through to the remote subsystem.	Snapdragon Auto, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wearables in MDM9150, MDM9607, MDM9650, MSM8909W, MSM8996AU, QCS405, QCS605, Qualcomm 215, SD 425, SD 427, SD 430, SD 435, SD 439 / SD 429, SD 450, SD 625, SD 632, SD 636, SD 665, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820A, SD 835, SD 845 / SD 850, SD 855, SDA660, SDM439, SDM630, SDM660, SDX20, SDX24		x

CVE #	Severity	Description	Affected Models	9206	9607
2019-2322	9.8	Buffer overflow can occur when playing specific clip which is non-standard.	Snapdragon Auto, Snapdragon Compute, Snapdragon Connectivity, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon IoT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wearables in MDM9150, MDM9206, MDM9607, MDM9650, MSM8909W, MSM8996AU, QCS405, QCS605, Qualcomm 215, SD 210/SD 212/SD 205, SD 425, SD 427, SD 430, SD 435, SD 439 / SD 429, SD 450, SD 600, SD 615/16/SD 415, SD 625, SD 632, SD 636, SD 665, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820, SD 820A, SD 835, SD 845 / SD 850, SD 855, SDA660, SDM439, SDM630, SDM660, SDX20, Snapdragon_High_Med_2016	x	x
2019-2326	7.8	Data token is received from ADSP and is used without validation as an index into the array leads to out-of-bound access.	Snapdragon Auto, Snapdragon Compute, Snapdragon Connectivity, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wearables in MDM9150, MDM9206, MDM9607, MDM9640, MDM9650, MSM8909W, MSM8996AU, QCS405, QCS605, Qualcomm 215, SD 210/SD 212/SD 205, SD 425, SD 427, SD 430, SD 435, SD 439 / SD 429, SD 450, SD 625, SD 632, SD 636, SD 665, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820, SD 820A, SD 835, SD 845 / SD 850, SD 855, SDA660, SDM439, SDM630, SDM660, SDX20, SDX24	x	x
2019-2327	9.8	Possible buffer overflow can occur when playing clip with incorrect element size.	Snapdragon Auto, Snapdragon Compute, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon IoT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wearables in MDM9150, MDM9206, MDM9607, MDM9650, MSM8909W, MSM8996AU, QCS405, QCS605, Qualcomm 215, SD 210/SD 212/SD 205, SD 425, SD 427, SD 430, SD 435, SD 439 / SD 429, SD 450, SD 600, SD 615/16/SD 415, SD 625, SD 632, SD 636, SD 665, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820, SD 820A, SD 835, SD 845 / SD 850, SD 855, SDA660, SDM439, SDM630, SDM660, SDX20, Snapdragon_High_Med_2016	x	x

CVE #	Severity	Description	Affected Models	9206	9607
2019-2328	7.8	Possible buffer overflow when number of channels passed is more than size of channel mapping array.	Snapdragon Auto, Snapdragon Compute, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon IoT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wearables in MDM9150, MDM9206, MDM9607, MDM9640, MDM9650, MSM8909W, MSM8996AU, QCS405, QCS605, Qualcomm 215, SD 210/SD 212/ SD 205, SD 425, SD 427, SD 430, SD 435, SD 439 / SD 429, SD 450, SD 600, SD 615/16/SD 415, SD 625, SD 632, SD 636, SD 665, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820, SD 820A, SD 835, SD 845 / SD 850, SD 855, SDA660, SDM439, SDM630, SDM660, SDX20, SDX24	x	x
2019-2330	5.5	Improper input validation in allocation request for secure allocations can lead to page fault.	Snapdragon Auto, Snapdragon Compute, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wearables, Snapdragon Wired Infrastructure and Networking in IPQ4019, IPQ8064, IPQ8074, MDM9150, MDM9640, MDM9650, MSM8909W, MSM8996AU, QCS405, QCS605, Qualcomm 215, SD 425, SD 427, SD 430, SD 435, SD 439 / SD 429, SD 450, SD 625, SD 632, SD 636, SD 665, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820A, SD 835, SD 845 / SD 850, SD 855, SDA660, SDM439, SDM630, SDM660, SDX20, SDX24		
2019-2334	7.5	Null pointer dereferencing can happen when playing the clip with wrong block group id.	Snapdragon Auto, Snapdragon Compute, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon IoT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wearables in MDM9150, MDM9206, MDM9607, MDM9650, MSM8909W, MSM8996AU, QCS405, QCS605, Qualcomm 215, SD 210/SD 212/SD 205, SD 425, SD 427, SD 430, SD 435, SD 439 / SD 429, SD 450, SD 600, SD 615/ 16/SD 415, SD 625, SD 632, SD 636, SD 665, SD 675, SD 712 / SD 710 / SD 670, SD 730, SD 820, SD 820A, SD 835, SD 845 / SD 850, SD 855, SDA660, SDM439, SDM630, SDM660, SDX20, Snapdragon_High_Med_2016	x	x

CVE #	Severity	Description	Affected Models	9206	9607
2019-2346	7.8	Firmware is getting into loop of overwriting memory when scan command is given from host because of improper validation.	Snapdragon Compute, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wired Infrastructure and Networking in IPQ8074, QCA8081, QCS404, QCS405, QCS605, SD 425, SD 427, SD 430, SD 435, SD 450, SD 625, SD 636, SD 712 / SD 710 / SD 670, SD 820, SD 835, SD 845 / SD 850, SD 855, SD 8CX, SDA660, SDM630, SDM660		

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. QUECTEL PRODUCT BROCHURES

The Quectel EC20 product line has been discontinued after the release of revision 2.1 modems. Table 15 concatenates relevant LTE parameters into a single table for comparison between the EC20 revision 1 and 2.1.

Table 15. EC20 LTE Parameters by Revision. Adapted from [27] and [29].

Parameter	**Revision 1**	**Revision 2.1**
LTE Bands	FDD: B1/B3/B5/B7/B8/B20	FDD: B1/B3/B5/B8 TDD: B34/B38/B39/B40/B41
LTE Version	3GPP E-UTRA Release 9	3GPP LTE Release 11
Max Data Rate Downlink	FDD: 100 Mbps	FDD: 150 Mbps TDD: 130 Mbps
Max Data Rate Uplink	FDD: 50 Mbps	FDD: 50 Mbps TDD: 30 Mbps
Output Power	23 dBm +/- 2dB	23 dBm +/- 2dB
Sensitivity	FDD B1: -97 dBm FDD B3: -96 dBm FDD B5: -99 dBm FDD B7: -97 dBm FDD B8: -98 dBm FDD B20: -96 dBm	FDD B1: -101.6 dBm FDD B3: -101.9 dBm FDD B5: -102 dBm FDD B8: -102.1 dBm TDD B34: -101 dBm TDD B38: -101.3 dBm TDD B39: -101.2 dBm TDD B40: -101.4 dBm TDD B41: -101.4 dBm
AT Command Compliance	3GPP TS27.007 Enhanced AT commands	3GPP TS27.007 Enhanced AT commands

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. USRP B200 DATASHEET

This appendix contains relevant data about the USRP B200 and B210 (Table 16).

Table 16. USRP B200 and B210 Product Specifications. Adapted from [43].

Parameter	**USRP B200**	**USRP B210**
Channels	1 Tx & 1 Rx	2 TX & 2 Rx
FPGA	Xilinx Spartan 6 XC6SLX75	Xilinx Spartan 6 XC6SLX150
Instantaneous Bandwidth	56 MHz	56 MHz in 1x1 30.72 MHz in 2x2
Frequency Coverage	70 MHz - 6 GHz	70 MHz - 6 GHz
Power Output	Over 10 dBm	Over 10 dBm
Receiver Noise Figure	Less than 8 dB	Less than 8 dB
Max ADC/DAC Sample Rate	61.44 MS/s	61.44 MS/s
ADC/DAC Resolution	12 bits	12 bits

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. OAI EPC SOFTWARE INSTALLATION GUIDE

Section III.B.4 details the steps required to configure a host machine kernel for usage with the OAI EPC software. This appendix provides a detailed guide for each of the steps required to install and configure the OAI EPC software that are overviewed in Section III.B.5. The steps shown reflect a specific configuration of the OAI EPC with the Quectel EC20 UE and may require changes for other open stack network implementations.

A. PREREQUISITES AND INITIAL DOCKER SET-UP

Install Docker

1. Remove any old docker files if applicable

```
sudo apt-get remove docker docker-engine docker.io containerd runc
```

2. Set-up the Docker repository

```
sudo apt-get update

sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent \ software-
    properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/
    share/keyrings/docker-archive-keyring.gpg

echo \ “deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
    https://download.docker.com/linux/ubuntu \ $(lsb_release -cs) stable” | sudo tee
    /etc/apt/sources.list.d/docker.list > /dev/null
```

3. Install the Docker engine

```
sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io
```

4. Create a Docker hub account

5. Add the user to Docker group so that future commands do not require “sudo.” In this step, replace $USER with the actual username. After completion, the user must log out, then log back in before continuing.

```
sudo groupadd docker

sudo usermod -a -G docker $USER
```

6. Verify Docker installation with “Hello World” example

docker run hello-world

Check Python version

1. The installed Python version must be at least 3.6, which can be checked with the following command:

python3 –version

2. If this condition is not met, then use the package manager to update or install Python version 3.0 with:

sudo apt-get install python3

Pull Base Images

1. If not already done, log into the user’s Docker account
2. Pull the base images for Ubuntu 18.04 and Cassandra

docker pull ubuntu:bionic

docker pull Cassandra:2.1

3. If desired, the user may now logout of their Docker account

Network Configuration

1. Enable IP address forwarding

sudo sysctl net.ipv4.conf.all.forwarding=1

sudo iptables -P FORWARD ACCEPT

In /etc/sysctl.conf, uncomment the line net.ipv4.ip_forward

sudo sysctl -p /etc/sysctl.conf

Reboot the machine

2. Verify that IP forwarding is enabled by checking that the following command outputs a 1.

```
sysctl net.ipv4.ip_forward
```

3. Alter the Docker network bridge to an IP address on the local network

Add a file /etc/docker/deamon.json with the desired IP range as shown:

```
{
"bip": 192.168.17.1/24"
}
```

4. Restart the Docker daemon for the changes to take effect

```
sudo service docker restart

docker info

docker network inspect bridge
```

B. BUILD IMAGES

The series of commands required to create each of the EPC network function image files are consolidated in the script "Create_4G_Images.sh." The script only requires an initial clone of the OAI EPC repository.

1. Clone the OAI EPC repository

```
git clone https://github.com/OPENAIRINTERFACE/openair-epc-fed.git
```

2. Change directories to the downloaded openair-epc-fed folder

```
cd openair-epc-fed
```

3. Change desired branch to master and synchronize files with the repository

```
git checkout master

git pull origin master

./scripts/syncComponents.sh
```

4. Build the HSS image

```
docker build --target oai-hss –tag oai-hss:production --file component/oai-hss/docker/
    Dockerfile.ubuntu18.04 component/oai-hss
```

```
docker image prune --force

docker image ls
```

5. Build the MME image. This image is the only one that appears to stall for a significant period of time while building. The stall occurs around step 13 and 14, but it will successfully build if left alone.

```
docker build --target oai-mme –tag oai-mme:production --file component/oai-mme/docker/Dockerfile.ubuntu18.04 component/oai-mme

docker image prune --force

docker image ls
```

6. Build SPGWC

```
docker build --target oai-spgwc --tag oai-spgwc:production --file component/oai-spgwc/docker/Dockerfile.ubuntu18.04 component/oai-spgwc

docker image prune --force

docker image ls
```

7. Build SPGWU

```
docker build --target oai-spgwu --tag oai-spgwu:production --file component/oai-spgwu/docker/Dockerfile.ubuntu18.04 component/oai-spgwu

docker image prune --force

docker image ls
```

C. CREATE AND CONFIGURE CONTAINERS

The series of commands required to create each of the EPC network function containers are consolidated in the script "Configure_4G_Containers.sh." The first three steps are not included in the script. The first step creates a network within the docker environment for the containers to communicate with each other. This step only needs to be completed once. The second step creates a route on the eNB machine to the OAI EPC network on the EPC machine. The third step is not required but is provided as a set of

commands that can be used to permanently rename modern-style network interfaces to their more classical “eth0” style.

1. Create a network for the OAI EPC containers

```
docker network create --attachable --subnet 192.168.61.0/26 --ip-range 192.168.61.0/26 prod-oai-public-net
```

2. Create a route for the eNB to reach the EPC containers.

```
sudo ip route add 192.168.61.0/26 via 192.168.17.1 dev eth0
```

3. If desired, rename network interfaces similar to “enp0s25” to the more classical “eth0” style

```
sudo nano /etc/default/grub
```

Edit the line GRUB_CMDLINE_LINUX=“” to:
GRUB_CMDLINE_LINUX=“net.ifnames=0 biosdevname=0”

```
sudo grub-mkconfig -o /boot/grub/grub.cfg
```

Restart the computer

4. Create each container. This is the step where the “Configure_4G_Containers.sh” script begins and where a different approach for permanent deployment of the containers could be followed.

```
docker run --name prod-cassandra -d -e CASSANDRA_CLUSTER_NAME=“OAI HSS Cluster” -e CASSANDRA_ENDPOINT_SNITCH=GossipingPropertyFileSnitch cassandra:2.1

docker run --privileged --name prod-oai-hss -d --entrypoint /bin/bash oai-hss:production -c “sleep infinity”

docker network connect prod-oai-public-net prod-oai-hss

docker run --privileged --name prod-oai-mme --network prod-oai-public-net -d --entrypoint /bin/bash oai-mme:production -c “sleep infinity”

docker run --privileged --name prod-oai-spgwc --network prod-oai-public-net -d --entrypoint /bin/bash oai-spgwc:production -c “sleep infinity”

docker run --privileged --name prod-oai-spgwu-tiny --network prod-oai-public-net -d --entrypoint /bin/bash oai-spgwu-tiny:production -c “sleep infinity”
```

5. Configure the Cassandra container

```
docker cp component/oai-hss/src/hss_rel14/db/oai_db.cql prod-cassandra:/home

docker exec -it prod-cassandra /bin/bash -c "nodetool status"

Cassandra_IP=`docker inspect --
    format="{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}" prod-
    cassandra`

docker exec -it prod-cassandra /bin/bash -c "cqlsh --file /home/oai_db.cql
    ${Cassandra_IP}"
```

6. Configure the HSS container

```
HSS_IP=`docker exec -it prod-oai-hss /bin/bash -c "ifconfig eth1 | grep inet" | sed -f ./ci-
    scripts/convertIpAddrFromIfconfig.sed`

python3 component/oai-hss/ci-scripts/generateConfigFiles.py --kind=HSS --
    cassandra=${Cassandra_IP} \
    --hss_s6a=${HSS_IP} --apn1=apn1.carrier.com \
    --apn2=NPS4G.apn.epc.mnc092.mcc404.3gppnetwork.org \
    --users=200 --imsi=404921000000001 \
    --ltek=0c0a34601d4f07677303652c0462535b \
    --op=63bfa50ee6523365ff14c1f45f88737d \
    --nb_mmes=1 --from_docker_file

docker cp ./hss-cfg.sh prod-oai-hss:/openair-hss/scripts

docker exec -it prod-oai-hss /bin/bash -c "cd /openair-hss/scripts && chmod 777 hss-
    cfg.sh && ./hss-cfg.sh"
```

7. Configure the MME container

```
MME_IP=`docker inspect --
    format="{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}" prod-
    oai-mme`

SPGW0_IP=`docker inspect --
    format="{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}" prod-
    oai-spgwc`

python3 component/oai-mme/ci-scripts/generateConfigFiles.py --kind=MME \
    --hss_s6a=${HSS_IP} --mme_s6a=${MME_IP} \
    --mme_s1c_IP=${MME_IP} --mme_s1c_name=eth0 \
    --mme_s10_IP=${MME_IP} --mme_s10_name=eth0 \
    --mme_s11_IP=${MME_IP} --mme_s11_name=eth0 \
    --spgwc0_s11_IP=${SPGW0_IP} \
```

```
--mcc=404--mnc=92 --tac_list="5 6 7" --from_docker_file

docker cp ./mme-cfg.sh prod-oai-mme:/openair-mme/scripts

docker exec -it prod-oai-mme /bin/bash -c "cd /openair-mme/scripts && chmod 777
    mme-cfg.sh && ./mme-cfg.sh"
```

8. Configure the SPGW-C container

```
python3 component/oai-spgwc/ci-scripts/generateConfigFiles.py --kind=SPGW-C \
    --s11c=eth0 --sxc=eth0 --apn=apn1.carrier.com \
    --dns1_ip=172.20.20.12 --dns2_ip=8.8.8.8 --from_docker_file

docker cp ./spgwc-cfg.sh prod-oai-spgwc:/openair-spgwc

docker exec -it prod-oai-spgwc /bin/bash -c "cd /openair-spgwc && chmod 777 spgwc-
    cfg.sh && ./spgwc-cfg.sh"
```

9. Configure the SPGW-U-tiny container

```
python3 component/oai-spgwu-tiny/ci-scripts/generateConfigFiles.py --kind=SPGW-U \
    --sxc_ip_addr=${SPGW0_IP} --sxu=eth0 --s1u=eth0 \
    --network_ue_nat_option=yes --from_docker_file

docker cp ./spgwu-cfg.sh prod-oai-spgwu-tiny:/openair-spgwu-tiny

docker exec -it prod-oai-spgwu-tiny /bin/bash -c "cd /openair-spgwu-tiny && chmod 777
    spgwu-cfg.sh && ./spgwu-cfg.sh"
```

D. START NETWORK FUNCTIONS

The series of commands required to start the network functions requires a specific ordering: HSS, MME, SPGW-C, and SPGW-U-tiny. These commands initialize the network functions and start packet captures in each container. Additionally, a colored display of the MME log is printed to the screen. The script that contains these commands is "Launch_4G_Containers.sh."

1. Launch tshark in each container to start packet captures

```
docker exec -d prod-oai-hss /bin/bash -c "nohup tshark -i eth0 -i eth1 -w /tmp/
    hss_check_run.pcap 2>&1 > /dev/null"

docker exec -d prod-oai-mme /bin/bash -c "nohup tshark -i eth0 -i lo:s10 -w /tmp/
    mme_check_run.pcap 2>&1 > /dev/null"
```

```
docker exec -d prod-oai-spgwc /bin/bash -c “nohup tshark -i eth0 -i lo:p5c -i lo:s5c -w
    /tmp/spgwc_check_run.pcap 2>&1 > /dev/null”

docker exec -d prod-oai-spgwu-tiny /bin/bash -c “nohup tshark -i eth0 -w /tmp/
    spgwu_check_run.pcap 2>&1 > /dev/null”
```

2. Launch each network function

```
docker exec -d prod-oai-hss /bin/bash -c “nohup ./bin/oai_hss -j ./etc/hss_rel14.json --
    reloadkey true > hss_check_run.log 2>&1”

sleep 2

docker exec -d prod-oai-mme /bin/bash -c “nohup ./bin/oai_mme -c ./etc/mme.conf >
    mme_check_run.log 2>&1”

sleep 2

docker exec -d prod-oai-spgwc /bin/bash -c “nohup ./bin/oai_spgwc -o -c ./etc/
    spgw_c.conf > spgwc_check_run.log 2>&1”

sleep 2

docker exec -d prod-oai-spgwu-tiny /bin/bash -c “nohup ./bin/oai_spgwu -o -c ./etc/
    spgw_u.conf > spgwu_check_run.log 2>&1”
```

3. Print the running MME log to screen

```
docker exec -it prod-oai-mme tail -f mme_check_run.log
```

E. STOPPING NETWORK FUNCTIONS

The series of commands required to stop the network functions requires the same ordering as the commands to start the network functions: HSS, MME, SPGW-C, and SPGW-U-tiny. These commands stop the network functions by sending each container a “killall” signal twice. Additionally, the logs, configuration files, and packet captures are consolidated into a compressed archive folder, which must be renamed after executing the script to prevent the archive from being overwritten. The script is called “Stop_4G_Containers.sh.”

1. Send the “killall” signal to each container twice

```
docker exec -it prod-oai-hss /bin/bash -c “killall --signal SIGINT oai_hss tshark
    tcpdump”

docker exec -it prod-oai-mme /bin/bash -c “killall --signal SIGINT oai_mme tshark
    tcpdump”

docker exec -it prod-oai-spgwc /bin/bash -c “killall --signal SIGINT oai_spgwc tshark
    tcpdump”

docker exec -it prod-oai-spgwu-tiny /bin/bash -c “killall --signal SIGINT oai_spgwu
    tshark tcpdump”

sleep 10

docker exec -it prod-oai-hss /bin/bash -c “killall --signal SIGKILL oai_hss tshark
    tcpdump”

docker exec -it prod-oai-mme /bin/bash -c “killall --signal SIGKILL oai_mme tshark
    tcpdump”

docker exec -it prod-oai-spgwc /bin/bash -c “killall --signal SIGKILL oai_spgwc tshark
    tcpdump”

docker exec -it prod-oai-spgwu-tiny /bin/bash -c “killall --signal SIGKILL oai_spgwu
    tshark tcpdump”
```

2. Remove the old archive folder and create a new archive folder

```
rm -Rf archives

mkdir -p archives/oai-hss-cfg archives/oai-mme-cfg archives/oai-spgwc-cfg archives/oai-
    spgwu-cfg
```

3. Copy the configurations, logs, and packet captures into the new archive

```
docker cp prod-oai-hss:/openair-hss/etc/. archives/oai-hss-cfg

docker cp prod-oai-mme:/openair-mme/etc/. archives/oai-mme-cfg

docker cp prod-oai-spgwc:/openair-spgwc/etc/. archives/oai-spgwc-cfg

docker cp prod-oai-spgwu-tiny:/openair-spgwu-tiny/etc/. archives/oai-spgwu-cfg

docker cp prod-oai-hss:/openair-hss/hss_check_run.log archives

docker cp prod-oai-mme:/openair-mme/mme_check_run.log archives

docker cp prod-oai-spgwc:/openair-spgwc/spgwc_check_run.log archives
```

```
docker cp prod-oai-spgwu-tiny:/openair-spgwu-tiny/spgwu_check_run.log archives

docker cp prod-oai-hss:/tmp/hss_check_run.pcap archives

docker cp prod-oai-mme:/tmp/mme_check_run.pcap archives

docker cp prod-oai-spgwc:/tmp/spgwc_check_run.pcap archives

docker cp prod-oai-spgwu-tiny:/tmp/spgwu_check_run.pcap archives
```

4. Compress the newly created archive for storage

```
zip -r -qq docker_files.zip archives
```

APPENDIX F. DIAGPARSER AND MINICOM GUIDE

This appendix summarizes a series of commands originally presented in an ECE department laboratory that explored the EC20 modem. These commands are divided into four sections: the first to capture packets from the modem, the second to send and receive AT commands, the third to configure GNSS functions, and the fourth to access the Linux OS on the modem.

A. CONFIGURE PACKET CAPTURE

1. In a terminal window, run “nc -u -l -p 4729”
2. In a second terminal, run “sudo wireshark -I lo -f ‘port 4729’ -k

B. SEND AND RECEIVE MODEM-LEVEL COMMANDS

1. Open a terminal window and launch the minicom utility with “sudo minicom -s.” Within the utility, select “setup” and change the path to “/dev/ttyUSB0.” Escape and exit the menu. The raw byte level messages from the modem will now display on the screen.
2. In a second terminal window, change directory to the DiagParser utility with “cd ~/diag_parser.” Run the utility with “sudo ./diag_parser -g 127.0.0.1 -i /dev/ttyUSB0 -v” to decode the packets received by the modem.
3. In a third terminal window, run “sudo minicom -s.” Select “setup” and change the device path to “/dev/ttyUSB2.” Escape and exit the menu. The modem will attempt to execute its initialization AT commands. To make the user input visible, type “ATE1” and hit the return key.

C. CONFIGURE GNSS FUNCTIONALITY

1. After configuring the modem to send and receive AT commands, type “AT+QGPS=1” and hit return. The modem should respond with “ok.”

2. Open a new terminal window and run “sudo minicom -s.” Select “setup” and change the device path to “/dev/ttyUSB1.” Escape and exit the menu. The modem will display NMEA data in the terminal window.

D. ACCESS MODEM EMBEDDED LINUX OS

1. Open a terminal window and run “sudo adb start-server.”
2. In the same window, run “sudo minicom -s,” select “setup” and change the device path to “/dev/ttyUSB2.”
3. Once the modem has completed its initial set of AT commands, type “ATE1” and hit return to make user-input visible. Then enter the following command
4. AT+QLINUXCMD=“/usr/bin/usb_uartdiag”
5. Once executed the previous command will cause minicom to display “Cannot open /dev/ttyUSB2!.”
6. In a new terminal, run “adb shell” to enter the Linux OS on the modem.

APPENDIX G. NETWORK TESTING FILE GUIDE

This appendix contains a guide to the network tests performed for this thesis and the file structure used to store the data gathered from each test.

A. FILE STRUCTURE

Oai-hss-cfg

- “acl.conf”: Configuration file for peer whitelist extension
- “cacert.pem”: CA certificate file
- "hss.cert.pem”: Signed certificate and HSS public key
- “hss.key.pem”: HSS private key (RSA)
- “hss_rel14.conf”: Contains IP address for the Cassandra server (MySQL)
- “hss_rel14_fd.conf”: Free Diameter configuration information for HSS including preferences for TCP/SCTP, listening IP address and port numbers
- “hss_rel14.json”: Information for connecting HSS to docker bridge including the username/password for the VHSS database
- “oss.json”: Configuration information for logging structure

Oai-mme-cfg

- “mme.cacert.pem”: CA certificate file
- “mme.cakey.pem”: MME certificate private key
- “mme.cert.pem”: Signed certificate and MME public key

- “mme.conf”: Configuration file, including interface IP addresses, allowed PLMNs, max number of UE/eNBs, emergency signaling provisioning, logging preferences galore
- “mme_fd.conf”: Free Diameter configuration information for MME including an option for TLS security in connection with HSS
- “mme.key.pem”: MME private key (RSA)

Oai-spgwc-cfg

- “spgw_c.conf”: configuration file for SPGW-C, including UE address pool and primary/secondary DNS servers

Oai-spgwu-cfg

- “spgw_u.conf”: configuration file for SPGW-U, including NAT option

Log Files and Packet Capture

- hss_check_run.log
- hss_check_run.pcap
- mme_check_run.log
- mme_check_run.pcap
- spgwc_check_run.log
- spgwc_check_run.pcap
- spgwu_check_run.log
- spgwu_check_run.pcap

B. NETWORK TESTING ARCHIVE GUIDE

No.	Description	Components	RF and Networking HW
1	No connectivity between UE/eNB	UE/eNB	B200/X310 respectively; SMA connected with 30 dB attenuator
2	eNB fail to attach	UE/eNB	B200/X310 respectively; SMA connected with 30 dB attenuator
3	UE attach to eNB, but RX gain too high	UE/eNB	B200/X310 respectively; SMA connected with 30 dB attenuator
4	Multiple attach/detach until finally UE attaches to eNB	UE/eNB	B200/X310 respectively; SMA connected with 30 dB attenuator
5	No RF full system attach and connect; UE unable to ping Internet	UE/eNB/EPC	Laptops only via Ethernet switch
6	No RF full system attach and connect; successful UE ping Internet	UE/eNB/EPC	Laptops only via Ethernet switch; EPC as NAT from eth0<->wlan0
7	RF over SMA cables; successful UE connect/attach and ping to Internet	UE/eNB/EPC	eNB and EPC via Ethernet; UE to eNB via USRPS via SMA with 30 dB attenuator; EPC as NAT from eth0<->wlan0
8	RF over SMA cables; UE attach rejected (T-Mobile and Netherlands SIMs)	EC20/eNB/EPC	eNB and EPC via Ethernet; EC20 to eNB (USRP B200) via SMA with 30 dB attenuator; EPC as NAT from eth0<->wlan0; Netherlands KPN Mobile SIM
9	RF over SMA cables; UE attach rejected (My 404/92 SIM)	EC20/eNB/EPC	eNB and EPC via Ethernet; EC20 to eNB (USRP B200) via SMA with 30 dB attenuator; EPC as NAT from eth0<->wlan0; My 404/92 Programmed SIM
10	RF over SMA cables; UE attach rejected (My 404/92 SIM) after fixing SPGWC	EC20/eNB/EPC	eNB and EPC via Ethernet; EC20 to eNB (USRP B200) via SMA with 30 dB attenuator; EPC as NAT from eth0<->wlan0; My 404/92 Programmed SIM

11	RF over SMA cables; successful soft UE connect/attach and ping to Internet	UE/eNB/EPC	eNB and EPC via Ethernet; soft UE to eNB (USRP B200) via SMA with 30 dB attenuator; EPC as NAT from eth0<->wlan0; My 404/92 Programmed SIM
12	RF over SMA cables; successful soft UE connect/attach and ping to Internet without nokrnmod flag	UE/eNB/EPC	eNB and EPC via Ethernet; soft UE to eNB (USRP B200) via SMA with 30 dB attenuator; EPC as NAT from eth0<->wlan0; My 404/92 Programmed SIM
13	RF over SMA cables; successful soft UE connect/attach and ping to Internet with old nasmesh kernel option	UE/eNB/EPC	eNB and EPC via Ethernet; soft UE to eNB (USRP B200) via SMA with 30 dB attenuator; EPC as NAT from eth0<->wlan0; My 404/92 Programmed SIM

APPENDIX H. USRP SOFTWARE GUIDE

The following code is used to provide vendor information for several USRP devices to the OS. This information is used to inform the OS how to interact with the USRP over the USB connection. The file is created and saved at "/etc/udev/rules.d/10-usrp.rules".

```
#
# Copyright 2011,2015 Ettus Research LLC
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <http://www.gnu.org/licenses/>.
#
#USRP1
SUBSYSTEMS=="usb," ATTRS{idVendor}=="fffe," ATTRS{idProduct}=="0002,"
MODE:="0666"
#B100
SUBSYSTEMS=="usb," ATTRS{idVendor}=="2500," ATTRS{idProduct}=="0002,"
MODE:="0666"
#B200
SUBSYSTEMS=="usb," ATTRS{idVendor}=="2500," ATTRS{idProduct}=="0020,"
MODE:="0666"
SUBSYSTEMS=="usb," ATTRS{idVendor}=="2500," ATTRS{idProduct}=="0022,"
MODE:="0666"
SUBSYSTEMS=="usb," ATTRS{idVendor}=="3923," ATTRS{idProduct}=="7813,"
MODE:="0666"
SUBSYSTEMS=="usb," ATTRS{idVendor}=="3923," ATTRS{idProduct}=="7814,"
MODE:="0666"
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX I. EC20 MODEM AT COMMANDS

This appendix contains the output from execution of the AT+CLAC command via the minicom utility and lists all commands that can be executed by the EC20. While documentation of some commands is provided by Quectel, many commands do not have provided documentation. Of these undocumented commands, some can be deduced through other resources such as [38]. The remaining commands are likely proprietary to either Qualcomm or Quectel. The results are presented in three columns and exclude the AT prefix or any arguments required to successfully execute the command.

&C	S7	+QFPOSITION
&D	S8	+QFTUCAT
&E	S9	+QFSEEK
&F	S10	+QFREAD
&S	S11	+QFWRITE
&V	S30	+QFDELROOTFILE
&W	S103	+QUSBWAITTIME
E	S104	+QPRTPARA
I	+FCLASS	+CBST
L	+ICF	+CRLP
M	+IFC	+CV120
Q	+IPR	+CHSN
V	+GMI	+CSSN
X	+GMM	+CREG
Z	+GMR	+CGREG
T	+GCAP	+CEREG
P	+GSN	+CSCS
\Q	+DR	+CSTA
\S	+DS	+CR
\V	+CMEE	+CEER
%V	+WS46	+CRC
D	+PACSP	+CGDCONT
A	+CFUN	+CGDSCONT
H	+CLCC	+CGTFT
O	+QFLDS	+CGEQREQ
S0	+QFLST	+CGEQMIN
S2	+QFUPL	+CGEQOS
S3	+QFDWL	+CGCONTRDP
S4	+QFDEL	+CGSCONTRDP
S5	+QFOPEN	+CGTFTRDP
S6	+QFCLOSE	+CGEQOSRDP

+CGQREQ
+CGQMIN
+CGEREP
+CGPADDR
+CGDATA
+CGCLASS
+CGPIAF
+CGSMS
+CSMS
+CMGF
+CSAS
+CRES
+CSCA
+CSMP
+CSDH
+CSCB
+FDD
+FAR
+FCL
+FIT
+ES
+ESA
+CMOD
+CEMODE
+CVHU
+CECALL
+CFUN
+QPOWD
+QSCLK
+QFASTBOOT
+QPRINT
+QFOTADL
+QNDISDUP
+QNAND
+CMUT
+QAUDLOOP
+QDAI
+CLVL
+QSIDET
+QAUDMOD
+QEEC
+QAUDRD
+QAUDPLAY
+QAUDSTOP
+QTONEDET
+QMIC
+QRXGAIN
+QTTS
+QTTSETUP
+QIIC
+QAUDLPVOL
+QPCMV
+QAUDVOL
+QWIFI
+QWSSID
+QWSSIDHEX
+QWBCAST
+QWAUTH
+QWMOCH
+QWISO
+QWDHCP
+QWNAT
+QWCLICNT
+QWDCNT
+QWRSTD
+QWCLIP
+QWSETMAC
+QWSERVER
+QWIFIRESP
+QWRFLVL
+QWCLILST
+QWCLIRM
+QWTOCLIEN
+QWTOCLI
+QWCFG
+QDATAFWD
+QWPARAM
+QLINUXCMD
+QCFG
+QLOPEN
+QLCLOSE
+QLREAD
+QLWRITE
$QCSIMSTAT
$QCPBMPREF
$CREG
$CCLK
$QCCNMI
*CNTI
$QCCLR
$QCDMR
$QCDNSP
$QCDNSS
$QCTER
$QCSLOT
$QCSIMAPP
$QCPINSTAT
$QCPDPP
$QCPDPLT
$QCPWRDN
$QCDGEN
$QCPDPCFGE
$BREW
$QCANTE
$QCRPW
$QCSQ
$CSQ
$QCSYSMODE
$QCCTM
$QCBANDPREF
^PREFMODE
^SYSINFO
^SYSCONFIG
^CARDMODE
^DSCI
$QCVOLT
$QCHWREV
$QCBOOTVER
$ECALL
$QCDEFPROF
$QCMRUE
$QCMRUC
$QCAPNE
$QCPDPIMSCFGE
$QCCLAC
^SPN
$QCRMCALL
$QCDRX
$QCRSRP
$QCRSRQ
$QCATMOD

APPENDIX J. EC20 NETWORK OPERATOR LIST

This appendix contains the output from execution of the AT+COPN command via the minicom utility and lists all MNOs stored on the EC20. The results are presented in two columns. Each entry contains the PLMN of the MNO and the operator name in long alphanumeric format. Although not shown, some MNOs have an additional shot alphanumeric format to facilitate displaying on the UE.

+COPN: "00011","Uganda Telecom"
+COPN: "00011","INX Inmarsat"
+COPN: "00101","Test PLMN 1-1"
+COPN: "00102","Test PLMN 1-2"
+COPN: "00201","Test PLMN 2-1"
+COPN: "20201","GR COSMOTE"
+COPN: "20201","GR COSMOTE"
+COPN: "20205","vodafone GR"
+COPN: "20210","WIND GR"
+COPN: "20210","WIND GR"
+COPN: "20210","WIND GR"
+COPN: "20404","vodafone NL"
+COPN: "20404","vodafone NL"
+COPN: "20408","NL KPN"
+COPN: "20408","NL KPN"
+COPN: "20412","NL Telfort"
+COPN: "20416","T-Mobile NL"
+COPN: "20416","T-Mobile NL"
+COPN: "20420","Orange NL"
+COPN: "20601","BEL PROXIMUS"
+COPN: "20601","BEL PROXIMUS"
+COPN: "20610","B Mobistar"
+COPN: "20610","B mobistar"
+COPN: "20610","B mobistar"
+COPN: "20620","BASE"
+COPN: "20620","BASE"
+COPN: "20801","Orange F"
+COPN: "20801","Orange F"
+COPN: "20801","Orange F"
+COPN: "20802","F - Contact"
+COPN: "20802","F - Contact"
+COPN: "20810","F SFR"
+COPN: "20810","F SFR"
+COPN: "20810","F SFR"
+COPN: "20813","F - Contact"
+COPN: "20813","F - Contact"
+COPN: "20815","Free"
+COPN: "20815","Free"
+COPN: "20820","F-Bouygues Telecom"
+COPN: "20820","F-Bouygues Telecom"
+COPN: "20820","F-Bouygues Telecom"
+COPN: "20888","F - Contact"
+COPN: "20888","F - Contact"
+COPN: "21201","vala"
+COPN: "21210","Monaco"
+COPN: "21303","MOBILAND"
+COPN: "21303","MOBILAND"
+COPN: "21303","MOBILAND"
+COPN: "21401","Vodafone ES"
+COPN: "21401","Vodafone ES"
+COPN: "21401","Vodafone ES"
+COPN: "21403","Orange"
+COPN: "21403","Orange"
+COPN: "21403","Orange"
+COPN: "21404","Yoigo"
+COPN: "21407","Movistar"
+COPN: "21407","Movistar"
+COPN: "21407","Movistar"
+COPN: "21601","Telenor HU"
+COPN: "21601","Telenor HU"
+COPN: "21601","Telenor HU"
+COPN: "21630","T-Mobile H"
+COPN: "21630","T-Mobile H"
+COPN: "21670","vodafone HU"
+COPN: "21670","vodafone HU"

+COPN: "21803","HT ERONET"
+COPN: "21805","m:tel"
+COPN: "21805","m:tel"
+COPN: "21890","BH Mobile"
+COPN: "21890","BHMobile"
+COPN: "21890","BHMobile"
+COPN: "21901","T-Mobile HR"
+COPN: "21902","Tele2 HR"
+COPN: "21910","HR VIP"
+COPN: "22001","Telenor SRB"
+COPN: "22001","Telenor SRB"
+COPN: "22002","ProMonte"
+COPN: "22002","ProMonte"
+COPN: "22003","mt:s"
+COPN: "22003","mt:s"
+COPN: "22003","mt:s"
+COPN: "22004","T-Mobile CG"
+COPN: "22005","Vip SRB"
+COPN: "22005","Vip SRB"
+COPN: "22005","Vip SRB"
+COPN: "22201","I TIM"
+COPN: "22201","I TIM"
+COPN: "22201","I TIM"
+COPN: "22210","vodafone IT"
+COPN: "22210","vodafone IT"
+COPN: "22288","I WIND"
+COPN: "22288","I WIND"
+COPN: "22299","3 ITA"
+COPN: "22601","RO Vodafone RO"
+COPN: "22601","RO Vodafone RO"
+COPN: "22601","RO Vodafone RO"
+COPN: "22603","RO COSMOTE"
+COPN: "22603","RO COSMOTE"
+COPN: "22605","RO Digi.Mobil"
+COPN: "22610","RO ORANGE"
+COPN: "22610","RO ORANGE"
+COPN: "22801","Swisscom"
+COPN: "22801","Swisscom"
+COPN: "22801","Swisscom"
+COPN: "22802","Sunrise"
+COPN: "22802","Sunrise"
+COPN: "22802","Sunrise"
+COPN: "22803","orange CH"
+COPN: "22807","In&Phone"
+COPN: "22808","CHE Tele2 Switzerland"
+COPN: "22815","OnAir"
+COPN: "23001","T-Mobile CZ"
+COPN: "23001","T-Mobile CZ"
+COPN: "23002","O2 - CZ"
+COPN: "23002","O2 - CZ"
+COPN: "23003","Vodafone CZ"
+COPN: "23003","Vodafone CZ"
+COPN: "23101","Orange SK"
+COPN: "23101","Orange SK"
+COPN: "23102","Telekom SK"
+COPN: "23102","Telekom SK"
+COPN: "23102","Telekom SK"
+COPN: "23106","O2 - SK"
+COPN: "23106","O2 - SK"
+COPN: "23201","A1"
+COPN: "23201","A1"
+COPN: "23201","A1"
+COPN: "23203","T-Mobile Austria"
+COPN: "23203","T-Mobile Austria"
+COPN: "23203","T-Mobile Austria"
+COPN: "23205","Orange AT"
+COPN: "23205","Orange AT"
+COPN: "23207","tele - ring"
+COPN: "23207","tele - ring"
+COPN: "23207","tele - ring"
+COPN: "23210","3 AT"
+COPN: "23212","Orange AT"
+COPN: "23212","Orange AT"
+COPN: "23401","UK01"
+COPN: "23403","Airtel-Vodafone"
+COPN: "23403","Airtel-Vodafone"
+COPN: "23407","Cable&Wireless UK"
+COPN: "23409","PMN UK"
+COPN: "23410","O2 - UK"
+COPN: "23410","O2 - UK"
+COPN: "23410","O2 - UK"
+COPN: "23410","O2 - UK"
+COPN: "23415","vodafone UK"
+COPN: "23415","vodafone UK"
+COPN: "23416","Opal UK"
+COPN: "23419","PMN UK"
+COPN: "23420","3 UK"
+COPN: "23428","Marathon"

+COPN: "23428","Marathon"
+COPN: "23428","Marathon"
+COPN: "23430","EE"
+COPN: "23431","EE"
+COPN: "23432","EE"
+COPN: "23433","EE"
+COPN: "23450","JT"
+COPN: "23450","JT"
+COPN: "23450","JT"
+COPN: "23455","C&W (SURE)"
+COPN: "23455","C&W (SURE)"
+COPN: "23455","C&W (SURE)"
+COPN: "23458","Manx Telecom"
+COPN: "23458","Manx Telecom"
+COPN: "23801","TDC Mobil"
+COPN: "23801","TDC Mobil"
+COPN: "23801","TDC Mobil"
+COPN: "23802","Telenor DK"
+COPN: "23802","Telenor DK"
+COPN: "23806","3 DK"
+COPN: "23820","TELIA DK"
+COPN: "23820","TELIA DK"
+COPN: "23866","Telia Telenor DK"
+COPN: "23866","Telia Telenor DK"
+COPN: "23866","Telia Telenor DK"
+COPN: "23877","Telenor DK"
+COPN: "23877","Telenor DK"
+COPN: "24001","TELIA S"
+COPN: "24001","TELIA S"
+COPN: "24002","3 SE"
+COPN: "24004","SWEDEN"
+COPN: "24005","Sweden 3G"
+COPN: "24007","Tele2 SE"
+COPN: "24008","Telenor SE"
+COPN: "24008","Telenor SE"
+COPN: "24008","Telenor SE"
+COPN: "24010","SpringMobil SE"
+COPN: "24010","SpringMobil SE"
+COPN: "24024","Sweden Mobile"
+COPN: "24024","Sweden Mobile"
+COPN: "24201","N Telenor"
+COPN: "24201","N Telenor"
+COPN: "24202","N NetCom"
+COPN: "24202","N NetCom"
+COPN: "24203","MTU"
+COPN: "24204","Mobile Norway"
+COPN: "24205","Mobile Norway"
+COPN: "24205","Mobile Norway"
+COPN: "24403","DNA"
+COPN: "24403","DNA"
+COPN: "24405","FI elisa"
+COPN: "24405","FI elisa"
+COPN: "24405","FI elisa"
+COPN: "24412","DNA"
+COPN: "24412","DNA"
+COPN: "24412","DNA"
+COPN: "24414","FI AMT"
+COPN: "24491","FI SONERA"
+COPN: "24491","FI SONERA"
+COPN: "24601","OMNITEL LT"
+COPN: "24602","LT BITE GSM"
+COPN: "24602","LT BITE GSM"
+COPN: "24603","Tele2 LT"
+COPN: "24603","Tele2 LT"
+COPN: "24701","LV LMT"
+COPN: "24701","LV LMT"
+COPN: "24701","LV LMT"
+COPN: "24702","Tele2 LV"
+COPN: "24702","Tele2 LV"
+COPN: "24705","BITE LV"
+COPN: "24705","BITE LV"
+COPN: "24705","BITE LV"
+COPN: "24801","EE EMT"
+COPN: "24801","EE EMT"
+COPN: "24801","EE EMT"
+COPN: "24802","EE elisa"
+COPN: "24802","EE elisa"
+COPN: "24802","EE elisa"
+COPN: "24803","Tele2 EE"
+COPN: "24803","Tele2 EE"
+COPN: "25001","MTS-RUS"
+COPN: "25001","MTS-RUS"
+COPN: "25002","MegaFon RUS"
+COPN: "25002","MegaFon RUS"
+COPN: "25003","ROSTELECOM"
+COPN: "25004","SIBCHALLENGE
RUS"
+COPN: "25005","ROSTELECOM"
+COPN: "25005","ROSTELECOM"

+COPN: "25007","RUS 07, RUS SMARTS"
+COPN: "25007","RUS 07, RUS SMARTS"
+COPN: "25010","RUS DTC"
+COPN: "25012","ROSTELECOM"
+COPN: "25012","ROSTELECOM"
+COPN: "25013","RUS Kuban-GSM"
+COPN: "25013","RUS Kuban-GSM"
+COPN: "25015","RUS15, RUS SMARTS"
+COPN: "25016","RUS16,250 16"
+COPN: "25016","RUS16,250 16"
+COPN: "25017","ROSTELECOM"
+COPN: "25017","ROSTELECOM"
+COPN: "25019","RUS INDIGO"
+COPN: "25020","TELE2"
+COPN: "25020","TELE2 RU"
+COPN: "25028","RUS Beeline"
+COPN: "25035","MOTIV"
+COPN: "25037","KODOTEL"
+COPN: "25038","ROSTELECOM"
+COPN: "25039","ROSTELECOM"
+COPN: "25039","ROSTELECOM"
+COPN: "25099","Beeline"
+COPN: "25099","Beeline"
+COPN: "25501","MTS UKR"
+COPN: "25501","MTS UKR"
+COPN: "25502","Beeline UA"
+COPN: "25502","Beeline UA"
+COPN: "25503","UA-KYIVSTAR"
+COPN: "25503","UA-KYIVSTAR"
+COPN: "25505","UA-GT"
+COPN: "25506","life:)"
+COPN: "25506","life:)"
+COPN: "25507","TriMob"
+COPN: "25701","BY VELCOM"
+COPN: "25701","BY VELCOM"
+COPN: "25702","MTS BY"
+COPN: "25702","MTS BY"
+COPN: "25704","life:) BY"
+COPN: "25704","life:) BY"
+COPN: "25704","life:) BY"
+COPN: "25901","Orange MD"
+COPN: "25902","MD MOLDCELL"
+COPN: "25904","MDA EVENTIS"
+COPN: "25904","MDA EVENTIS"
+COPN: "25905","UNITE"
+COPN: "26001","Plus"
+COPN: "26001","Plus"
+COPN: "26001","Plus"
+COPN: "26002","T-Mobile.pl"
+COPN: "26002","T-Mobile.pl"
+COPN: "26003","Orange PL"
+COPN: "26003","Orange PL"
+COPN: "26003","Orange PL"
+COPN: "26006","Play"
+COPN: "26006","Play"
+COPN: "26201","Telekom.de"
+COPN: "26201","Telekom.de"
+COPN: "26201","Telekom.de"
+COPN: "26202","Vodafone.de"
+COPN: "26202","Vodafone.de"
+COPN: "26202","Vodafone.de"
+COPN: "26203","E-Plus"
+COPN: "26207","o2 - de"
+COPN: "26208","o2 - de"
+COPN: "26601","GIBTEL GSM"
+COPN: "26606","CTS"
+COPN: "26609","Shine"
+COPN: "26609","Shine"
+COPN: "26801","vodafone P"
+COPN: "26801","vodafone P"
+COPN: "26803","P OPTIMUS"
+COPN: "26803","P OPTIMUS"
+COPN: "26806","P TMN"
+COPN: "26806","P TMN"
+COPN: "26806","P TMN"
+COPN: "27001","L LUXGSM"
+COPN: "27001","L LUXGSM"
+COPN: "27077","L TANGO"
+COPN: "27077","L TANGO"
+COPN: "27099","L Orange-LU"
+COPN: "27099","L Orange-LU"
+COPN: "27099","L Orange-LU"
+COPN: "27201","vodafone IE"
+COPN: "27201","vodafone IE"
+COPN: "27202","o2 IRL"
+COPN: "27202","o2 IRL"
+COPN: "27202","o2 IRL"

+COPN: "27203","IRL - METEOR"
+COPN: "27203","IRL - METEOR"
+COPN: "27205","3 IRL"
+COPN: "27401","Siminn"
+COPN: "27401","Siminn"
+COPN: "27401","Siminn"
+COPN: "27402","Vodafone"
+COPN: "27402","Vodafone"
+COPN: "27404","Viking"
+COPN: "27407","IS-IceCell"
+COPN: "27408","On-waves"
+COPN: "27408","On-waves"
+COPN: "27411","NOVA IS"
+COPN: "27412","Siminn"
+COPN: "27412","Siminn"
+COPN: "27412","Siminn"
+COPN: "27601","AMC - AL"
+COPN: "27601","AMC - AL"
+COPN: "27602","Vodafone AL"
+COPN: "27602","Vodafone AL"
+COPN: "27603","EAGLE AL"
+COPN: "27603","EAGLE AL"
+COPN: "27604","PLUS AL"
+COPN: "27604","PLUS AL"
+COPN: "27801","vodafone MT"
+COPN: "27821","go mobile"
+COPN: "27877","3GT MT"
+COPN: "28001","CYTAMOBILE-VODAFONE"
+COPN: "28001","CYTAMOBILE-VODAFONE"
+COPN: "28010","MTN"
+COPN: "28010","MTN"
+COPN: "28010","MTN"
+COPN: "28201","GEO-GEOCELL"
+COPN: "28201","GEO-GEOCELL"
+COPN: "28202","MAGTI-GSM-GEO"
+COPN: "28202","MAGTI-GSM-GEO"
+COPN: "28204","BEELINE GE"
+COPN: "28300","MTS ARM"
+COPN: "28301","Beeline AM"
+COPN: "28301","Beeline AM"
+COPN: "28305","MTS ARM"
+COPN: "28305","MTS ARM"
+COPN: "28305","MTS ARM"
+COPN: "28310","Orange AM"
+COPN: "28310","Orange AM"
+COPN: "28310","Orange AM"
+COPN: "28401","Mtel"
+COPN: "28401","Mtel"
+COPN: "28401","Mtel"
+COPN: "28403","Vivacom"
+COPN: "28403","Vivacom"
+COPN: "28403","Vivacom"
+COPN: "28405","BG GLOBUL"
+COPN: "28405","BG GLOBUL"
+COPN: "28601","TR TURKCELL"
+COPN: "28602","VODAFONE TR"
+COPN: "28603","AVEA"
+COPN: "28801","Foroya Tele"
+COPN: "28801","Foroya Tele"
+COPN: "28802","VODAFONE FO"
+COPN: "28802","VODAFONE FO"
+COPN: "29001","TELE Greenland"
+COPN: "29266","SMT"
+COPN: "29266","SMT"
+COPN: "29340","Si.mobil"
+COPN: "29340","Si.mobil"
+COPN: "29341","MOBITEL"
+COPN: "29341","MOBITEL"
+COPN: "29364","T-2"
+COPN: "29370","SI TUSMOBIL"
+COPN: "29370","SI TUSMOBIL"
+COPN: "29401","T-Mobile MK"
+COPN: "29402","ONE MK"
+COPN: "29402","ONE MK"
+COPN: "29403","Vip MKD"
+COPN: "29403","Vip MKD"
+COPN: "29501","SwisscomFL"
+COPN: "29501","SwisscomFL"
+COPN: "29502","Orange FL"
+COPN: "29502","Orange FL"
+COPN: "29505","FL1"
+COPN: "29505","FL1"
+COPN: "29505","FL1"
+COPN: "29577","LI TANGO"
+COPN: "295295","SwisscomFL"
+COPN: "29702","Telekom.me"
+COPN: "29702","Telekom.me"
+COPN: "29702","Telekom.me"

+COPN: "30264","Bell"
+COPN: "30266","MTS"
+COPN: "30268","CANST"
+COPN: "30272","CAN Rogers Wireless Inc."
+COPN: "302072","CAN Rogers Wireless Inc."
+COPN: "30286","TELUS"
+COPN: "302220","TELUS"
+COPN: "302270","Eastlink"
+COPN: "302320","Mobilicity"
+COPN: "302340","Execulink Telecom"
+COPN: "302350","CANFN"
+COPN: "302370","Fido"
+COPN: "302370","Fido"
+COPN: "302370","Fido"
+COPN: "302370","Fido"
+COPN: "302380","DMTS GSM"
+COPN: "302490","WIND"
+COPN: "302610","Bell"
+COPN: "302610","Bell"
+COPN: "302610","Bell"
+COPN: "302660","MTS"
+COPN: "302720","Rogers Wireless"
+COPN: "302720","Rogers Wireless"
+COPN: "302730","CAN TerreStar Solutions"
+COPN: "302770","CANRU"
+COPN: "302780","SaskTel"
+COPN: "302880","FastRoam"
+COPN: "302940","Wightman"
+COPN: "30801","SPM AMERIS"
+COPN: "310000","NEP Wireless"
+COPN: "310002","NEP Wireless"
+COPN: "310009","NEP Wireless"
+COPN: "31020","US - Union Telephone"
+COPN: "310020","US - Union Telephone"
+COPN: "310026","T-Mobile"
+COPN: "31030","Centennial Wireless"
+COPN: "310030","Centennial Wireless"
+COPN: "310031","T-Mobile"
+COPN: "310032","IT&E"
+COPN: "31040","Cellular One"
+COPN: "310040","Cellular One"
+COPN: "310046","USA SIMMETRY"
+COPN: "31050","DIGICEL"
+COPN: "310050","DIGICEL"
+COPN: "31070","AT&T"
+COPN: "310070","AT&T"
+COPN: "31070","AT&T"
+COPN: "31070","AT&T"
+COPN: "310077","Iowa Wireless USA"
+COPN: "31080","Corr Wireless"
+COPN: "310080","Corr Wireless"
+COPN: "31090","AT&T"
+COPN: "310090","AT&T"
+COPN: "31090","AT&T"
+COPN: "31090","AT&T"
+COPN: "310100","US PLATEAU"
+COPN: "310100","US PLATEAU"
+COPN: "310100","US PLATEAU"
+COPN: "310120","Sprint"
+COPN: "310120","Sprint"
+COPN: "310140","GTA Mpulse"
+COPN: "310140","GTA Mpulse"
+COPN: "310140","GTA Mpulse"
+COPN: "310150","AT&T"
+COPN: "310150","AT&T"
+COPN: "310150","AT&T"
+COPN: "310150","AT&T"
+COPN: "310160","T-Mobile"
+COPN: "310160","T-Mobile"
+COPN: "310170","AT&T"
+COPN: "310170","AT&T"
+COPN: "310170","AT&T"
+COPN: "310170","AT&T"
+COPN: "310180","West Central Wireless"
+COPN: "310180","West Central Wireless"
+COPN: "310190","USA Dutch Harbor"
+COPN: "310190","Alaska Wireless"
+COPN: "310200","T-Mobile"
+COPN: "310200","T-Mobile"
+COPN: "310210","T-Mobile"
+COPN: "310210","T-Mobile"

+COPN: "310220","T-Mobile"
+COPN: "310220","T-Mobile"
+COPN: "310230","T-Mobile"
+COPN: "310230","T-Mobile"
+COPN: "310240","T-Mobile"
+COPN: "310240","T-Mobile"
+COPN: "310250","T-Mobile"
+COPN: "310250","T-Mobile"
+COPN: "310260","T-Mobile"
+COPN: "310260","T-Mobile"
+COPN: "310270","T-Mobile"
+COPN: "310270","T-Mobile"
+COPN: "310290","NEP Wireless"
+COPN: "310300","BigSkyUS"
+COPN: "310310","T-Mobile"
+COPN: "310310","T-Mobile"
+COPN: "310320","USA - CellularOne"
+COPN: "310320","USA - CellularOne"
+COPN: "310330","Cellular One"
+COPN: "310330","Cellular One"
+COPN: "310340","WestLink Comm"
+COPN: "310340","WestLink"
+COPN: "310350","Carolina Phone"
+COPN: "310350","Carolina Phone"
+COPN: "310380","AT&T"
+COPN: "310380","AT&T"
+COPN: "310380","AT&T"
+COPN: "310380","AT&T"
+COPN: "310390","Cell1ET"
+COPN: "310390","Cell1ET"
+COPN: "310400","USA iCAN"
+COPN: "310410","AT&T"
+COPN: "310410","AT&T"
+COPN: "310410","AT&T"
+COPN: "310410","AT&T"
+COPN: "310420","Cincinnati Bell Wireless"
+COPN: "310420","Cincinnati Bell Wireless"
+COPN: "310450","Cell One of NE Colorado"
+COPN: "310450","Viaero Wireless"
+COPN: "310460","USA1L"
+COPN: "310470","USA DOCOMO PACIFIC"
+COPN: "310470","USA DOCOMO PACIFIC"
+COPN: "310490","T-Mobile"
+COPN: "310490","T-Mobile"
+COPN: "310530","West Virginia Wireless"
+COPN: "310530","Iowa Wireless USA"
+COPN: "310560","AT&T"
+COPN: "310560","AT&T"
+COPN: "310560","AT&T"
+COPN: "310560","AT&T"
+COPN: "310570","Chinook"
+COPN: "310580","T-Mobile"
+COPN: "310580","T-Mobile"
+COPN: "310590","Verizon"
+COPN: "310590","Verizon"
+COPN: "310610","EpicTouch"
+COPN: "310610","EpicTouch"
+COPN: "310630","AmeriLink PCS"
+COPN: "310630","USA AmeriLink"
+COPN: "310640","Einstein PCS"
+COPN: "310640","USA AE Airadigm"
+COPN: "310650","Jasper"
+COPN: "310660","T-Mobile"
+COPN: "310660","T-Mobile"
+COPN: "310670","Wireless 2000 PCS"
+COPN: "310670","Wireless 2000 PCS"
+COPN: "310680","AT&T"
+COPN: "310680","AT&T"
+COPN: "310680","AT&T"
+COPN: "310680","AT&T"
+COPN: "310690","Immix Wireless"
+COPN: "310690","USA - Immix Wireless"
+COPN: "310700","USABIGFOOT"
+COPN: "310710","USA ASTAC"
+COPN: "310730","U.S.Cellular"
+COPN: "310740","USA OTZ"
+COPN: "310740","Telemetrix"
+COPN: "310760","PTSI"
+COPN: "310760","PTSI"
+COPN: "310770","i wireless"
+COPN: "310770","Iowa Wireless USA"

+COPN: "310780","AirLink PCS"
+COPN: "310780","AirLink PCS"
+COPN: "310790","Pinpoint"
+COPN: "310790","Pinpoint"
+COPN: "310800","T-Mobile"
+COPN: "310800","T-Mobile"
+COPN: "310840","telna Mobile"
+COPN: "310870","US"
+COPN: "310880","USAACSI"
+COPN: "310890","Verizon"
+COPN: "310900","Texas Cellular"
+COPN: "310950","USA XIT Wireless"
+COPN: "310980","AT&T"
+COPN: "310980","AT&T"
+COPN: "310980","AT&T"
+COPN: "310980","AT&T"
+COPN: "31100","Wilkes USA"
+COPN: "311000","USA Mid-Tex Cellular, Lt"
+COPN: "31101","Wilkes USA"
+COPN: "31105","Wilkes USA"
+COPN: "311030","Indigo"
+COPN: "31130","Indigo"
+COPN: "31140","USA - Commnet"
+COPN: "311040","USA - Commnet"
+COPN: "31170","USAEC"
+COPN: "31173","USA Proximiti"
+COPN: "31180","Pine Cellular"
+COPN: "311090","USASXLP"
+COPN: "311110","High Plains"
+COPN: "311130","Cell One Amarillo"
+COPN: "311140","Sprocket"
+COPN: "311140","Sprocket"
+COPN: "311150","AT&T"
+COPN: "311150","AT&T"
+COPN: "311150","AT&T"
+COPN: "311170","PetroCom"
+COPN: "311170","AT&T"
+COPN: "311170","AT&T"
+COPN: "311170","AT&T"
+COPN: "311180","USAWC"
+COPN: "311180","USAWC"
+COPN: "311180","AT&T"
+COPN: "311180","AT&T"
+COPN: "311190","USAC1ECI"
+COPN: "311210","FARMERS"
+COPN: "311210","FARMERS"
+COPN: "311240","USACWCI"
+COPN: "311250","USA i CAN"
+COPN: "311260","SLO Cellular"
+COPN: "311310","Lamar Cellular"
+COPN: "311311","USANCW"
+COPN: "311330","BTW"
+COPN: "311360","Stelera Wireless"
+COPN: "311370","GCI"
+COPN: "311380","AT&T"
+COPN: "311380","AT&T"
+COPN: "311380","AT&T"
+COPN: "311410","AT&T"
+COPN: "311410","AT&T"
+COPN: "311410","AT&T"
+COPN: "311480","Verizon"
+COPN: "311500","MOSAIC"
+COPN: "311530","USANW"
+COPN: "311540","USA Proximiti"
+COPN: "311680","GreenFly"
+COPN: "311710","Northeast Wireless"
+COPN: "311720","MainePCS"
+COPN: "311730","USA Proximiti"
+COPN: "311940","CLEAR"
+COPN: "312060","CoverageCo"
+COPN: "330110","PR Claro"
+COPN: "330110","PR Claro"
+COPN: "33211","Blue Sky"
+COPN: "332011","Blue Sky"
+COPN: "334003","movistar"
+COPN: "33409","Nextel 3G"
+COPN: "334009","Nextel 3G"
+COPN: "33420","TELCEL"
+COPN: "33420","TELCEL"
+COPN: "334020","TELCEL"
+COPN: "33450","Iusacell GSM"
+COPN: "334050","Iusacell GSM"
+COPN: "33450","Iusacell 4G"
+COPN: "33850","DIGICEL"
+COPN: "33850","DIGICEL"
+COPN: "33870","CLARO"
+COPN: "338180","LIME"
+COPN: "34001","F-Orange"
+COPN: "34002","ONLY"

+COPN: "34002","ONLY"
+COPN: "34003","CHIPPIE"
+COPN: "34003","CHIPPIE"
+COPN: "34008","DAUPHIN"
+COPN: "34008","DAUPHIN"
+COPN: "34020","DIGICEL"
+COPN: "342600","LIME"
+COPN: "342750","DIGICEL"
+COPN: "342750","DIGICEL"
+COPN: "342750","DIGICEL"
+COPN: "34430","APUA PCS ANTIGUA"
+COPN: "344030","APUA imobile"
+COPN: "344920","LIME"
+COPN: "344930","Cingular"
+COPN: "346140","LIME"
+COPN: "346140","LIME"
+COPN: "348170","LIME"
+COPN: "348570","CCT Boatphone"
+COPN: "348570","CCT Boatphone"
+COPN: "350000","CELLONE"
+COPN: "35000","CELLONE"
+COPN: "35002","M3 WIRELESS"
+COPN: "350002","BTC MOBILITY LTD."
+COPN: "350010","Cingular"
+COPN: "35230","DIGICEL"
+COPN: "35230","DIGICEL"
+COPN: "352110","LIME"
+COPN: "354860","LIME"
+COPN: "356110","LIME"
+COPN: "356110","LIME"
+COPN: "35850","DIGICEL"
+COPN: "35850","DIGICEL"
+COPN: "358050","DIGICEL"
+COPN: "358110","LIME"
+COPN: "36070","DIGICEL"
+COPN: "36070","DIGICEL"
+COPN: "360110","LIME"
+COPN: "36251","Telcell GSM"
+COPN: "36269","ANT CURACAO TELECOM GSM"
+COPN: "36269","ANT CURACAO TELECOM GSM"
+COPN: "36291","ANT"
+COPN: "36301","SETAR GSM"
+COPN: "363001","SETAR GSM"
+COPN: "36320","DIGICEL"
+COPN: "36320","DIGICEL"
+COPN: "36439","BaTelCell"
+COPN: "364039","BaTelCell"
+COPN: "365840","LIME"
+COPN: "36620","Cingular"
+COPN: "366020","Cingular"
+COPN: "366110","LIME"
+COPN: "36801","CUBACEL"
+COPN: "37001","orange"
+COPN: "37001","orange"
+COPN: "370001","orange"
+COPN: "37002","CLARO DOM"
+COPN: "370002","CLARO DOM"
+COPN: "37002","CLARO DOM"
+COPN: "370004","VIVA"
+COPN: "37201","COMCEL"
+COPN: "37203","Natcom"
+COPN: "37412","TSTT"
+COPN: "376350","LIME"
+COPN: "376350","IslandCom TCI"
+COPN: "376350","IslandCom TCI"
+COPN: "376352","IslandCom TCI"
+COPN: "376360","IslandCom TCI"
+COPN: "40001","AZE - AZERCELL GSM"
+COPN: "40001","AZE - AZERCELL GSM"
+COPN: "40002","BAKCELL- AZ"
+COPN: "40002","BAKCELL - AZ"
+COPN: "40002","BAKCELL - AZ"
+COPN: "40004","AZ Nar"
+COPN: "40004","AZ Nar"
+COPN: "40101","Beeline KZ"
+COPN: "40102","KZ KCELL"
+COPN: "40177","NEO-KZ"
+COPN: "40211","BT B-Mobile"
+COPN: "40277","TASHICELL"
+COPN: "40277","TASHICELL"
+COPN: "40401","Vodafone IN"
+COPN: "40402","IND airtel"
+COPN: "40403","IND airtel"
+COPN: "40404","IDEA"

+COPN: "40405","Vodafone IN"
+COPN: "40407","IDEA"
+COPN: "40409","Reliance"
+COPN: "40410","IND airtel"
+COPN: "40411","Vodafone IN"
+COPN: "40411","Vodafone IN"
+COPN: "40412","IDEA"
+COPN: "40413","Vodafone IN"
+COPN: "40414","IDEA"
+COPN: "40415","Vodafone IN"
+COPN: "40416","IND airtel"
+COPN: "40417","AIRCEL"
+COPN: "40417","AIRCEL"
+COPN: "40418","Reliance"
+COPN: "40419","IDEA"
+COPN: "40420","Vodafone IN"
+COPN: "40420","Vodafone IN"
+COPN: "40421","BPL MOBILE"
+COPN: "40422","IDEA"
+COPN: "40424","IDEA"
+COPN: "40425","AIRCEL"
+COPN: "40425","AIRCEL"
+COPN: "40427","Vodafone IN"
+COPN: "40428","AIRCEL"
+COPN: "40428","AIRCEL"
+COPN: "40429","AIRCEL"
+COPN: "40429","AIRCEL"
+COPN: "40430","Vodafone IN"
+COPN: "40431","IND airtel"
+COPN: "40433","AIRCEL"
+COPN: "40433","AIRCEL"
+COPN: "40434","CellOne"
+COPN: "40435","AIRCEL"
+COPN: "40435","AIRCEL"
+COPN: "40436","Reliance"
+COPN: "40437","AIRCEL"
+COPN: "40437","AIRCEL"
+COPN: "40438","CellOne"
+COPN: "40440","IND airtel"
+COPN: "40441","AIRCEL"
+COPN: "40442","AIRCEL"
+COPN: "40443","Vodafone IN"
+COPN: "40444","IDEA"
+COPN: "40445","IND airtel"
+COPN: "40446","Vodafone IN"
+COPN: "40449","IND airtel"
+COPN: "40450","Reliance"
+COPN: "40451","CellOne"
+COPN: "40452","Reliance"
+COPN: "40453","CellOne"
+COPN: "40454","CellOne"
+COPN: "40455","CellOne"
+COPN: "40456","IDEA"
+COPN: "40457","CellOne"
+COPN: "40458","CellOne"
+COPN: "40459","CellOne"
+COPN: "40460","Vodafone IN"
+COPN: "40462","CellOne"
+COPN: "40464","CellOne"
+COPN: "40466","CellOne"
+COPN: "40467","Reliance"
+COPN: "40468","IN-DOLPHIN"
+COPN: "40469","IN-DOLPHIN"
+COPN: "40470","IND airtel"
+COPN: "40471","CellOne"
+COPN: "40472","CellOne"
+COPN: "40473","CellOne"
+COPN: "40474","CellOne"
+COPN: "40475","CellOne"
+COPN: "40476","CellOne"
+COPN: "40477","CellOne"
+COPN: "40478","IDEA"
+COPN: "40479","CellOne"
+COPN: "40480","CellOne"
+COPN: "40481","CellOne"
+COPN: "40482","IDEA"
+COPN: "40483","Reliance"
+COPN: "40484","Vodafone IN"
+COPN: "40485","Reliance"
+COPN: "40486","Vodafone IN"
+COPN: "40487","IDEA"
+COPN: "40488","Vodafone IN"
+COPN: "40489","IDEA"
+COPN: "40490","IND airtel"
+COPN: "40491","Aircel"
+COPN: "40492","IND airtel"
+COPN: "40493","IND airtel"
+COPN: "40494","IND airtel"
+COPN: "40495","IND airtel"
+COPN: "40496","IND airtel"

+COPN: "40497","IND airtel"
+COPN: "40498","IND airtel"
+COPN: "40500","TATA DOCOMO"
+COPN: "40501","Reliance"
+COPN: "40503","TATA DOCOMO"
+COPN: "40505","Reliance"
+COPN: "40506","Reliance"
+COPN: "40507","Reliance"
+COPN: "40509","Reliance"
+COPN: "40510","Reliance"
+COPN: "40511","Reliance"
+COPN: "40513","Reliance"
+COPN: "40515","Reliance"
+COPN: "40518","Reliance"
+COPN: "40519","Reliance"
+COPN: "40520","Reliance"
+COPN: "40521","Reliance"
+COPN: "40522","Reliance"
+COPN: "40525","TATA DOCOMO"
+COPN: "40527","TATA DOCOMO"
+COPN: "40529","TATA DOCOMO"
+COPN: "40530","TATA DOCOMO"
+COPN: "40531","TATA DOCOMO"
+COPN: "40532","TATA DOCOMO"
+COPN: "40534","TATA DOCOMO"
+COPN: "40535","TATA DOCOMO"
+COPN: "40536","TATA DOCOMO"
+COPN: "40537","TATA DOCOMO"
+COPN: "40538","TATA DOCOMO"
+COPN: "40539","TATA DOCOMO"
+COPN: "40541","TATA DOCOMO"
+COPN: "40542","TATA DOCOMO"
+COPN: "40543","TATA DOCOMO"
+COPN: "40544","TATA DOCOMO"
+COPN: "40545","TATA DOCOMO"
+COPN: "40546","TATA DOCOMO"
+COPN: "40547","TATA DOCOMO"
+COPN: "40550","Reliance"
+COPN: "40551","IND airtel"
+COPN: "40552","IND airtel"
+COPN: "40553","IND airtel"
+COPN: "40554","IND airtel"
+COPN: "40555","IND airtel"
+COPN: "40556","IND airtel"
+COPN: "40566","Vodafone IN"
+COPN: "40567","Vodafone IN"
+COPN: "40570","IDEA"
+COPN: "405750","Vodafone IN"
+COPN: "405751","Vodafone IN"
+COPN: "405752","Vodafone IN"
+COPN: "405753","Vodafone IN"
+COPN: "405754","Vodafone IN"
+COPN: "405755","Vodafone IN"
+COPN: "405756","Vodafone IN"
+COPN: "405799","IDEA"
+COPN: "405800","Aircel"
+COPN: "405801","Aircel"
+COPN: "405802","Aircel"
+COPN: "405803","Aircel"
+COPN: "405804","Aircel"
+COPN: "405805","Aircel"
+COPN: "405806","Aircel"
+COPN: "405807","Aircel"
+COPN: "405808","Aircel"
+COPN: "405809","Aircel"
+COPN: "405810","Aircel"
+COPN: "405811","Aircel"
+COPN: "405812","Aircel"
+COPN: "405813","IN UNITECH"
+COPN: "405814","IN UNITECH"
+COPN: "405815","IN UNITECH"
+COPN: "405816","IN UNITECH"
+COPN: "405817","IN UNITECH"
+COPN: "405818","IN UNITECH"
+COPN: "405819","IN UNITECH"
+COPN: "405820","IN UNITECH"
+COPN: "405821","IN UNITECH"
+COPN: "405822","IN UNITECH"
+COPN: "405823","VIDEOCON"
+COPN: "405824","VIDEOCON"
+COPN: "405825","VIDEOCON"
+COPN: "405827","VIDEOCON"
+COPN: "405828","VIDEOCON"
+COPN: "405829","VIDEOCON"
+COPN: "405830","VIDEOCON"
+COPN: "405831","VIDEOCON"
+COPN: "405832","VIDEOCON"
+COPN: "405833","VIDEOCON"
+COPN: "405834","VIDEOCON"
+COPN: "405835","VIDEOCON"

+COPN: "405836","VIDEOCON"
+COPN: "405837","VIDEOCON"
+COPN: "405838","VIDEOCON"
+COPN: "405839","VIDEOCON"
+COPN: "405840","VIDEOCON"
+COPN: "405841","VIDEOCON"
+COPN: "405842","VIDEOCON"
+COPN: "405843","VIDEOCON"
+COPN: "405844","IN UNITECH"
+COPN: "405845","IDEA"
+COPN: "405846","IDEA"
+COPN: "405848","IDEA"
+COPN: "405849","IDEA"
+COPN: "405850","IDEA"
+COPN: "405852","IDEA"
+COPN: "405853","IDEA"
+COPN: "405854","IN Loop"
+COPN: "405855","IN Loop"
+COPN: "405856","IN Loop"
+COPN: "405857","IN Loop"
+COPN: "405858","IN Loop"
+COPN: "405859","IN Loop"
+COPN: "405860","IN Loop"
+COPN: "405861","IN Loop"
+COPN: "405862","IN Loop"
+COPN: "405863","IN Loop"
+COPN: "405864","IN Loop"
+COPN: "405865","IN Loop"
+COPN: "405866","IN Loop"
+COPN: "405867","IN Loop"
+COPN: "405868","IN Loop"
+COPN: "405869","IN Loop"
+COPN: "405870","IN Loop"
+COPN: "405871","IN Loop"
+COPN: "405872","IN Loop"
+COPN: "405873","IN Loop"
+COPN: "405874","IN Loop"
+COPN: "405875","IN UNITECH"
+COPN: "405876","IN UNITECH"
+COPN: "405877","IN UNITECH"
+COPN: "405878","IN UNITECH"
+COPN: "405879","IN UNITECH"
+COPN: "405880","IN UNITECH"
+COPN: "405912","etisalat"
+COPN: "405913","etisalat"
+COPN: "405914","etisalat"
+COPN: "405915","etisalat"
+COPN: "405916","etisalat"
+COPN: "405917","etisalat"
+COPN: "405918","etisalat"
+COPN: "405919","etisalat"
+COPN: "405920","etisalat"
+COPN: "405921","etisalat"
+COPN: "405922","etisalat"
+COPN: "405923","etisalat"
+COPN: "405924","etisalat"
+COPN: "405925","IN UNITECH"
+COPN: "405926","IN UNITECH"
+COPN: "405927","IN UNITECH"
+COPN: "405928","IN UNITECH"
+COPN: "405929","IN UNITECH"
+COPN: "41001","Mobilink"
+COPN: "41003","PK-UFONE"
+COPN: "41004","ZONG"
+COPN: "41004","ZONG"
+COPN: "41006","Telenor PK"
+COPN: "41006","Telenor PK"
+COPN: "41007","WaridTel"
+COPN: "41007","WaridTel"
+COPN: "41201","AF AWCC"
+COPN: "41201","AF AWCC"
+COPN: "41220","ROSHAN"
+COPN: "41240","MTN AF"
+COPN: "41240","MTN AF"
+COPN: "41250","Etisalat Af"
+COPN: "41250","Etisalat Af"
+COPN: "41301","Mobitel"
+COPN: "41302","SRI DIALOG"
+COPN: "41302","SRI DIALOG"
+COPN: "41303","SRI Etisalat"
+COPN: "41303","SRI Etisalat"
+COPN: "41305","SRI airtel"
+COPN: "41305","SRI AIRTEL"
+COPN: "41305","SRI AIRTEL"
+COPN: "41308","Hutch"
+COPN: "41308","Hutch"
+COPN: "41401","MM 900"
+COPN: "41501","alfa"
+COPN: "41503","RL MTC Lebanon"

+COPN: "41505","LBN OGERO Mobile"
+COPN: "41601","zain JO"
+COPN: "41603","UMNIAH"
+COPN: "41677","Orange JO"
+COPN: "41701","SYRIATEL"
+COPN: "41701","SYRIATEL"
+COPN: "41702","MTN"
+COPN: "41702","MTN"
+COPN: "41709","SYR MOBILE SYR"
+COPN: "41800","ASIACELL"
+COPN: "41805","ASIACELL"
+COPN: "41820","zain IQ"
+COPN: "41830","IRAQNA"
+COPN: "41840","KOREK"
+COPN: "41902","Zain KW"
+COPN: "41902","Zain KW"
+COPN: "41902","Zain KW"
+COPN: "41903","KT WATANIYA"
+COPN: "41903","KT WATANIYA"
+COPN: "41904","KT, VIVA"
+COPN: "41904","KT, VIVA"
+COPN: "41904","KT, VIVA"
+COPN: "42001","STC"
+COPN: "42001","STC"
+COPN: "42003","Mobily-KSA"
+COPN: "42003","Mobily-KSA"
+COPN: "42003","Mobily-KSA"
+COPN: "42004","Zain SA"
+COPN: "42004","Zain SA"
+COPN: "42004","Zain SA"
+COPN: "42101","SabaFon"
+COPN: "42102","MTN"
+COPN: "421700","YemYY"
+COPN: "42202","OMAN MOBILE"
+COPN: "42203","nawras"
+COPN: "42402","ETISALAT"
+COPN: "42402","ETISALAT"
+COPN: "42403","du"
+COPN: "42403","du"
+COPN: "42403","du"
+COPN: "42501","Orange IL"
+COPN: "42501","Orange IL"
+COPN: "42501","Orange IL"
+COPN: "42502","CH"
+COPN: "42502","IL Cellcom"
+COPN: "42503","IL Pelephone"
+COPN: "42505","JAWWAL-PALESTINE"
+COPN: "42506","PS, Wataniya Mobile"
+COPN: "42506","PS, Wataniya Mobile"
+COPN: "42507","Hot Mobile"
+COPN: "42601","BATELCO"
+COPN: "42602","Zain BH"
+COPN: "42602","Zain BH"
+COPN: "42602","Zain BH"
+COPN: "42604","VIVA BH"
+COPN: "42604","VIVA BH"
+COPN: "42701","Qat - Qtel"
+COPN: "42701","Qat - Qtel"
+COPN: "42702","vodafone"
+COPN: "42702","vodafone"
+COPN: "42702","vodafone"
+COPN: "42806","GMOBILE_MN"
+COPN: "42888","MONGOLIA UNITEL LLC"
+COPN: "42888","MONGOLIA UNITEL LLC"
+COPN: "42888","MONGOLIA UNITEL LLC"
+COPN: "42899","MN MobiCom"
+COPN: "42902","Ncell"
+COPN: "42902","Ncell"
+COPN: "43211","IR-TCI"
+COPN: "43211","IR-TCI"
+COPN: "43214","IR KISH"
+COPN: "43219","IR MTCE"
+COPN: "43220","IRN 20"
+COPN: "43232","Iran Taliya"
+COPN: "43235","MTN Irancell"
+COPN: "43235","MTN Irancell"
+COPN: "43404","Beeline UZ"
+COPN: "43404","Beeline UZ"
+COPN: "43404","Beeline UZ"
+COPN: "43405","UZB Ucell"
+COPN: "43405","UZB Ucell"
+COPN: "43405","UZB Ucell"
+COPN: "43407","UZB MTS"

+COPN: "43407","UZB MTS"
+COPN: "43407","UZB MTS"
+COPN: "43601","TCELL"
+COPN: "43601","TCELL"
+COPN: "43601","TCELL"
+COPN: "43602","TCELL"
+COPN: "43602","TCELL"
+COPN: "43602","TCELL"
+COPN: "43603","MegaFon TJK"
+COPN: "43603","MegaFon TJK"
+COPN: "43603","MegaFon TJK"
+COPN: "43604","Babilon-M"
+COPN: "43605","BEELINE TJ"
+COPN: "43605","BEELINE TJ"
+COPN: "43612","INDIGO-3G"
+COPN: "43701","Beeline KG"
+COPN: "43701","Beeline KG"
+COPN: "43701","Beeline KG"
+COPN: "43705","MEGACOM"
+COPN: "43705","MEGACOM"
+COPN: "43709","O!"
+COPN: "43709","O!"
+COPN: "43801","MTS TM"
+COPN: "43801","MTS TM"
+COPN: "43802","TM CELL"
+COPN: "43802","TM CELL"
+COPN: "44000","EMOBILE"
+COPN: "44000","EMOBILE"
+COPN: "44010","JP DOCOMO"
+COPN: "44020","SoftBank"
+COPN: "44050","KDDI"
+COPN: "45002","KT"
+COPN: "45005","KOR SK Telecom"
+COPN: "45006","KOR LG Uplus"
+COPN: "45008","KT"
+COPN: "45201","VN MobiFone"
+COPN: "45201","VN MobiFone"
+COPN: "45201","VN MobiFone"
+COPN: "45202","VN VINAPHONE"
+COPN: "45204","VIETTEL"
+COPN: "45204","VNM and
VIETTEL"
+COPN: "45204","VNM and
VIETTEL"
+COPN: "45205","VN Vietnamobile"
+COPN: "45207","Gmobile"
+COPN: "45400","CSL"
+COPN: "45400","CSL"
+COPN: "45402","CSL"
+COPN: "45402","CSL"
+COPN: "45402","CSL"
+COPN: "45403","3 HK"
+COPN: "45404","3(2G)"
+COPN: "45404","3(2G)"
+COPN: "45406","SmarTone HK"
+COPN: "45406","SmarTone HK"
+COPN: "45406","SmarTone HK"
+COPN: "45410","CSL"
+COPN: "45412","China Mobile HK"
+COPN: "45415","SmarTone HK"
+COPN: "45415","SmarTone HK"
+COPN: "45415","SmarTone HK"
+COPN: "45416","PCCW"
+COPN: "45417","SmarTone HK"
+COPN: "45417","SmarTone HK"
+COPN: "45417","SmarTone HK"
+COPN: "45418","CSL"
+COPN: "45418","CSL"
+COPN: "45418","CSL"
+COPN: "45419","PCCW"
+COPN: "45500","SmarTone MAC"
+COPN: "45500","SmarTone MAC"
+COPN: "45501","CTM"
+COPN: "45501","CTM"
+COPN: "45503","3 Macau"
+COPN: "45503","3 Macau"
+COPN: "45503","3 Macau"
+COPN: "45504","CTM"
+COPN: "45505","3 Macau"
+COPN: "45601","MOBITEL - KHM"
+COPN: "45602","Hello Axiata"
+COPN: "45602","Hello Axiata"
+COPN: "45602","Hello Axiata"
+COPN: "45604","CADCOMMS"
+COPN: "45604","CADCOMMS"
+COPN: "45605","SMART"
+COPN: "45605","SMART"
+COPN: "45606","SMART"
+COPN: "45606","SMART"
+COPN: "45606","SMART"

+COPN: "45608","Metfone"
+COPN: "45608","Metfone"
+COPN: "45609","Beeline KH"
+COPN: "45609","Beeline KH"
+COPN: "45618","Mfone"
+COPN: "45618","Mfone"
+COPN: "45701","LAO GSM"
+COPN: "45702","ETL MOBILE NETWORK"
+COPN: "45703","Unitel"
+COPN: "45703","Unitel"
+COPN: "45708","TIGO LAO"
+COPN: "45708","TIGO LAO"
+COPN: "46000","CHINA MOBILE"
+COPN: "46001","CHN-UNICOM"
+COPN: "46001","CHN-UNICOM"
+COPN: "46011","CHN-CT"
+COPN: "46601","Far EasTone"
+COPN: "46601","Far EasTone"
+COPN: "46668","ACeS"
+COPN: "46688","KGT-Online"
+COPN: "46689","VIBO"
+COPN: "46692","Chunghwa Telecom"
+COPN: "46692","Chunghwa Telecom"
+COPN: "46692","Chunghwa Telecom"
+COPN: "46693","TWN MOBITAI"
+COPN: "46697","TW Mobile"
+COPN: "46699","TWN TransAsia Telecom GS"
+COPN: "46703","KP SUN"
+COPN: "46705","Koryolink"
+COPN: "47001","Grameenphone"
+COPN: "47001","Grameenphone"
+COPN: "47002","BGD ROBI AXIATA "
+COPN: "47003","Banglalink"
+COPN: "47004","BGD bMobile"
+COPN: "47007","airtel"
+COPN: "47201","MV DHIMOBILE"
+COPN: "47202","WATANIYA"
+COPN: "47202","WATANIYA"
+COPN: "50200","TIME3G"
+COPN: "50201","TIME3G"
+COPN: "50212","MY MAXIS"
+COPN: "50212","MY MAXIS"
+COPN: "50213","MY CELCOM 3G"
+COPN: "50216","DiGi"
+COPN: "50218","U MOBILE"
+COPN: "50219","MY CELCOM"
+COPN: "50501","Telstra Mobile"
+COPN: "50501","Telstra Mobile"
+COPN: "50502","YES OPTUS"
+COPN: "50502","YES OPTUS"
+COPN: "50503","vodafone AU"
+COPN: "50503","vodafone AU"
+COPN: "50506","3TELSTRA"
+COPN: "50510","Norfolk Telecom"
+COPN: "51000","ACeS"
+COPN: "51001","IND INDOSAT"
+COPN: "51001","IND INDOSAT"
+COPN: "51008","AXIS"
+COPN: "51008","AXIS"
+COPN: "51010","IND TELKOMSEL"
+COPN: "51010","IND TELKOMSEL"
+COPN: "51011","IND XL"
+COPN: "51011","IND XL"
+COPN: "51021","IND INDOSAT"
+COPN: "51089","3"
+COPN: "51089","3"
+COPN: "51402","TLS-TT"
+COPN: "51501","ISLACOM"
+COPN: "51502","Globe Telecom-PH"
+COPN: "51502","Globe Telecom-PH"
+COPN: "51503","SMART"
+COPN: "51503","SMART"
+COPN: "51505","PH Sun Cellular"
+COPN: "51505","PH Sun Cellular"
+COPN: "51511","ACeS"
+COPN: "51518","CURE"
+COPN: "52000","TH 3G+"
+COPN: "52001","TH GSM"
+COPN: "52015","TOT3G"
+COPN: "520015","TH ACT 1900"
+COPN: "52018","TH-DTAC"
+COPN: "52020","ACeS"
+COPN: "52023","TH GSM 1800"
+COPN: "52099","TRUE"
+COPN: "52501","SingTel"
+COPN: "52501","SingTel"
+COPN: "52501","4G SingTel"

+COPN: "52502","SingTel-G18"
+COPN: "52503","SGP - M1"
+COPN: "52503","SGP - M1"
+COPN: "52503","SGP-M1"
+COPN: "52503","SGP-M1"
+COPN: "52505","STARHUB"
+COPN: "52505","STARHUB"
+COPN: "52507","SGP Call Zone"
+COPN: "52802","b-mobile"
+COPN: "52811","BRU-DSTCom"
+COPN: "53001","vodafone NZ"
+COPN: "53005","Telecom NZ"
+COPN: "53005","Telecom NZ"
+COPN: "53024","2degrees"
+COPN: "53024","2degrees"
+COPN: "53701","PNGBMobile"
+COPN: "53703","DIGICEL"
+COPN: "53901","U-CALL"
+COPN: "53988","Digicel Tonga"
+COPN: "54001","SI BREEZE"
+COPN: "540002","SIS"
+COPN: "54002","SIS"
+COPN: "54002","SIS"
+COPN: "54101","VUT SMILE"
+COPN: "54105","Digicel"
+COPN: "54105","Digicel"
+COPN: "54201","FJ VODAFONE"
+COPN: "54202","DIGICEL"
+COPN: "54202","DIGICEL"
+COPN: "544110","Bluesky
Communications"
+COPN: "544110","Bluesky
Communications"
+COPN: "544110","Bluesky
Communications"
+COPN: "544544","Blue Sky
Communications"
+COPN: "54509","KL-Frigate"
+COPN: "54601","NCL MOBILIS"
+COPN: "54720","F-VINI"
+COPN: "54801","CK KOKANET"
+COPN: "54900","DIGICEL"
+COPN: "54927","Bluesky"
+COPN: "54927","Bluesky"
+COPN: "55001","FSM Telecom"
+COPN: "55201","PalauCel"
+COPN: "55280","PLWPMC"
+COPN: "55280","PLWPMC"
+COPN: "60201","EGY MobiNiL"
+COPN: "60202","vodafone EG"
+COPN: "60203","etisalat"
+COPN: "60203","etisalat"
+COPN: "60203","etisalat"
+COPN: "60301","ALG Mobilis"
+COPN: "60301","ALG Mobilis"
+COPN: "60302","Djezzy"
+COPN: "60302","Djezzy"
+COPN: "60303","DZA NEDJMA"
+COPN: "60303","DZA NEDJMA"
+COPN: "60400","MOR MEDITEL"
+COPN: "60401","MOR IAM"
+COPN: "60401","MOR IAM"
+COPN: "60401","MOR IAM"
+COPN: "60501","Orange"
+COPN: "60501","Orange"
+COPN: "60502","TUNISIE
TELECOM"
+COPN: "60503","TUNISIANA"
+COPN: "60600","Libyana"
+COPN: "60601","Al Madar"
+COPN: "60691","Almadar"
+COPN: "606218","Almadar"
+COPN: "60701","GAMCEL"
+COPN: "60702","AFRICELL"
+COPN: "60702","AFRICELL"
+COPN: "60703","GM COMIUM"
+COPN: "60703","GM COMIUM"
+COPN: "60704","Qcell"
+COPN: "60704","Qcell"
+COPN: "60801","SN ALIZE"
+COPN: "60802","SN-SENTEL SG"
+COPN: "60803","SEN expresso"
+COPN: "60803","SEN expresso"
+COPN: "60901","MR MATTEL"
+COPN: "60902","MR Expresso"
+COPN: "60902","MR Expresso"
+COPN: "60902","MR Expresso"
+COPN: "60910","MAURITEL"
+COPN: "61001","MALITEL ML"
+COPN: "61002","ORANGE ML"

+COPN: "61101","Orange GN"
+COPN: "61101","Orange GN"
+COPN: "61102","GN LAGUI"
+COPN: "61104","GNMTN"
+COPN: "61104","GNMTN"
+COPN: "61105","GINCL"
+COPN: "61105","GINCL"
+COPN: "61105","GINCL"
+COPN: "61202","ETISALAT CI"
+COPN: "61202","ETISALAT CI"
+COPN: "61203","Orange CI"
+COPN: "61204","KoZ"
+COPN: "61204","KoZ"
+COPN: "61205","MTN CI"
+COPN: "61302","BF Celtel"
+COPN: "61402","CELTEL"
+COPN: "61403","ETISALAT NER"
+COPN: "61404","Orange NE"
+COPN: "61404","Orange NE"
+COPN: "61501","TG-TOGO CELL"
+COPN: "61503","ETISALAT TOGO"
+COPN: "61602","ETISALAT BENIN"
+COPN: "61603","MTN BENIN"
+COPN: "61603","MTN BENIN"
+COPN: "61604","BELL BENIN COMMUNICATION"
+COPN: "61604","BELL BENIN COMMUNICATION"
+COPN: "61605","GloBenin"
+COPN: "61605","GloBenin"
+COPN: "61701","CELLPLUS-MRU"
+COPN: "61703","MTML"
+COPN: "61710","EMTEL-MRU"
+COPN: "61801","LBR Lonestar Cell"
+COPN: "61807","Celcom GSM"
+COPN: "61807","Celcom GSM"
+COPN: "61901","CELTEL SL"
+COPN: "61902","MILLICOM SL"
+COPN: "61907","GreenN SL"
+COPN: "61907","GreenN SL"
+COPN: "62001","GH MTN"
+COPN: "62002","GH Vodafone"
+COPN: "62003","tiGO"
+COPN: "62006","Airtel GH"
+COPN: "62006","Airtel GH"
+COPN: "620620","Glo Ghana"
+COPN: "620620","Glo Ghana"
+COPN: "62120","AirtelNG"
+COPN: "62120","AirtelNG"
+COPN: "62130","MTN - NG"
+COPN: "62130","MTN - NG"
+COPN: "62140","NG Mtel"
+COPN: "62140","NG Mtel"
+COPN: "62150","Glo NG"
+COPN: "62150","Glo NG"
+COPN: "62160","EMTS NGA"
+COPN: "62160","EMTS NGA"
+COPN: "62201","CELTEL TCD"
+COPN: "62301","ETISALAT RCA"
+COPN: "62302","Telecel"
+COPN: "62304","NationLink"
+COPN: "62401","MTN CAM"
+COPN: "62402","Orange CAM"
+COPN: "62501","CPV MOVEL"
+COPN: "62501","CPV MOVEL"
+COPN: "62502","CPV T+"
+COPN: "62502","CPV T+"
+COPN: "62601","STP CSTmovel"
+COPN: "62701","GNQ01"
+COPN: "62703","HiTs-GQ"
+COPN: "62703","HiTs-GQ"
+COPN: "62801","LIBERTIS"
+COPN: "62802","ETISALAT GAB"
+COPN: "62803","ZAIN GA"
+COPN: "62901","CELTEL"
+COPN: "62902","AZUR COG"
+COPN: "62902","AZUR COG"
+COPN: "63005","SCELL CD"
+COPN: "63005","SCELL CD"
+COPN: "62907","WARID RC"
+COPN: "62907","WARID RC"
+COPN: "62910","COG MTN"
+COPN: "63001","VODACOM CD"
+COPN: "63001","VODACOM CD"
+COPN: "63002","CELTEL DRC"
+COPN: "63005","SCELL CD"
+COPN: "63005","SCELL CD"
+COPN: "63086","CD COD"
+COPN: "63086","CD COD"
+COPN: "63088","CD Smart"

+COPN: "63088","CD Smart"
+COPN: "63089","CD OASIS"
+COPN: "63102","UNITEL"
+COPN: "63102","UNITEL"
+COPN: "63202","MTN"
+COPN: "63202","MTN"
+COPN: "63203","Orange BS"
+COPN: "63203","Orange BS"
+COPN: "63207","GTM"
+COPN: "63301","C&W SEY"
+COPN: "63310","SEZ AIRTEL"
+COPN: "63401","Zain SDN"
+COPN: "63401","Zain SDN"
+COPN: "63401","Zain SDN"
+COPN: "63402","MTN"
+COPN: "63402","MTN"
+COPN: "63402","MTN"
+COPN: "63405","Vivacell"
+COPN: "63405","Vivacell"
+COPN: "63406","Zain SD"
+COPN: "63501","RWAAR"
+COPN: "63501","RWAAR"
+COPN: "63504","RWAAR"
+COPN: "63504","RWAAR"
+COPN: "63510","MTN Rwanda"
+COPN: "63512","RWTEL"
+COPN: "63512","RWTEL"
+COPN: "63513","TIGO"
+COPN: "63513","TIGO"
+COPN: "63513","TIGO"
+COPN: "63514","RWAAR"
+COPN: "63514","RWAAR"
+COPN: "63514","RWAAR"
+COPN: "63601","ETH-MTN"
+COPN: "63701","SO Telesom"
+COPN: "63704","SOMAFONE"
+COPN: "63704","SOMAFONE"
+COPN: "63707","SO SALAMGSM"
+COPN: "63730","Som Golis"
+COPN: "63740","SO/MONTYSOM"
+COPN: "63740","SO/MONTYSOM"
+COPN: "63771","SOMTEL"
+COPN: "63771","SOMTEL"
+COPN: "63782","Telsom"
+COPN: "63801","DJ EVATIS"
+COPN: "63902","Safaricom"
+COPN: "63903","Airtel Networks Kenya Lt"
+COPN: "63903","Airtel Networks Kenya Lt"
+COPN: "63907","GSM Telkom"
+COPN: "63907","GSM Telkom"
+COPN: "64002","TIGO - TZ"
+COPN: "64002","TIGO - TZ"
+COPN: "64003","ZANTEL-TZ"
+COPN: "64003","ZANTEL-TZ"
+COPN: "64004","VodaCom"
+COPN: "64004","VodaCom"
+COPN: "64005","celtel"
+COPN: "64005","celtel"
+COPN: "64008","Smart"
+COPN: "64009","Hits TZ"
+COPN: "64101","UG CelTel"
+COPN: "64110","MTN-UGANDA"
+COPN: "64111","Uganda Telecom"
+COPN: "64111","Uganda Telecom"
+COPN: "64114","ORANGE UGANDA"
+COPN: "64114","ORANGE UGANDA"
+COPN: "64122","WaridTel"
+COPN: "64122","WaridTel"
+COPN: "64201","BDI ECONET"
+COPN: "64201","BDI ECONET"
+COPN: "64201","BDI ECONET"
+COPN: "64202","BDI TEMPO-AFRICELL"
+COPN: "64203","ONATEL BDI"
+COPN: "64207","SMART"
+COPN: "64282","TELECEL-BDI"
+COPN: "64301","MOZ - mCel"
+COPN: "64301","MOZ - mCel"
+COPN: "64303","MOVITEL"
+COPN: "64303","MOVITEL"
+COPN: "64303","MOVITEL"
+COPN: "64304","VodaCom-MZ"
+COPN: "64304","VodaCom-MZ"
+COPN: "64501","ZM CELTEL"
+COPN: "64502","MTN ZM"
+COPN: "64601","ZAIN MG"

+COPN: "64601","ZAIN MG"
+COPN: "64602","Orange MG"
+COPN: "64604","TELMA"
+COPN: "64700","Orange re"
+COPN: "64700","Orange re"
+COPN: "64702","ONLY"
+COPN: "64702","ONLY"
+COPN: "64710","SFR REUNION"
+COPN: "64801","ZW NET*ONE"
+COPN: "64803","TELECEL ZW"
+COPN: "64804","ZW ECONET"
+COPN: "64901","MTC NAMIBIA"
+COPN: "64901","MTC NAMIBIA"
+COPN: "64903","Leo"
+COPN: "64903","Leo"
+COPN: "65001","TNM"
+COPN: "65001","TNM"
+COPN: "65010","CELTEL MW"
+COPN: "65101","Vodacom Lesotho"
+COPN: "65102","LS-ETL"
+COPN: "65201","BW MASCOM"
+COPN: "65202","Orange"
+COPN: "65204","BWA04"
+COPN: "65204","BWA04"
+COPN: "65310","Swazi-MTN"
+COPN: "65401","HURI"
+COPN: "65501","VodaCom-SA"
+COPN: "65502","8.ta"
+COPN: "65502","8.ta"
+COPN: "65507","Cell C"
+COPN: "65507","Cell C"
+COPN: "65510","MTN-SA"
+COPN: "65902","MTN"
+COPN: "65902","MTN"
+COPN: "65902","MTN"
+COPN: "65903","Gemtel"
+COPN: "65903","Gemtel"
+COPN: "65904","Vivacell"
+COPN: "65904","Vivacell"
+COPN: "65906","ZAIN SS"
+COPN: "65906","ZAIN SS"
+COPN: "702067","BTL"
+COPN: "70401","CLARO GTM"
+COPN: "704001","CLARO GTM"
+COPN: "70401","Claro GTM"
+COPN: "70402","Comcel_GSM"
+COPN: "704003","movistar"
+COPN: "706001","CLARO SLV"
+COPN: "70602","Digicel"
+COPN: "70603","TELEMOVIL"
+COPN: "70604","movistar"
+COPN: "706004","movistar"
+COPN: "70610","ESV PERSONAL"
+COPN: "706010","ESV PERSONAL"
+COPN: "708001","CLARO HND"
+COPN: "70802","CELTELHND"
+COPN: "708030","HND"
+COPN: "71001","CLARO NIC"
+COPN: "710001","CLARO NIC"
+COPN: "71002","CLARO NIC"
+COPN: "710002","CLARO NIC"
+COPN: "71003","CLARO NIC"
+COPN: "710003","CLARO NIC"
+COPN: "71007","CLARO NIC"
+COPN: "710007","CLARO NIC"
+COPN: "71021","CLARO NIC"
+COPN: "710021","CLARO NIC"
+COPN: "71021","CLARO NIC"
+COPN: "71073","CLARO NIC"
+COPN: "710073","CLARO NIC"
+COPN: "71073","CLARO NIC"
+COPN: "710300","MOVISTARNI"
+COPN: "71200","I.C.E."
+COPN: "71201","I.C.E."
+COPN: "71201","I.C.E."
+COPN: "71202","I.C.E."
+COPN: "71202","I.C.E."
+COPN: "71203","CLARO CR"
+COPN: "71204","Movistar"
+COPN: "71204","Movistar"
+COPN: "71204","Movistar"
+COPN: "71401","+Movil - C&W PAN"
+COPN: "714003","CLARO PA"
+COPN: "714004","DIGICEL"
+COPN: "71420","Movistar"
+COPN: "71606","MOVISTAR"
+COPN: "71610","CLARO PER"
+COPN: "716010","CLARO PER"
+COPN: "716015","Viettel Peru S.A.C."

+COPN: "71617","PERN3"
+COPN: "72207","Movistar"
+COPN: "722007","Movistar"
+COPN: "72234","AR PERSONAL"
+COPN: "722034","AR PERSONAL"
+COPN: "72236","AR PERSONAL"
+COPN: "722036","AR PERSONAL"
+COPN: "72270","Movistar"
+COPN: "722070","Movistar"
+COPN: "722310","CLARO ARGENTINA"
+COPN: "722310","CLARO ARGENTINA"
+COPN: "722341","AR PERSONAL"
+COPN: "722341","AR PERSONAL"
+COPN: "72402","TIM BRASIL"
+COPN: "72402","TIM BRASIL"
+COPN: "72403","TIM BRASIL"
+COPN: "72403","TIM BRASIL"
+COPN: "72404","TIM BRASIL"
+COPN: "72404","TIM BRASIL"
+COPN: "72405","Claro"
+COPN: "72406","VIVO"
+COPN: "724006","VIVO"
+COPN: "72410","VIVO"
+COPN: "724010","VIVO"
+COPN: "72411","VIVO"
+COPN: "724011","VIVO"
+COPN: "72415","BRA SCTL"
+COPN: "72415","BRA SCTL"
+COPN: "72415","BRA SCTL"
+COPN: "72416","Oi"
+COPN: "72423","VIVO"
+COPN: "72423","VIVO"
+COPN: "72423","VIVO"
+COPN: "72424","AMAZONIA"
+COPN: "72424","AMAZONIA"
+COPN: "72431","Oi"
+COPN: "72432","CTBC"
+COPN: "72432","CTBC"
+COPN: "72433","CTBC"
+COPN: "72433","CTBC"
+COPN: "72434","CTBC"
+COPN: "72434","CTBC"
+COPN: "724039","Nextel Brazil 3G"
+COPN: "72439","Nextel Brasil 3G"
+COPN: "73001","CL ENTEL PCS"
+COPN: "730001","CL ENTEL PCS"
+COPN: "73002","Movistar"
+COPN: "73002","Movistar"
+COPN: "730002","Movistar"
+COPN: "730003","CLARO CHL"
+COPN: "73008","CHL VTR"
+COPN: "73009","Nextel3G"
+COPN: "73010","CL ENTEL PCS"
+COPN: "730010","CL ENTEL PCS"
+COPN: "730730","CHL Movistar"
+COPN: "732101","Claro"
+COPN: "732101","Claro"
+COPN: "732101","Claro"
+COPN: "732103","COL MOV / TIGO"
+COPN: "732111","OLA"
+COPN: "732111","COL MOV / TIGO"
+COPN: "732123","Movistar"
+COPN: "732123","Movistar"
+COPN: "73401","DIGITEL GSM"
+COPN: "73402","DIGITEL GSM"
+COPN: "73403","DIGITEL GSM"
+COPN: "73404","movistar"
+COPN: "73406","VE_MOVILNET"
+COPN: "73601","VIVA"
+COPN: "736001","VIVA"
+COPN: "73602","BOMOV"
+COPN: "736002","BOMOV"
+COPN: "73603","Telecel Bolivia GSM"
+COPN: "73801","DIGICEL"
+COPN: "73802","GUY CLNK PLS"
+COPN: "74000","Movistar"
+COPN: "74001","CLARO"
+COPN: "74001","CLARO"
+COPN: "74401","HOLA PARAGUAY S.A."
+COPN: "744001","HOLA PARAGUAY S.A."
+COPN: "744002","CLARO PY"
+COPN: "74404","TIGO PY"
+COPN: "74405","PY Personal"
+COPN: "744005","PY Personal"
+COPN: "74602","SR.TELESUR.GSM"

+COPN: "74602","SR.TELESUR.GSM"
+COPN: "74603","DIGICEL"
+COPN: "74603","DIGICEL"
+COPN: "74604","UNIQA"
+COPN: "74801","Antel"
+COPN: "74807","Movistar"
+COPN: "748007","Movistar"
+COPN: "748010","CLARO URUGUAY"
+COPN: "75001","C&W FLK"
+COPN: "79502","TM CELL"
+COPN: "79502","TM CELL"
+COPN: "90105","Thuraya"
+COPN: "90106","Thuraya"
+COPN: "90112","MCP Maritime Com"
+COPN: "90114","AeroMobile"
+COPN: "90115","OnAir"
+COPN: "90117","Navitas"
+COPN: "90118","WMS"
+COPN: "901018","WMS"
+COPN: "90121","Seanet"
+COPN: "90126","TIM@sea"
+COPN: "90126","TIM@sea"

THIS PAGE INTENTIONALLY LEFT BLANK

SUPPLEMENTALS: NETWORK TESTING ARCHIVES AND ATTACK VECTOR TAXONOMY

This supplemental material contains an archive of each network test. The archive includes packet captures, configuration files, and log files for each test. Appendix G serves as a guide for the network set-up for each test and the file structure for the packet captures and log files. The material can be accessed at https://calhoun.nps.edu/handle/10945/67451.

This supplemental material contains a reprint of material originally published in the IEEE *Computer* magazine for the convenience of the reader. The article presents a novel attack vector taxonomy that utilizes CUPS in the 5G NR architecture to provide an alternative method for analyzing mobile telephony attack vectors.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Center for Internet Security, "Multiple vulnerabilities in Google Android OS could allow for arbitrary code execution," July 5, 2019. [Online]. Available: https://www.cisecurity.org/advisory/multiple-vulnerabilities-in-google-android-os-could-allow-for-arbitrary-code-execution_2019-069/

[2] Qualcomm, "July 2019 Qualcomm Technologies, Inc. security bulletin," July 1, 2019. [Online]. Available: https://www.qualcomm.com/company/product-security/bulletins/july-2019-bulletin#_CVE-2019-2254.

[3] "CVE-2019-2254 detail," NIST, Gaithersburg, MD, NVD-CVE-2019-2254, 2019. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2019-2254.

[4] "CVE-2018-13887 detail," NIST, Gaithersburg, MD, NVD-CVE-2018-13887, 2018. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2018-13887

[5] M. Lanoue, C. A. Bollmann, J. B. Michael, J. Roth, and D. Wijesekera, "An attack vector taxonomy for mobile telephony security vulnerabilities," in *Computer,* vol. 54, no. 4, pp. 76–84, 2021. doi: 10.1109/MC.2021.3057059

[6] M. Eckstein and M. Shelbourne, "Navy to field early 'Project Overmatch' battle network on Theodore Roosevelt CSG in 2023," USNI News, February 08, 2021. [Online]. Available: https://news.usni.org/2021/02/08/navy-to-field-early-project-overmatch-battle-network-on-theodore-roosevelt-csg-in-2023

[7] *Resources and Requirements Review Board and Naval Capabilities Board*, OPNAV Instruction 5420.117, Department of the Navy, Washington, DC, USA, 2019.

[8] P. Cerwall, P. Jonsson, S. Carson, S. Davis, P. Linder, A. Gomroki, A. Zaida et al., "Ericsson mobility report," Ericcson, Stockholm, Sweden, Tech. Rep. EAB-20:009174, 2020. [Online]. Available: https://www.ericsson.com/4adc87/assets/local/mobility-report/documents/2020/november-2020-ericsson-mobility-report.pdf

[9] GSMArena, "AT&T has officially shut down its 2G network," January 18, 2017. [Online]. Available: https://www.gsmarena.com/at_t_has_officially_shut_down_its_2g_network-blog-22811.php

[10] AT&T, "Get ready, 3G is going away in 2022," May 18, 2021. [Online]. Available: https://www.att.com/support/article/wireless/KM1324171/

[11] Verizon, “3G CDMA network shut off date set for December 31, 2022,” March 30, 2021. [Online]. Available: https://www.verizon.com/about/news/3g-cdma-network-shut-date-set-december-31-2022

[12] E. Dahlman, S. Parkvall, and J. Sköld, *5G NR:The Next Generation Wireless Access Technology*. San Diego, CA, USA: Elsevier, Academic Press, 2018.

[13] S. Mjølsnes, and R. Olimid, “Easy 4G/LTE IMSI catchers for non-programmers,” in *Computer Network Security*, 2017. 235–246. [Online]. doi: 10.1007/978-3-319-65127-9_19

[14] “Base Station (BS) radio transmission and reception (FDD),” 3GPP, Sophia Antipolis, Tech. Spec. 25.104. V16.0.0, Jan. 15, 2019.

[15] “Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception,” 3GPP, Sophia Antipolis, Tech. Spec. 36.104. V16.9.0, Apr. 8, 2021.

[16] “NR; Base Station (BS) radio transmission and reception,” 3GPP, Sophia Antipolis, Tech. Spec. 38.104 V16.7.0, Apr. 8, 2021.

[17] “Synchronization in UTRAN Stage 2,” 3GPP, Sophia Antipolis, Tech. Spec. 25.402 V16.0.0, July 16, 2020.

[18] “Evolved Universal Terrestrial Radio Access (E-UTRA); LTE Positioning Protocol (LPP),” 3GPP, Sophia Antipolis, Tech. Spec. 36.355. V16.0.0, July 24, 2020.

[19] “LTE Positioning Protocol (LPP),” 3GPP, Sophia Antipolis, Tech. Spec. 37.355. V16.4.0, Mar. 29, 2021.

[20] “NR; Requirements for support of Assisted Global Navigation Satellite System (A-GNSS),” 3GPP, Sophia Antipolis, Tech. Spec. 38.171. V16.0.0, July 17, 2020.

[21] K. Krewell, “Qualcomm adds complete RF portfolio; paves way to 5G,” Forbes, February 22, 2017. [Online]. Available: https://www.forbes.com/sites/tiriasresearch/2017/02/22/qualcomm-adds-complete-rf-portfolio-pavews-way-to-5g

[22] Qualcomm, “Snapdragon System-in-Package,” [Online]. Available: https://www.qualcomm.com/products/snapdragon-system-package

[23] European Union Agency for the Space Programme, “Constellation Information,” Accessed 28 May, 2021. [Online]. Available: https://www.gsc-europa.eu/system-service-status/constellation-information

[24] European Union Agency for the Space Programme, “What is SBAS?,” May 8, 2021. [Online]. Available: https://www.euspa.europa.eu/european-space/what-euspace/what-sbas

[25] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen, *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach*, 1st ed. Boston, MA: Birkhäuser Boston, 2007.

[26] “Characteristics of the Universal Subscriber Identity Module (USIM) application,” 3GPP, Sophia Antipolis, Tech. Spec. 31.102. V16.6.0, Dec. 11, 2020.

[27] “Smart Cards; UICC-Terminal interface; Physical and logical characteristics,” ETSI, Sophia Antipolis, Tech. Spec. 102 221 V16.4.0, Apr. 22, 2021.

[28] Osmocom, “EC20.” Accessed January 14, 2021. [Online]. Available: https://osmocom.org/projects/quectel-modems/wiki/EC20

[29] Quectel, *Quectel EC20 Mini PCIe*, 2015. [Online]. Available: https://www.quectel.com

[30] Quectel, *EC20 R2.1 Hardware Design*, 2019. [Online]. Available: https://www.quectel.com

[31] G. Shupeng, H. Zheng, X. Haikuo, and Z. Ye, “All the 4G modules could be hacked,” in *Blackhat USA*, 2019. [Online]. Available: https://i.blackhat.com/USA-19/Wednesday/us-19-Shupeng-All-The-4G-Modules-Could-Be-Hacked.pdf

[32] FreedomFi, “Frequently asked questions.” Accessed January 14, 2021. [Online]. Available: https://freedomfi.com/faq/

[33] Open5GS, “Open source project of 5GC and EPC (Release-16).” Accessed January 14, 2021. [Online]. Available: https://open5gs.org/

[34] OpenAirInterface, “OpenAirInterface: 5G software alliance for democratising wireless innovation.” Accessed January 14, 2021. [Online]. Available: https://openairinterface.org/

[35] srsRAN, “Your own mobile network.” Accessed January 14, 2021. [Online]. Available: https://www.srslte.com/

[36] “3G Security; Specification of the MILENAGE algorithm set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 1: General,” 3GPP, Sophia Antipolis, Tech. Spec. 35.205. V16.0.0, July 10, 2020.

[37] Simjacker Technical Paper, Tech. rep. V1.01, AdaptiveMobile Security Ltd., Dublin, Ire., 10 Oct. 2019.

[38] "AT command set for User Equipment (UE)," 3GPP, Sophia Antipolis, Tech. Spec. 27.007. V16.8.0, Apr. 2, 2021.

[39] L. Ropek, "Qualcomm chip flaw could leave 30 percent of the world's phones vulnerable to hackers," Gizmodo, May 6, 2021. [Online]. Available: https://gizmodo.com/qualcomm-chip-flaw-could-leave-30-percent-of-the-worlds-1846837667

[40] "Security architecture and procedures for 5G System," 3GPP, Sophia Antipolis, Tech. Spec. 33.501. V16.6.0, Apr. 6, 2021.

[41] J. B. Tsui, Fundamentals of Global Positioning System Receivers : a Software Approach , 2nd ed. Hoboken, NJ, USA: John Wiley and Sons, 2005.

[42] National Institute of Standards and Techonology, "National Vulnerability Database." Accessed April 12, 2021. [Online]. Available: https://nvd.nist.gov/

[43] Ettus Research, *USRP B200/B210 Bus Series*, 2019. [Online]. Available: https://www.ettus.com/wp-content/uploads/2019/01/b200-b210_spec_sheet.pdf

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California