

Prototyping and Emulation Studies of Large-Scale 5G Wireless Systems
DESIGN DOCUMENT

Team Number-35
Client- Hongwei Zhang
Advisers-Professor Hongwei Zhang
Team Members Nolan Cardona, Andrew Whitehead, Rohan Willis, Da'Zhawn Davis,
Ousmane Lloyd Ntutume
Team Email sdmay20-35@iastate.edu
Team Website-<https://sdmay20-35.sd.ece.iastate.edu>

Executive Summary

Engineering Standards and Design Practices

- OpenAirInterface software:
 - OAI Radio Access Network (OAI-RAN)
- GITLAB
 - Team Git Repository, OAI Git Repository

Summary of Requirements

- Running simulation of the PHY abstraction layer on two different versions of code (v.0.5.2 and v.1.1.0)
- Management of interference and latency between small cell base-station
- Enhancement of the device to device wireless communication (D2D communication)

Applicable Courses from Iowa State University Curriculum

- Signal and systems 1 and 2 (EE224 and EE324)
- Embedded Systems 1 (CprE 288)
- Linux (Com S 252)
- Operating Systems (CprE 308)

New Skills/Knowledge acquired that was not taught in courses

- Navigate throughout OAI (OpenAirInterface)
- Different layer structure and technologies involved with the 5G NR wireless network
- Installation of virtual machine (VM) and ubuntu 14.04 into personal computer
- Emulation/Simulation command of the PHY abstraction layer using ubuntu
- Explore different branch of the main repository in OAI

Table of Contents

1.1 Acknowledgement	4
1.2 Problem and Project Statement	4
1.3 Operational Environment	4
1.4 Requirements	5
1.5 Intended Users and Uses	5
1.6 Assumptions and Limitations	5
1.7 Expected End Product and Deliverables	6
2. Specifications and Analysis	6
2.1 Proposed Design	6
2.2 Design Analysis	6-7
2.3 Development Process	8
2.4 Design Plan	8
3. Statement of Work	8
3.1 Previous Work And Literature	8
3.2 Technology Considerations	9
3.3 Task Decomposition	9
3.4 Possible Risks And Risk Management	10
3.5 Project Proposed Milestones and Evaluation Criteria	10
3.6 Project Tracking Procedures	11
3.7 Expected Results and Validation	11
4. Project Timeline, Estimated Resources, and Challenges	11
4.1 Project Timeline	12
4.2 Feasibility Assessment	12
4.3 Personnel Effort Requirements	13
4.4 Other Resource Requirements	14
4.5 Financial Requirements	14
5. Testing and Implementation	14
5.1 Interface Specifications	15
5.2 Hardware and software	15
5.3 Functional Testing/Non-Functional Testing	15

5.4 Process	15
5.5 Results	15
6. Closing Material	16
6.1 Conclusion	16
6.2 References/Appendices	17

List of figures/tables/symbols/definitions

- OAI OpenAirInterface: A flexible platform towards an open LTE ecosystem and the emulation of its various layers.
- 5G NR (New Radio): Newest generation of wireless communication using millimeter wave technology
- Layer 1: Physical Layer(Simulate the BLER rate process of each block of a particular resource/signal)
- BLER: Block Error
- Layer 2: Made up of different sublayers:
 - MAC (Medium Access Control)
 - RLC (Radio Link Control)
 - PDCP (Packet Data Convergence Protocol)
- LTE: Long term evolution for mobile telephony
- 3GPP: the third-generation partnership project, which is a standards organization in charge of developing protocols for mobile telephony.
- D2D: term used to specify a device to device communication
- UE: user equipment
- MIMO: Multiple input multiple output

1 Introduction

1.1 ACKNOWLEDGEMENT

We would like to acknowledge our faculty advisor Professor Hongwei Zhang and the PhD Student Advisor Matthias Sander - Frigau for the guidance and assistance on our project

1.2 PROBLEM AND PROJECT STATEMENT

5th generation wireless systems, or 5G NR, are the next generation wireless communications beyond the 4G LTE that we currently have. 5G wireless system is not only an evolution of the legacy 4G LTE networks, but also a system with many new service capabilities. In fact, research

and development toward 5G aim at various advanced characteristics and one of these characteristics is a lower latency and lower energy consumption. In order to accomplish these tasks, large scale emulation of the PHY layer has been done, but because of the high complexity of the PHY layer in terms of algorithms, many simulator systems would rather rely on a PHY abstraction technique to run simulation. It is the case for the OAI which is one of the only tools that can be either run with a full PHY or with PHY abstraction and still obtain the same result but at a way faster simulation time.

The problem that we are faced with in this project is improving a specific version of the codes (v.0.5.2) from the OpenAirInterface5g(OAI) platform, to be able to stimulate and emulate the present PHY abstraction layer, then using that to modify code to enable a device to device communication with 5G network. In order to resolve this issue, we will have to gain knowledge about 5G and how the PHY abstraction is structured, and then use that knowledge to make the needed changes, then test to verify that it emulates at the right speed. The next problem that we will have is getting this to work for device to device connections (D2D). One issue we will face is making sure that we can hold the right frequency between devices signals that are being transmitted and received. We will have to use frequency modulation to help us hold the correct frequency to emulate the 5G. Some of the things we hope to accomplish are learning how 5G works, why it can be beneficial in everyday use and also learn about all other technologies that will be involved in the development of this New Radio network. The outputs of this project will be being able to emulate 5g and use that to connect the device to device.

1.3 OPERATIONAL ENVIRONMENT

The operational environment will mainly be the Gitlab from the OpenAirInterface and the end product will be able to simulate and emulate the 5g physical abstraction layer of a cell with multiple nodes. Once cloning the necessary code from the GitLab, we will run the codes from our repository. As an end product, we should be able to increase the wireless communication speed between UE, and also be able to have the capability of having a device to device connection

1.4 REQUIREMENTS

For this project, we are required to get familiar with the different technologies involved with improving the latency in the 5G core network. Once we acquired enough knowledge, we should set up a virtual machine in our personal computer to start the simulations and we will need Ubuntu 14.03 64-bit architecture. Once our machines are ready for simulation we will need to clone the version of the OAI Gitlab repository codes that we will need for the simulation and we should finally engage into finding codes that will need to be modified in order to accomplish our main goal.

1.5 INTENDED USERS AND USES

It is said that by 2020, 5G NR network should be commercially available to anybody wanting to test out 5g or use 5g speed on their device, this will include pretty much everybody owning a 5G device capability. With that being said, we should also underline the fact that 5G network will enable other technology such as the internet of things, VR, AR, self-driving cars and many more, so any individual in those areas will be strongly impacted by 5G.

1.6 ASSUMPTIONS AND LIMITATIONS

Limitations

- Must utilize the most recent version of the 5G code that still support the PHY abstraction layer simulation. At first we were working with the version 1.0.2 for the basic simulation, but we had to switch to the version 1.1.0 for a better compatibility with the Unbutu.
- Implementation of Device to Device communication as well as having the ability for channel modeling, interference modeling, and radio modeling
- 5G is still under development as we speak, so it is hard to fix codes
- Navigate through the files in OAI

Assumptions

- The Physical Abstraction layers will need to allow for simultaneous modeling to provide an accurate test of the connection between towers and devices
- Minimal code will be added to the existing code revision

1.7 EXPECTED END PRODUCT AND DELIVERABLES

- November - December 2019: Demonstration and performance evaluation of existing OpenAirInterface simulation/emulation systems, detailed design and (preliminary) implementation of the “Physical Layer Abstraction” extension of OpenAirInterface simulation/emulation
- January - March 2020: Spiral evaluation and refinement of the aforementioned enhanced OpenAirInterface simulation/emulation system, as well as the use of the enhanced system for studying 5G wireless solutions
- April 2020: demonstration and report

2. Specifications and Analysis

2.1 PROPOSED DESIGN

As of right now, we were able to successfully clone two different versions of the source codes into our own repository (v.0.5.2 and v.1.03). The version v.0.5.2 is the version in which the PHY abstraction layer is not working as it should, and the version v.1.0.3 does have some aspect of the PHY abstraction layer, but the structure of the codes is different. From these two versions of the codes, we have been running simulations in parallel and observed the result, and the following is just a few differences found:

V1.0.3 compared to V0.5.2

openair1/PHY/CODING

1. *3gpplte.c*

a. *Implements all the interleaver into the class*

Most of the changes were small in this folder. Changing of variable names, added log instead of msg, didn't change much of the logic

openair/PHY/INIT

1. *init_top.c*

a. *Added code. Added functions generate_64qam_table, generate_16qam_table, generate_qpsk_table, init_lte_top and free_lte_top*

2. *lte_init.c*

a. *Added a lot of logic to handle ue, added other logic as well and changed some variable names.*

3. *lte_param_init.c*

a. *Added and removed some code*

Files removed

Rate_matching.c, scrambler.h, vars.h, defs.h, extern.h,

Added files

Phy_init.h, defs_NB_IoT.h, lte_ru.c, lte_init_ue.c,

After a good amount of simulations, we realized that the current version used was missing a lot of functionalities, as the following shows:

Openair1/PHY/TOOLS

Missing files

Twiddle12288.h

Twiddle1536.h

Twiddle24576.h

Twiddle6144.h

Twiddles4096.h

Twiddles8192.h

We then decided to switch to a newer version of the code V 1.1.0 which should allowed for better simulation results.

2.2 DESIGN ANALYSIS

Throughout our analysis, we have been looking over the OAI code and learning about 5G and how it works. We have learned that each cell is composed of different layers and the most relevant to us has been layer 1, which includes the PHY abstraction layer, and we have been running simulations on it to test the MIMO process. The following links are a few results obtained from the simulation.

- <https://trello-attachments.s3.amazonaws.com/5dbc9f8930bbfd355a380d07/5dcf1f3d962fef28bccda2bf/922b031e66e65aff079cac3573486f0e/image.png>
- <https://trello-attachments.s3.amazonaws.com/5dbc9f8930bbfd355a380d07/5dcf1f3d962fef28bccda2bf/38aa6ddb436d2ce6a6c956458ebdde68/image.png>
- <https://trello-attachments.s3.amazonaws.com/5dbc9f8930bbfd355a380d07/5dcf1f3d962fef28bccda2bf/4c0248143a93066e22a8d6ce10db3c19/image.png>
- <https://trello-attachments.s3.amazonaws.com/5dbc9f8930bbfd355a380d07/5dcf1f3d962fef28bccda2bf/7d282ea5af3af0822894e6f928d9ff0f/image.png>

After the simulations, we have been trying to analyze and working on the interpretation of results obtained. Furthermore here is a more detailed composition of the PHY layer functionalities:

Functions:

- Error detection on the transport channel and indication to higher layers
- FEC encoding/decoding of the transport channel
- Hybrid ARQ soft-combining
- Rate matching of the coded transport channel to physical channels
- Mapping of the coded transport channel onto physical channels
- Power weighting of physical channels
- Modulation and demodulation of physical channels
- Frequency and time synchronisation
- Radio characteristics measurements and indication to higher layers
- Multiple Input Multiple Output (MIMO) antenna processing
- Transmit Diversity (TX diversity)
- Digital and Analog Beamforming
- RF(Radio Frequency) processing

2.3 DEVELOPMENT PROCESS

As part of a Spiral development process, throughout our project, we will be using simulator command, including basic simulator command and layer 1 simulator (PHY abstraction layer). In fact, we will try to follow the following PHY layer processing layout:

Physical Layer Processing Layout

Transport block -> CRC (Cycle Redundancy Check) Attachment -> LDPC (Low-density parity-check) Base Graph Selection -> Code block segmentation and CRC attachment -> LDPC encoding -> Rate matching -> Code block concatenation -> Scrambling -> Modulation -> Layer mapping -> Antenna port mapping -> Mapping to RBs (Resource Block)

2.4 DESIGN PLAN

Our design plan was to focus mainly on the physical layer of abstraction. We have been modifying the OAI source code in order to improve the tower to tower and the device to device wireless communication, while minimizing interferences and latency during signal transmission. In order to accomplish that, we have been looking deeper into the abstraction flags dissemination found in one of the C code (pbchsim.c) and the differences observed in each header file.

3. Statement of Work

3.1 PREVIOUS WORK AND LITERATURE

Literature and Relevant Links:

Security for 5G Mobile Wireless Networks:

D. Fang, Y. Qian and R. Q. Hu, "Security for 5G Mobile Wireless Networks," in *IEEE Access*, vol. 6, pp. 4850-4874, 2018. doi: 10.1109/ACCESS.2017.2779146,

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8125684&isnumber=8274985>

http://www.eurecom.fr/~nikaeinn/files/oai/openair_emulator.pdf

<https://www.digitaltrends.com/mobile/what-is-5g/>

Previous Work:

Previous versions of this software have been available for us to analyze and conduct our project off of. This has been beneficial for us as we haven't had to create code from scratch and we have multiple sources to compare algorithms from. This has assisted us in finding existing code that causes issues within the previous versions in order to implement changes into the new version. The shortcomings of following the previous work were, although we had access to the previous code, we weren't provided with sufficient documentation that would give us a clear understanding of the functionality of specific blocks of code. Some of the code we were able to interpret based on labeling and prior classroom experience, but we didn't have the experience or knowledge to understand the main blocks of code we need to focus on. This is why we had to spend an immense amount of time researching the basics of OpenAirInterface and 5G networking.

3.2 TECHNOLOGY CONSIDERATIONS

In terms of completing the project, we didn't have a variety of technological options to choose from, since the previous workings of this project have solely been done via OpenAirInterface. The fact that this was the only technological resource available to us is a strength and a weakness as we could only base our knowledge of what we could implement on the system through similar and previous versions of projects done through the interface. In terms of hardware used to work on the project, we ran into issues using some of our personal computers to run simulations, which kept us behind on the project goal. We finally obtained a more powerful computer, from our advisor, and we have been trying to get caught up.

3.3 TASK DECOMPOSITION

Task 1: Conduct programming activities or google searches in order to revamp C programming skills and usage of Linux operating system

Task 2: Research 5G networking and current developments on area

Task 3: Install necessary software (Ubuntu and Virtual Box) and perform tutorials provided on OpenAirInterface website

Task 4: Practice conducting simulations of the tutorials completed and provided code in OpenAirInterface repository

Task 5: Pull previous versions of project from the provided repository and find differences between the code in the previous versions and in the current master branch, using git commands

Task 6: Using found differences, conduct simulations and tests in order to determine the functionality of each different block of code

Task 7: Search for functionality of different calls, classes, apis, methods, etc. within interface in order to replace previous versions of program

Task 8: Implement changes and start researching device-to-device communication

3.4 POSSIBLE RISKS AND RISK MANAGEMENT

As the OAI is a pretty new platform for most of our group members, we spent a lot of time trying to become familiar with it. In fact, we spent the first 8 weeks of the semester on learning about the platform and its involvement with 5G technology, as the platform has a huge codebase with thousands of codes. Besides, the fact that it is an open resource and everyone can contribute with their idea, it is not impossible to be misled by someone else's code just because he or she did not fix all the bugs in their source code. On top of the main platform that represents a big risk on its own, we have been dealing with the actual installation of the necessary software (VM, Ubuntu..) into our personal device.

In fact, while installing the software in his laptop, one of our group members crashed his computer and lost all information in his internal drive, so it is not only about the credibility of some code found in the OAI that we need to worry about, as we could all lose our personal computer. Furthermore, another risk with the project is concerning the right acquiring the right version of software (Ubuntu 14.04), as our advisor did specify that the use of other versions of Ubuntu would not be compatible with the PHY abstraction layer codes we are trying to run. Also, the cloning and branching process of the code took us some time to assimilate, as it is not as simple as copying and pasting a code to our repository.

In order to manage some of these risks, we had to get our advisor involved for more guidance and details about each step of the process, and fortunately we had a lot of good tips from him, which helped us a lot in our overall simulation task. In fact, from our meeting we decided to switch our simulation work to a more recent version of the source code, as we went from V1.0.3 to V1.1.0. Concerning our computer issue, we were able to acquire a more powerful laptop, which helped us to carry on with the simulations.

3.5 PROJECT PROPOSED MILESTONES AND EVALUATION CRITERIA

Our overall project is about modifying the PHY abstraction layer source in order to enable communication between small cells and nodes with minimum interference and latency. So in order to accomplish these tasks, we first had to get familiar with the OAI, which we did for about 2 weeks. After that, we started to look more into the PHY layer folder in the Gitlab and we

have been trying to isolate the source code needed for our goal, and then we cloned and branched out the necessary code from a specific versions (v.0.5.2) to our repository.

Next, we started the simulation with the basic simulator and we run a few tests to observe the result, and then we have been asked to repeat the process, but with a different version of the codebase (v.1.0.3). As for the rest of the semester and next semester, we have also been simulating the PHY abstraction layer of a most recent source code version (v.1.1.0), and then we have been focusing more toward the analysis of the results observed. We have also tried to isolate the part of the codes that need to be modified in order to observe the cell and the node and how changing their number in the code will affect the simulation result. Finally, We were also supposed to enable the D2D communication, but the lack of time could not allow us to go further in our project. So we decided to just focus on the basic simulations and the difference observed..

3.6 PROJECT TRACKING PROCEDURES

We have been using the trello application to monitor the tasks needed to be accomplished and to assign new ones during each of our team meetings. Also, for our group communication about the evolution of the project, we have been using groupMe application.

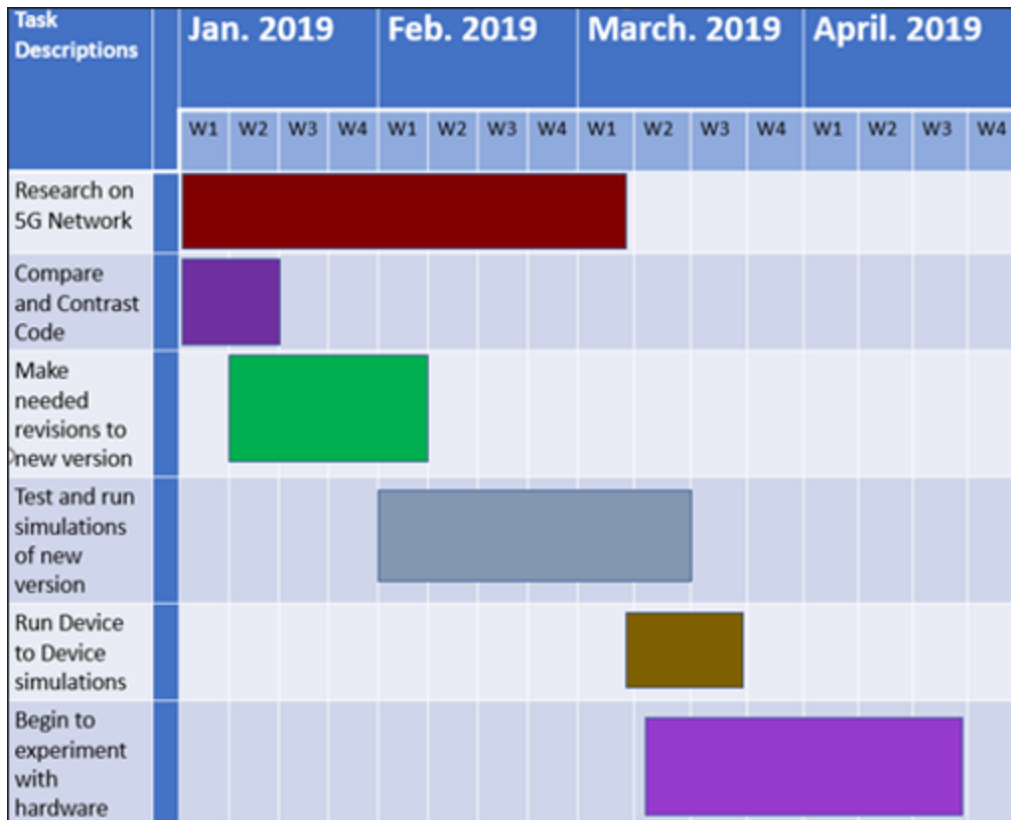
3.7 EXPECTED RESULTS AND VALIDATION

The desired outcome was to find the problems that openairinterface 5G had with an older branch in the physical abstraction layer. make the needed improvements to emulate 5g speed. Then take code and implement device to device connectivity. The way to confirm that our solution would work at a high level was by running a simple test on our code between a single UE and node. The test would give us the results of the speed of the connection. Once we were successful with running the latency test, we would expand our test to larger scale simulation, with multiple cells and node communication. the expected results should present a low latency and less interference between UE.

4. Project Timeline, Estimated Resources, and Challenges

4.1 PROJECT TIMELINE

- A realistic, well-planned schedule is an essential component of every well-planned project
- Most scheduling errors occur as a result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity
- A detailed schedule is needed as a part of the plan:



4.2 FEASIBILITY ASSESSMENT

Some of the challenges of the project is understanding all the code and what they each do . There is currently a lot of code and most of the code is poorly documented so it can be hard to know what the code is doing. Also, trying to fully understand how 5G works in a short time and understand what improvements we can make that will work and improve the speed. Based on our timeline, we assumed that our project was going to be feasible, and that by the end of the semester we would have successfully isolated the codes needed to enable a better cell to cell, cell to nodes and D2D communication. Unfortunately, we were not able to complete the full scale of our initial goals, but we still manage to run the simulations that emphasized the missing source codes needed to fulfil some main objectives.

4.3 PERSONNEL EFFORT REQUIREMENTS

Group member	Project contribution
Andrew Whitehead	<ul style="list-style-type: none"> - Organization of google drive folder for all necessary materials - Headed research on the first two layers of 5G and its functionality - Analyzed the previous and current versions of OAI code in order to find

	<p>differences between them</p> <ul style="list-style-type: none"> - Looked into the XML files and attempted to run simulations on outdated versions of the program - Compared the PHY and Simulation folders in v0.5.2 and v1.1.0 to determine physical abstraction issue - Created documents to interpret the functionality of each method and function within the c files - Ran basic simulations within the simulation folder through ubuntu
Nolan Cardona	<ul style="list-style-type: none"> - Created the Gnatt chart that would be used to demonstrate the overall progress of the project - Performed research on signal modulation as well as 5G emulation - Assisted in the creation and generation of the simulations that would be used to test the results of the different versions of code
Dazhawn Davis	<ul style="list-style-type: none"> - Put code into our team repository - Ran simulations on code - Analyzed the two branches of the code - Try to find ways to improve code - Researched how 5g works
Lloyd Ousmane Ntutume	<ul style="list-style-type: none"> - Analyzed the master branch of the Gitlab and recorded the different source file - in charge of uploading the meeting notes and recording to our google drive
Rohan Willis	<ul style="list-style-type: none"> - Understanding 5G emulation and downloading the virtual machine - Ran simulations of the v0.5.2 code on the needed platform

4.4 OTHER RESOURCE REQUIREMENTS

- OpenAirInterface
- Ubuntu

- Virtual Box Simulator

4.5 FINANCIAL REQUIREMENTS

As a software based project, the necessary financial requirements were already available online as an open source, so we did not discuss the financial aspect of the project.

5. Testing and Implementation

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or a software library

Although the tooling is usually significantly different, the testing process is typically quite similar regardless of CprE, EE, or SE themed project:

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study for functional and non-functional requirements)
2. Define the individual items to be tested
3. Define, design, and develop the actual test cases
4. Determine the anticipated test results for each test case
5. Perform the actual tests
6. Evaluate the actual test results
7. Make the necessary changes to the product being tested
8. Perform any necessary retesting
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you've determined.

5.1 INTERFACE SPECIFICATIONS

You need Ubuntu 64-bit LTS version 14.04.3 and Linux low-latency kernel version 3.19

5.2 HARDWARE AND SOFTWARE

Ubuntu is a complete Linux operating system. Low-latency kernel optimizes the process of very high volume of networking with minimal delay. With the kernel downloaded on top of the operating system it allows for testing in a real-time setting without any Radio Frequency equipment.

5.3 FUNCTIONAL TESTING/NON-FUNCTIONAL TESTING

For testing you will have to make an xml file and define some variables. Inside that file you need to define the area for the 5g coverage, how many nodes will be using the 5g, where the base tower is within the area defined, application type for the 5g connection, and the time that you will be emulating.

5.4 PROCESS

Each method was tested by using the same setting inside an xml file on both branches and comparing the files and making sure they get the same results.

5.5 RESULTS

– List and explain any and all results obtained so far during the testing phase

```
[OTG][D][1][0] 0x ffff][m2m Aggre 0][Flow 0][Type 0/no_predefined_traffic] TX
INFO pkt at time 637 Size= [payload 1] [Total 69] with seq num 0, state=3 : |
87172722610615949091771159776736563948129390885096385611598481030444|4|
[OTG][D][0]MY_SEQ 1
[OTG][I]traffic type 0
[EMU][I][UE 1] ADD pkt to OTG buffer with size 98 for dst 0 on rb_id 1
[OTG][I]Mod_id 0 Frame 0 Got a packet (0x7fdb2c00a3f0), HEAD of otg_pdcpc_buffer
[1] is (nil) and Nb elements is 0
[OTG][D][UE 0] Frame 0: sending packet 0 from module 0 on rab id 1 (src 1/5a0c,
dst 0) pkt size 98
[PDCP][D][FRAME 00000][ UE][MOD 00][RNTI 5a0c][DRB 01] Data request
notification pdu size 100 (header2, trailer0)
[PDCP][D]Setting PDU as a DATA PDU
[PDCP][D]Sequence number 0 is assigned to current PDU
[PDCP][D]Following content with size 100 will be sent over RLC (PDCP PDU header
is the first two bytes)
[RLC][D][FRAME 00000][ UE][MOD 00][RNTI 5a0c]rlc_data_req: rb_id 1 (MAX 11),
mulp 0, confirmP 0, sdu_sizeP 100, sdu_pP 0x13933300
[RLC][D][FRAME 00000][ UE][MOD 00][RNTI 5a0c][RB 1] Display of rlc_data_req:
[RLC][D]RLC_TYPE : 2
[RLC][D][FRAME 00000][ UE][MOD 00][RNTI 5a0c][DRB UM 01] rlc_um_data_req()
RLC_UM_DATA_REQ size 100 Bytes, BO 0 , NB SDU 0
[PDCP][D]Data sending request over RLC succeeded!
[PHY][D][UE] PUCCH 1a
[PHY][D][UE 0][SR 5a0c] Frame 63 subframe 8 Checking for SR TXOp
(sr_ConfigIndex 5)
[PHY][D][UE 0] Frame 63 subframe 7: Doing phy procedures UE RX(14)
```

This screenshot shows a log file. In the log file it shows trying to send a packet to another node and that node receiving that packet

6. Closing Material

6.1 CONCLUSION

In conclusion, based on our timeline, we were able to successfully clone two different versions of the master codebase of the PHY layer from the OAI repository and then branched them to our own repository available via our website.

Then, we could run simulations and isolated some parts of the source codes responsible for the latency. In fact, we have been running simulations from the openAir1 source file, which includes the PHY abstraction codes, and we could identify missing codes in previous versions of the code, such as V.1.0.3. In order to move forward we had to change our main source code version from v.1.0.3 to a more recent one v.1.1.0, which seemed to possess a similar architecture than the master branch and it was also more compatible with the version of ubuntu used.

6.2 REFERENCES/APPENDICES

Overview:

https://www.openairinterface.org/?page_id=2762

https://www.openairinterface.org/docs/workshop/1stOAINorthAmericaWorkshop/Training/KALT-ENBERGER-OAI_basics_June_2019.pdf

Getting started:

https://www.openairinterface.org/?page_id=25

Tutorials:

<https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirUsage>

Emulation:

https://www.openairinterface.org/?page_id=791

Predictable, real-time wireless networking algorithms for control and AR/VR:

PRKS: <http://www.ece.iastate.edu/~hongwei/group/publications/PRKS-TWC.pdf>

ktRT: <http://www.ece.iastate.edu/~hongwei/group/publications/pktRT-TII.pdf>

More 5G Links:

<https://www.rfwireless-world.com/Terminology/>

<https://www.rfwireless-world.com/5G/>

<https://www.rfwireless-world.com/Terminology/5G-Protocol-Stack-Layer-1-Layer-2-and-Layer-3.html>

<https://ieeexplore.ieee.org/Xplore/home.jsp>

<https://ieeexplore.ieee.org/document/8125684>