LTE base station scheduling information decoder

ZESEN ZHANG

The major model in wireless network is the connection between base station and multiple user equipment (UE). Base stations are designed to communicate with multiple UE, which leads to the requirement of scheduling the transmission at down-link channel. Learning the scheduling algorithm behind different base stations gives us opportunity to evaluate the performance of base stations under different circumstances. The scheduling algorithm is hidden in the control channel between base station and UE. In this project, we design a tool to decode the LTE control channel information and obtain the allocation information to learn the scheduling algorithm from signal. We test our tools with the simulated signal generated from matlab and found our tool is able to get 100% accurate result under idle test case. We also collected data from real world base stations and is going to further test the tools in real world.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: LTE, base station, control channel, scheduling

ACM Reference Format:
Zesen Zhang. 2023. LTE base station scheduling information decoder. *ACM Trans. Graph.* 37, 4, Article 111 (August 2023), 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Cellular network is starting taking more and more communication market. Different Internet Service Providers(ISPs) bought base stations from different brands. Base stations are designed to communicate with multiple UE. UEs sharing the same base station are actively listening and competing for receiving packets from base station. When packets come from wired network, base station will open up a buffer for each UE before sending out the packet to UEs (Fig. 1). Directly sending out these packets will cause packet congestion and drain too much power from UEs [Balasubramanian et al. 2009]. This leads to the requirement of scheduling the transmission at down-link channel after receiving packets from Internet [Ghosh et al. 2010]. While designing scheduling algorithm, base stations have to consider trade-off in different environments. We assume that providers will manage a pre-designed scheduling algorithm in base stations to deal with different circumstance. In commercial world, however, Internet Service Providers tend to purchase base station from certain brands (e.g. Ericsson, Nokia, Huawei) and deploy them without considering the efficiency of base stations in different environments.

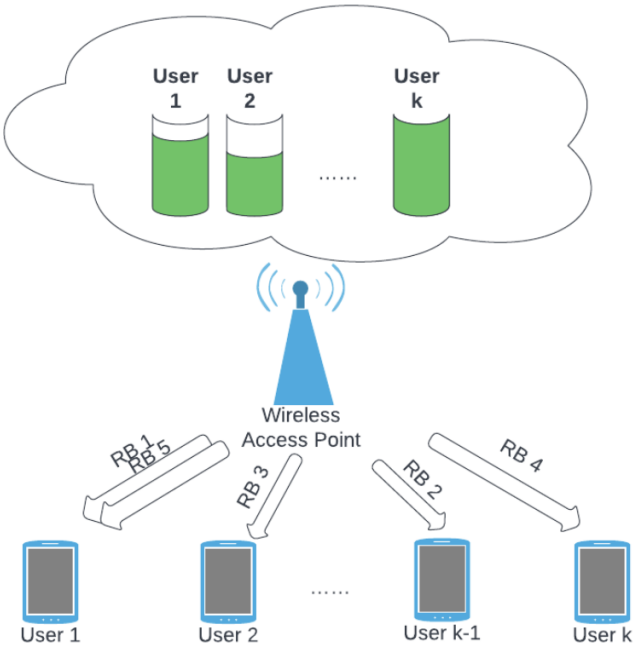

Fig. 1. Packet scheduler open buffers for each user. It use specific algorithm to schedule transmission to UE and once the user has been selected for transmission, the number of bits to be transmitted is based on user's SINR.

Motivated by this, we would like to use this project to figure out a way to capture scheduling pattern in base stations for further evaluating their performance in different base stations.

The opportunity to capture scheduling pattern remains in decoding control channel signal broadcasting from base stations. The base station broadcasts all the wireless parameters that affect its scheduling algorithm, including the modulation and coding rate (MCS), the number of MIMO spatial streams, the allocated frequency bandwidth, and the aggregated base stations, the way of allocating its data to UEs, to the mobile users via a physical control channel. By carefully design a decoding process of this channel gives us chance to learn how base stations allocate packets for certain number of UEs transmitting certain number of data. To further study the scheduling algorithm for different base stations, we can put these information into machine learning models to learn their models.

This project presents a tool we designed for decoding physical control channel with matlab, a tool that exposes the fine-grained UE and RB-allocation information related applications. This tool decodes the physical layer control channels, extracting the most likely users and their allocated packet position bitmaps in signals for each user. We accurately monitor the downlink delivery process of every packet the user receives in the Physical Layer (PHY).

Author's address: Zesen Zhang, zez003@ucsd.edu.

0730-0301/2023/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

The challenge we are facing is the unknowns. The key challenge of designing the this tool is that we do not have any pre-known information about a base station as well as the UEs served by the base station. First of all, the ID of mobile users, i.e., the C-RNTI, are unknown. This is required for both message verification and receiver identification. Without this information, we will not be able to verify our results and throw away the error result. Secondly, the total number and the distribution of control messages contained inside the control channel is unknown. Besides that, for each possible control message, its format are also unknown which gives multiple possible decoding pattern of a same message. To deal with this, we exhaustively searching all the possibility of the results and put four filters to proceed to the most possible result.

We also faces a problem with lacking of ground-truth. In the real world, we will never be able to know how many users are actually using the base station and we will never be able to know their C-RNTIs. This brings the difficulty for us to verify the accuracy of our results. To deal with this, we leverage the matlab LTE tool box to simulate LTE iq samples and feed into our tools to test the accuracy.

In section 2 we are going to introduce the background of LTE signals as well as the related work of decoding LTE signals. We also show a test for one of the most recent work and point out its problem. In section 3 and 4, we are going to show our design of this tool and in section 5 we evaluate our tool with our simulated LTE signals. In section 6, we will give a conclusion of the work.

2 BACKGROUND AND RELATED WORK

Demodulating LTE signals requires a thorough understanding of the LTE signal structure and advanced algorithms to extract information from the signal. In this section, we are going to briefly introduce the LTE signals and we will show our test result on one of the most recent work on decoding LTE signals.

2.1 LTE signals

An LTE signal is organized in frames of 10ms, each composed of ten 1ms subframes. Each subframe consists of 14 OFDM symbols and has a 2D resource grid of time and frequencies. The resource grid for a single subframe includes from 72 to 1200 subcarriers, and a resource block is a 12-by-7 set of resource elements. When an UE is turned on, it first has to detect and connect to the LTE network. LTE can be deployed with bandwidths ranging from 1.4MHz to 20MHz, with the information required for connection placed in the narrowest bandwidth. This information includes primary and secondary synchronization signals, the broadcast channel carrying the Master Information Block (MIB), and the Physical Control Format Indicator Channel. Each subframe includes 14 OFDM symbols, with the first few symbols reserved for control information and payload data going into the remaining symbols. The number of control symbols varies from subframe to subframe and is signaled by the Physical Control Format Indicator Channel. The network includes three additional channels: the Physical Hybrid Indicator Channel, Physical Downlink Control Channel, and Physical Downlink Shared Channel. Figure 2 shows the channels in a 10 MHz subframe.

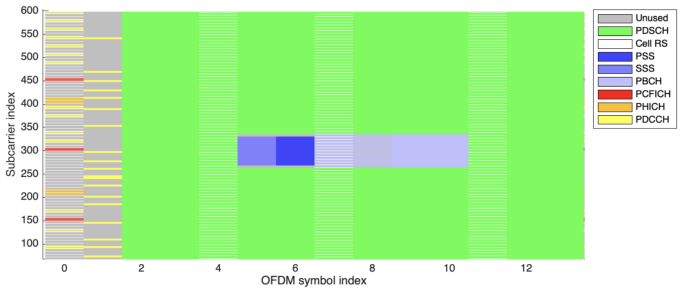

Fig. 2. Subframe 0 for 10MHz bandwidth showing the PDSCH, PSS, SSS, PBCH, PCFICH, PHICH, PDCCH.

LTE base station's information is provided by PSS, SSS and PBCH. PSS (Primary Synchronization Signal) and SSS (Secondary Synchronization Signal) are signals used in LTE networks to help devices synchronize with the network. PSS provides information about the frame timing, while SSS provides information about the cell identity. These signals are transmitted periodically, typically every five subframes, to ensure that devices stay synchronized with the network. PBCH (Physical Broadcast Channel) is a channel used in LTE networks to broadcast system information to devices. This channel carries the Master Information Block (MIB), which contains important information about the network, such as the cell bandwidth and cell identity. The PBCH is transmitted periodically, typically every 10 subframes.

The control information is contained in PCFICH and PDCCH. PCFICH (Physical Control Format Indicator Channel) is a channel used in LTE networks to indicate the format of the control information transmitted in the Physical Downlink Control Channel (PDCCH). The PCFICH signals how many OFDM symbols in each subframe are used for PDCCH transmission. PDCCH (Physical Downlink Control Channel) is a channel used in LTE networks to transmit control information to devices. This channel carries information such as which part of the resource grid is allocated to a particular user, which modulation and coding scheme is used, and whether data is being transmitted using PDSCH or PUSCH (Physical Uplink Shared Channel). The PDCCH is divided into Control Channel Elements (CCEs), which can be assigned to different users based on their priority. This control channel information provides us with the scheduling patterns that base stations use for transmitting packet with multiple users.

And the real data is provided by PDSCH and PHICH. PDSCH (Physical Downlink Shared Channel) is a channel used in Long-Term Evolution (LTE) networks to carry user data. It is a shared channel, meaning that multiple users can use it simultaneously. The data carried by PDSCH is encoded using channel coding techniques to improve its reliability and to ensure that it is correctly received by the user's device. PHICH (Physical Hybrid ARQ Indicator Channel) is a channel used in LTE networks to carry Hybrid Automatic Repeat Request (HARQ) acknowledgments. HARQ is a technique used to improve the reliability of data transmission by requesting retransmission of lost or corrupted data. The PHICH signals whether data

sent from the device to the network has been successfully received or not.

2.2 Related work

A few LTE decoding tools have already be invented. LTEye[Kumar et al. 2014] and OWL[Bui and Widmer 2016] introduces two passive LTE sniffers. They pressumed knowledge of user ID so that they are using bit errors inside the control message as an indicator to conduct message validation. However, in our case we show these information is vulnerable to channel fading and interference, and thus introduces a huge amount of false positives.

Another commercial software Airscope [Systems 2018] that uses the software stack of srsLTE [Gomez-Miguelez et al. 2016] and decodes for LTE downlink traffic in real-time. It is a high-performance LTE air interface analyzer. AirScope is a turnkey solution which provides the ability to passively observe, capture and analyze all activity in an LTE cell. The application works by detecting all active users in a given cell and decoding the control information broadcast by the network to those users. The problem with Airscope is srsRAN team already stopped maintaining it and they are not open sourced.

NG-Scope, proposed by Xie and Jamieson [Xie and Jamieson 2022], is an LTE control channel decoder that deals with the physical layer control channels of multiple base stations simultaneously. It achieves better carrier aggregation and channel capacity estimation by combining the data across users within one cell and fusing data across cells. It also monitors the downlink delivery process more accurately by reconciling control messages from the cellular physical layer with the packet arrival time series from the transport layer of the User Equipment (UE).

2.3 Problem with NG-Scope

In our work, we want to first try to test the accuracy of the most recent state-of-art NG-Scope. If the accuracy is precise enough, we will leverage their tools to decode LTE signal, otherwise we will build the tool on our own.

First of all, we simulated LTE data from Matlab by leveraging its LTE toolbox. It contains several most of the symbol channel we need for encoding and decoding LTE packets as well as transmitting signal over SDR. In this tool, they also have built up tools to simulate Reference Measurement Channel(RMC) which is normally used for test communication between base station and UE. We specify the CellID to be 4 and transmitting to a UE whose RNTI is 255. In order to avoid any potential LTE signal interference, we chose 2GHz as center frequency of the transmitting signal.

We set up NG-Scope [Xie and Jamieson 2022] on Ubuntu 22.02 and run it on USRP B200. We first have ng-scope listen at 2GHz frequency channel and confirm there is no LTE signal on this frequency band. Then we start transmitting Matlab simulated LTE waveform. We ran the simulation for around 15 seconds.

NG-Scope successfully catches the LTE signal and saved the decoding data in dcilog format. We found it caught 565 log whose RNTI is 255. However, it caught 86 "fake" RNTI. And these RNTIs are not just showing occasionally. For example, RNTI 34408 showed up 91 times. And it shows 100 tranmission from a UE whose RNTI is 41802. We listed top 10 RNTIs showed up in the result (Fig. 3)

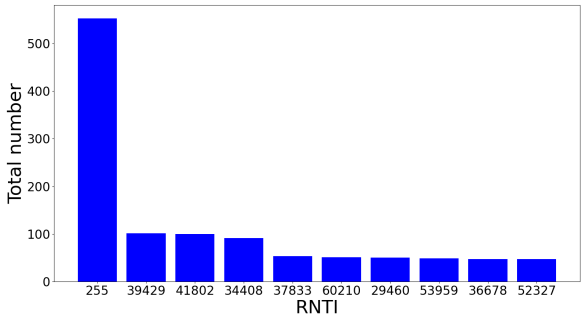

Fig. 3. Top 10 RNTIs showed up in decoded result file.

However, in our experiment, we want to figure out the scheduling pattern in different brands of base stations. These amount of "fake" RNTI will make our analysis impossible since they can not be easily removed and we have no way to differentiate base stations' scheduling pattern in different number of UEs' scenario.

In this situation, we are going to design our own LTE decoder to capture LTE control channel information.

3 DESIGN

We first introduce how we demodulate LTE signals then the control channel decoder as well as the filter we put to get rid of "fake" RNTIs and get the most accurate C-RNTIs.

3.1 Demodulate LTE signals

First of all, in order for a User Equipment (UE) to communicate with the network, it needs to obtain some basic system information of base station carried by the Master Information Block (MIB). The MIB contains essential system information such as system bandwidth, system frame number (SFN), and Physical Hybrid Automatic Repeat Request (HARQ) Indicator Channel (PHICH) configuration. It is transmitted on the Broadcast Channel (BCH) and mapped into the Physical Broadcast Channel (PBCH) with a fixed coding and modulation scheme that can be decoded after the initial cell search procedure.

To decode MIB, LTE signal contains primary synchronization signal and secondary synchronization signal for UE to sync up with base station. Since PSS and SSS are both only using the 72 subcarriers in the center of each subframe, we downsampled the iq data first and use cross-correlation to capture the peak of signal and get the beginning point of subframe. In the meanwhile, we perform cell searching to determine the duplex mode of signal and the cyclic prefix pattern. After that, we do CFO correction, OFDM demodulation and channel estimation with synchronization signals. Once the UE is fully synchronized with base station, we are able to demodulate PBCH channel and decode MIB to get the base station information including its bandwidth, CellID, Cell-specific reference signals and Frame/subframe number.

After we decoded the MIB and obtained the new base station information, we need to resample the data according to the cell-tower's bandwidth, correct CFO, demodulate the OFDM symbol and do channel estimation again before decoding the control channel.

Once the UE obtains information from the MIB, it can decode the Control Format Indicator (CFI) which indicates the Physical Downlink Control Channel (PDCCH) length. This enables the UE to decode and search for Downlink Control Information (DCI) messages carried on the PDCCH. If a DCI message CRC masked with System Information Radio Network Temporary Identifier (SI-RNTI) indicates that a SIB is carried in the same subframe, the SIBs are transmitted in the Broadcast Control Channel (BCCH) logical channel. Generally, BCCH messages are carried on the Downlink Shared Channel (DL-SCH) and transmitted on the Physical Downlink Shared Channel (PDSCH). The format and resource allocation of the PDSCH transmission is indicated by a DCI message on the PDCCH.

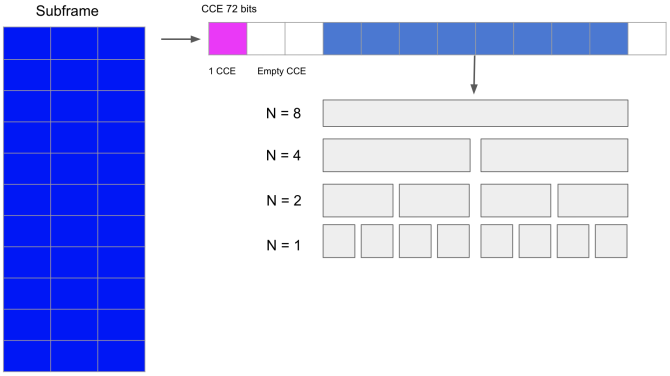

Fig. 4. PDCCH are formated in multiple CCEs and one CCE contains 72 bits. PDCCH has 4 different types of format each aggregates N=1,2,4 or 8 number of CCEs to transmit control message according to the channel SNR.

3.2 Decode control channel

The Physical Downlink Control Channel (PDCCH) carries scheduling assignments and other control information in the form of Downlink Control Information (DCI) messages. A PDCCH is transmitted on one Control Channel Element (CCE) or an aggregation of several consecutive CCEs. Each CCE corresponds to 9 Resource Element Groups (REGs). In PDCCH transmission, only the REGs which are not assigned to Physical Control Format Indicator Channel (PCFICH) or Physical Hybrid Automatic Repeat Request (PHICH) are used. Each REG contains 4 Resource Elements (REs), thus defining the mapping of control channels to resource elements.

PDCCH has 4 different formats decided by SNR of the environment. Each format aggregates N = 1,2,4 or 8 CCEs to form a DCI message. Each CCE contains 72 bits (Fig. 4). And each DCI message can have 12 different formats which contains different number of bits. The length of bits ranging from 27-45. The size of the convolutional encoded control message is smaller than one CCE so a message is repeated multiple times if one DCI message contains more than 1 CCEs, which provides extra redundancy over the convolutional code and thus extra protection over bit errors (Fig. 5). The base station transmits nothing in empty CCEs that contain no control messages. Therefore, once we found a consistent 8 CCEs, we will have 48 different ways to decode the message and might get 48 different results.

The empty CCEs will definitely not contain any information and it will set as a margin for two control messages. And at the empty space there will be just some gausian noise. So that we applied heuristic method to check the power inside of each CCE and skip the control message contains empty CCE.

One DCI message contains convolutional encoded message with a control message and an appended CRC code. The appended CRC code is the 16-bit CRC bits encoded from the control message xor with C-RNTI (Fig 6. Therefore, when a normal UE wants to decode the control message, it will use its C-RNTI to xor with the appended CRC and match with the computed CRC from control message to check whether it is the message for it or not.

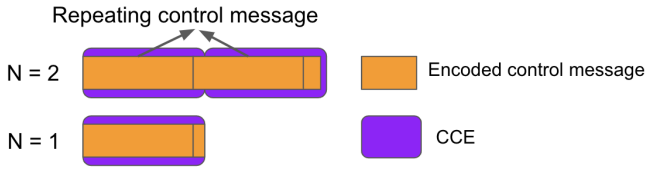

Fig. 5. Each control message has 12 different types of formats and the length of different type ranges from 27-45 bits. The control message will repeat in CCEs to padding the whole bits.

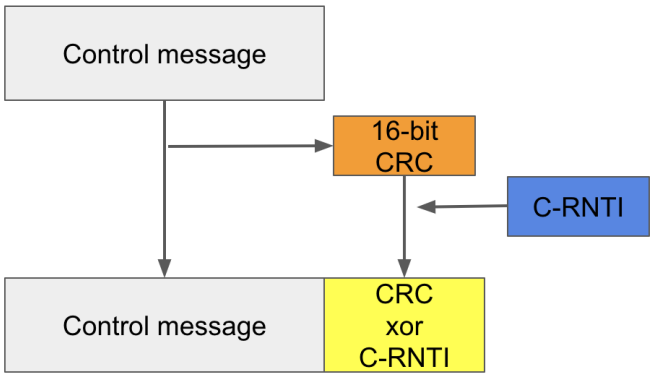

Fig. 6. Control message encode with appended CRC to get the encoded control message to put into PDCCH.

In this sense, in our work, we are going to exhaust searching all the possible formats of control message and get all the possible C-RNTIs in different PDCCH formats as well as different DCI formats. Once we get a list of candidate C-RNTIs, we performs four filters to get rid of all fake C-RNTIs and pick the most possible accurate C-RNTIs' list for each subframe. We maintain a conflict RNTI list where RNTIs are competing for the an overlapping space in the

control channel and we applied the following filter to pick the right RNTI for the space.

Location filter. We first check the validity of the RNTI result with the specific search space. The UE-specific search space carries control information specific to a particular UE and is monitored by at least one UE in a cell. Unlike the common space search, the starting location of the UE-specific search space may be varied for each subframe or UE. However, the starting location of the UE-specific search space is determined in every subframe using a hash function, as specified in TS36.213, Clause 9[Procedures 2009].

Active user filter. Normally in the real world, once a UE starts using Internet, it will send and receive multiple packets. From the control channel performance, we will see one C-RNTI keeps showing up in the next several subframes or frames. If this case happens, those RNTIs are very likely to be the real RNTIs and should be chosen as the active RNTI. To maintain this filter, we design a least recent used (LRU) vector which contains the most recent showed up RNTIs. And whenever a new RNTI showed up, we compared to the former RNTI list, and if any of them showed up in former time, we record them as the candidate RNTI.

PDSCH Energy filter. DCI message includes an allocation bitmap which shows where the location of resource blocks that the data is assigned to the UE. If the decoding of the DCI message is correct, we should be able to find the data power in the corresponding location showed up in PDSCH. Therefore, the third filter we built up is to check the power of the corresponding PDSCH to further filter out the "fake" RNTI.

Active frequency filter. If all of the former filter is not able to deal with all of the conflicts, the last filter we put is to use the check the frequency of the C-RNTIs that showed up competing for the same space. Since the real RNTI will show up much more time then the "fake" RNTI and normally RNTI will show up no more than 5 times in LRU. Therefore, this filter will be the last filter which will guarantee to pick one RNTI for the result.

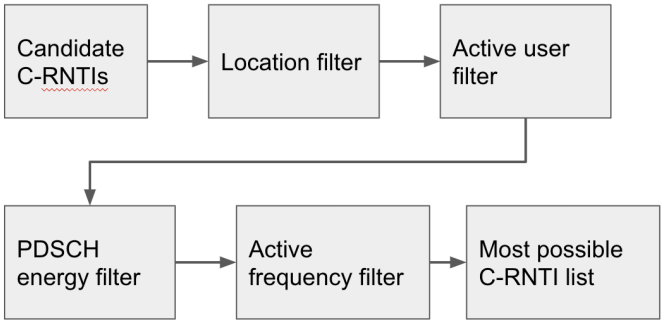

Fig. 7. We designed four filters to get rid of the most number of "fake" RNTIs and get the most accurate C-RNTI list.

4 IMPLEMENTATION

Our design is a pure software solution and designed on matlab with its LTE Toolbox, so we believe that it can be implemented on any laptop with matlab and with LTE Toolbox installed. To feed the data into our tool, we support feeding IQ samples captured from air into our tool. We also develop matlab tools to generate IQ samples which can be directly feed into our tool. Also, for people with SDR board, they can also transmitting a LTE waveform from one board using matlab tool box and use another SDR board to capture the IQ samples from another SDR board shown in the following fig **??**.

Fig. 8. Test bed set up for people who has SDR boards to transmit from one side and capture the signal from another side.

5 EVALUATION

We evaluate our tool with simulated IQ samples generated from matlab. We leverage their LTE waveform generator and modify their code to generate a more realistic LTE signals. Current LTE tool box could not generate more than 100 subframes and will repeat transmitting the same subframes aftewards. With our code, we are able generate the subframes as much as we want and directly converted to IQ samples and feed into our tools. We tried several test case with different RMC standards with FDD duplex mode and contains different RNTIs and our tool can always gets the accurate result. Compared to NG-Scope result (Fig 3) our tool did a higher accuracy in simulation results.

We also captured real world base station IQ samples from a base station in UCSD near hopkins parking lot9. Our tool are successfully capturing their Cell tower information but the real world signal has higher noise then the signal we simulated in matlab. Therefore, we

saw in several subframes, the PDCCH channel contains power but the our tool and filters are not able to decode them. We are trying to further improve our tools to be able to deal with these corner cases and will make a better tool to decode the real world signals.

Fig. 9. "Touchable" AT&T LTE base station on campus.

6 CONCLUSION & FUTURE WORK

In this project, we developed a tool by leveraging matlab LTE tool box to decode LTE control message in its PDCCH. Our tool is able to successfully decode the simulated LTE signals generated from matlab. We are also in the middle of putting more filters to deal with corner cases we observed from real world signals.

For the future work, we will make our matlab as a baseband to decode LTE signals and investigate into NG-Scope since it is a open sourced code and it can decode LTE signal in real world which is much quicker and we do not need to always capture a large amount of IQ samples. We will compare the NG-Scope result with our matlab result to check the error case happened in NG-Scope and check the reason to debug NG-Scope so that we will be able to make NG-Scope better and can capture and decode LTE signal in real time. In this case, we will be able to feed those results captured from NG-Scope into machine learning models and learn the scheduling pattern difference developed by different base stations.

7 TEAM CONTRIBUTION

We started our team with two people, but the other teammate made zero contributions to the technical part and almost none to the writing part (just two lines in the midterm report and maybe one line in the proposal). As a result, I removed her name from the final report. Although she appears to still be enrolled in the class, I will be taking credit for all the work done on this project.

REFERENCES

Niranjan Balasubramanian, Aruna Balasubramanian, and Arun Venkataramani. 2009. Energy consumption in mobile phones: a measurement study and implications for network applications. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*. 280–293.

Nicola Bui and Joerg Widmer. 2016. OWL: A reliable online watcher for LTE control channel measurements. In *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. 25–30.

Amitava Ghosh, Rapeepat Ratasuk, Bishwarup Mondal, Nitin Mangalvedhe, and Tim Thomas. 2010. LTE-advanced: next-generation wireless broadband technology. *IEEE wireless communications* 17, 3 (2010), 10–22.

Ismael Gomez-Miguelez, Andres Garcia-Saavedra, Paul D Sutton, Pablo Serrano, Cristina Cano, and Doug J Leith. 2016. srsLTE: An open-source platform for LTE evolution and experimentation. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*. 25–32.

Swarun Kumar, Ezzeldin Hamed, Dina Katabi, and Li Erran Li. 2014. LTE radio analytics made easy and accessible. *ACM SIGCOMM Computer Communication Review* 44, 4 (2014), 211–222.

Physical Layer Procedures. 2009. document 3GPP TS 36.213.

Software Radio Systems. 2018. Airscope. http://www.softwareradiosystems.com/products/

Yaxiong Xie and Kyle Jamieson. 2022. Ng-scope: Fine-grained telemetry for nextg cellular networks. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6, 1 (2022), 1–26.