Hochschule München

Fakultät für Informatik und Mathematik
Master Informatik

Mobile Netze
Projektdokumentation
SS 2016

Thema: Analyse von LTE Netzwerken
mit srsUE/srsLTE und openLTE

Betreuer: Prof. Dr. Lars Wischhof

Autor/en: Martin Eder
Andreas Kowasch
Christian Nagy
Michael Wolf

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 LTE

LTE ist ein Mobilfunktstandard der vierten Generation (3.9G) und steht für **Long Term Evolution**. Mit 300 Mbit/s bietet es im Downlink wesentlich höhere Datenraten als seine Vorgängerprotokolle. UMTS[1] mit der Erweiterung HSDPA[2] (3.5G) bietet in seinen höheren Ausführungen lediglich 42 Mbit/s. UMTS (3G) ohne Erweiterungen kommt durchschnittlich auf 384 kbit/s und der noch ältere Standard EDGE[3] (2.75G) liefert 220 kbit/s.

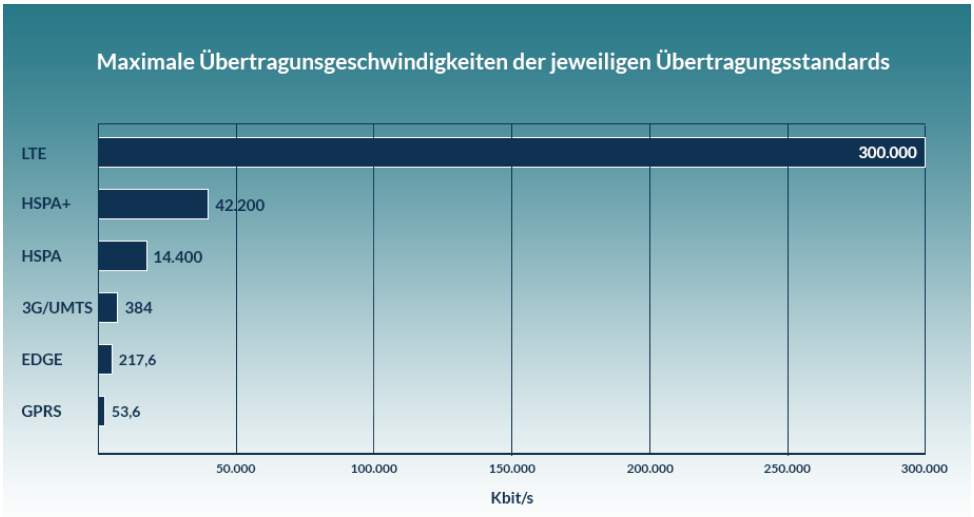

Abbildung 1.1.1: Downlink-Datenraten [sma16]

Es existiert eine Erweiterung namens **LTE-Advanced** (4G), die die mögliche Datenrate nochmals steigern kann. Diese liegt je nach Ausprägung und Gerätestandard bei bis zu 1 Gbit/s.

In Deutschland werden für LTE mehrere Frequenzen von den unterschiedlichen Providern eingesetzt. Sie liegen bei 700, 800, 1800 und 2600 MHz (vgl. Abbildung 1.1.2).

[1] Universal Mobile Telecommunications System
[2] High Speed Downlink Packet Access
[3] Enhanced Data Rates for GSM Evolution

	Provider	Downlink	Uplink	
LTE 700	Dt. Telekom	713 - 723 MHz	766 - 778 MHz	
	Vodafone	723 - 733 MHz	778 - 788 MHz	
	O2 Telefonica	703 - 713 MHz	758 - 768 MHz	
LTE 800/900	Provider	Downlink	Uplink	
	Dt. Telekom	811-821 MHz & 900-915 MHz	852-862 MHz & 945-960 MHz	
	Vodafone	801 – 811 MHz	842 – 852 MHz	
	O2 Telefonica	791–801 & 880-890 MHz	832–842 & 925-935 MHz	
LTE 1800	Provider	Downlink	Uplink	
	Dt. Telekom	1805 – 1835 MHz	1710 – 1740 MHz	
	Vodafone	1855 – 1880 MHz	1760 – 1785 MHz	
	O2 Telefonica	1835 – 1855 MHz	1740 – 1760 MHz	
LTE 2600	Provider	Downlink	Uplink	TDD*
	Dt. Telekom	2640 – 2660 MHz	2520 – 2540 MHz	2605 – 2610 MHz
	Vodafone	2620 – 2640 MHz	2500 – 2520 MHz	2580 – 2605 MHz
	O2 Telefonica	2670 – 2690 MHz	2540 – 2580 MHz	2610 – 2620 MHz

Abbildung 1.1.2: LTE Frequenzen in Deutschland [anb16]

1.2 Projektaufgabe

- Projekt: Analyse LTE-Mobilfunknetzwerk
- Software: srsLTE/UE und openLTE (Inbetriebnahme/Dokumentation)
- Hardware: Ettus Research USRP B210, testweise USRP N210

Das Ziel des Projekts ist es kommerzielle LTE-Mobilfunknetze der Provider zu analysieren. Da bei LTE die meisten Datenpakete verschlüsselt sind, sollen nur die unverschlüsselten Daten Gegenstand der Analyse sein. Zudem soll die Analyse passiv stattfinden. Dies bedeutet, dass keine aktive Verbindung zum Mobilfunknetzwerk (eNodeB) aufgebaut wird.

Als UE stehen jeweils ein Ettus Research USRP B210 USB 3.0 inkl. GPS und Ettus Research USRP N210 inkl. GPS zur Verfügung.

Die Inbetriebnahme der Hardware soll mit der Open Source Software srsLTE/UE erfolgen. Auch die Open Source Software openLTE soll getestet werden. Beide Programme müssen mit allen Abhängigkeiten kompiliert und auf einem Linux-Rechner installiert werden. Anschließend soll die Installation und die Funktionsweise, im Rahmen der Analyse, dokumentiert werden.

Hauptgegenstand der LTE-Analyse sind die Datenpakete MIB und alle empfangbaren SIBs. Diese sollen unter anderem mit Hilfe von oben genannten Tools empfangen, dekodiert und dann manuell interpretiert werden.

Neben der Datenanalyse soll auch der Standort der eNodeB Sendemasten bestimmt werden. Dazu sollen die vom UE empfangenen Daten ausgewertet werden. Mit den daraus gewonnen Informationen sollen die geographischen Koordinaten der Mobilfunkmasten unter Zuhilfenahme von Datenbanken im Internet ermittelt werden.

srsLTE / srsUE / srsGUI

The project is partially based on the srs-tools, which are provided by the company „software radio systems“[4]:

srsLTE is a free and open-source LTE library for SDR UE and eNodeB. It is a highly modular library with minimum inter-module or external dependencies. The codebase is entirely written in C and, if available in the system, uses the acceleration library VOLK distributed in GNURadio.

srsUE is written in C++ and builds upon the aforementioned srsLTE library. srsUE is released under the AGPLv3 license and uses software from the OpenLTE project for some security functions and for RRC/NAS message parsing.

openLTE

OpenLTE is an open source implementation of the 3GPP LTE specifications. The focus is on transmission and reception of the downlink. Currently the developer is working on extending the capabilities of the GNU Radio applications and adding capabilities to the simple base station application. Although the project exists since 2011, it's only available as an alpha version.

[4] http://www.softwareradiosystems.com

Kapitel 2

Konfiguration

2.1 srsLTE / srsUE / srsGUI

Dependencies

The following setup should work on any Debian-based Linux distribution if there are recent Ettus UHD-drivers in its software repository. It is mandatory to **install UHD-drivers before anything is going to be compiled**, otherwise the srsLTE library will not contain support for any hardware (bladeRF is another possibility).

1. First the are needed some prevalent development tools:

```
sudo apt -y install cmake build-essential git pkgconf
   checkinstall
```

2. The UHD-driver can be installed like this:

```
sudo apt -y install libuhd-dev libuhd003 uhd-host
```

3. Install srsUE dependencies:

```
sudo apt -y install libmbedtls-dev
```

4. Install srsGUI dependencies:

```
sudo apt -y install libboost-system-dev libboost-test-dev
   libboost-thread-dev libqwt-dev libqt4-dev
```

5. Wireshark is not a dependency, but it is useful to dissect .pcap dump files from srsUE.

```
sudo apt -y install wireshark
```

Math Kernel Library (MKL)

Intels Math Kernel Library provides some optimized math routines and the srs-suite has optional support for it. The library can be obtained for free at Intels website[1] after a short registration process.

There are some minor drawbacks at the system we used, as the installer does not add the MKL to the systems dynamic linker run-time bindings. If MKL is installed into the default location, this can be fixed with the following commands. Take into account to change the second path to the exact version you are using.

```
sudo sh -c "cat << EOF > /etc/ld.so.conf.d/mkl.conf
/opt/intel/mkl/lib/intel64
/opt/intel/compilers_and_libraries_2017.0.098/linux/compiler/lib/
   intel64_lin
EOF"
sudo ldconfig
```

Compiling srsGUI

The srsGUI library is compiled by the following commands. We use checkinstall here instead of a plain „make install“ to build (and install) a .deb-file, which can be handled by Debians standard package management utilities.

```
git clone https://github.com/suttonpd/srsgui.git
cd srsgui
mkdir build
cd build
cmake ../
make
sudo checkinstall -y --pkgname srsgui
```

[1] https://software.intel.com/en-us/intel-mkl

Compiling srsLTE

The main library „srsLTE“ shares its dependencies with the UHD drivers (VOLK-library, Boost), hence it is not necessary to install them separately by hand. For cmake to be able to find MKL, the additional „export“-command is mandatory.

```
git clone https://github.com/srsLTE/srsLTE.git
cd srsLTE
mkdir build
cd build
export MKL_DIR=/opt/intel/mkl/
cmake ../
make
sudo checkinstall -y --pkgname srslte
```

Compiling srsUE

The compiling of srsUE is equal to the previous processes. If you encounter a compiler error, stating that *„error: 'constexpr' needed for in-class initialization of static data member ...“*, you can try the commented fix with sed.

```
git clone https://github.com/srsLTE/srsUE.git
cd srsUE
mkdir build
cd build
#sed -i s/const /constexpr /g ../ue/hdr/radio/radio.h
cmake ../
make
```

Ettus USRP Images

For the Ettus driver to be able to flash the USRP firmware images onto a device, it's necessary to download them beforehand. The following command will download the default images so srsLTE won't complain about not finding them.

```
#download uhd firmware images
sudo /usr/lib/uhd/utils/uhd_images_downloader.py
```

2.2 openLTE

Prerequisites

- Ettus USPR B210
- Antenna VERT900
- USB 3.0 Interface
- Modern multicore CPU
  (Intel Core i5, Core i7 or equivalent with SSE4.1 SSE4.2 and AVX support)

OpenLTE is not only requiring a huge amount of processing power, but it also requires a very low latency due its need to transmit/receive a radio frame every 1ms. If there is any delay in the processing, the system will not going to be able respond in time and will lose samples. Therefore it is recommended to switch off any CPU and/or system features (mostly in your BIOS) which can cause any delays or can slow down the so called context switching time. „Intel SpeedStep“, deep and deeper sleep states etc. should be turned off. Especially with high bandwidth setups (10, 15 and 20MHz) it is recommended to switch off the GUI on Linux.

Dependencies

- UHD drivers (for USRP B210)
- GNURadio
- PolarSSL

GNURadio and UHD drivers

It's recommend not to use the binary version of GNURadio, but to compile the code. It already exists a build script, that handles the whole process of getting sources from git repositories, install dependent packages and build GNURadio. The supported systems are Fedora, Ubuntu, Redhat, Debian, Mint and OpenSuse.

It should be noted that existing installations of GNURadio will be deleted by the script. It must do this to prevent problems due to interference between a linux-distribution-installed Gnu Radio/UHD and one installed from GIT source.

To execute the script, it is necessary to be an ordinary user with sudo-rights and approximately 500MB of free disk space to perform the build. The whole process may take up to three hours to complete, depending on the capabilities of the used system.

To build the HEAD of GNURadio's master branch with latest UHD drivers and multiple concurrent jobs to speed up the process, give the following command:

```
$ mkdir gnuradio
$ cd gnuradio
$ wget http://www.sbrac.org/files/build-gnuradio
$ chmod a+x build-gnuradio
$ $./build-gnuradio -v -ja -m
```

An overview of all possible options can be achived by the following command:

```
$ ./build-gnu-radio -h
Usage: build-gnuradio [--help|-h] [-v|--verbose] [-jN] [-ja]
                      [-l|--logfile logfile ] [-u|--users ulist]
                      [-m] funcs

-m              - Use HEAD of *master* branch, rather than *maint*.
-o              - Use v3.6.5.1

-v|--verbose    - turn on verbose logging to stdout

-jN             - have make use N concurrent jobs

-ja             - have make use N concurrent jobs with auto setting
                  of N (based on number of cores on build system)

-u|--users ul   - add comma-separated users to 'usrp' group in
                  addition to calling user ( $USER )

-l|--logfile lf - log messages to 'lf'
-ut <tag>       - set tag for UHD checkout to <tag>
-ucf <ucflags>  - set UHD CMake flags to <ucflags>
-gt <tag>       - set tag for Gnu Radio checkout to <tag>
-gcf <gcflags>  - set Gnu Radio CMake flags to <gcflags>
-e|--extras     - add an item to "extras" to be built after
                  Gnu Radio/UHD/gs-osmosdr

available funcs are:

all             - do all functions
prereqs         - install prerequisites
gitfetch        - use GIT to fetch Gnu Radio and UHD
uhd_build       - build only UHD
firmware        - fetch firmware/FPGA
gnuradio_build  - build only Gnu Radio
mod_groups      - modify the /etc/groups and add user to group usrp
mod_udev        - add UDEV rule for USRP1
mod_sysctl      - modify SYSCTL for larger net buffers
```

PolarSSL

Another dependency of openLTE is PolarSSL. Installation of this package can be achived with the following command:

```
$ sudo apt-get install libpolarssl-dev
```

Difficulties

At first we decided to use Debian Testing (Stretch) as our operating System for the openLTE installation, since it has the most recent packages. At the current time, some of the necessary packages are missing and the build script is outdated. After a switch to Debian Stable (Jessie) the build script had to be slightly modified, because it covers only an earlier version ob Debian Jessie. After that everything worked as expected.

Functionality

To check the communication with the Ettus USRP B210 the device must be connected to a USB 3.0 port and the following command, which load the FPGA code to the device, must be entered:

```
$ uhd_usrp_probe

linux; GNU C++ version 4.8.2; Boost_105400; UHD_003.008.001-42-
   g8c87a524

-- Operating over USB 3.
-- Initialize CODEC control...
-- Initialize Radio control...
-- Performing register loopback test... pass
-- Performing CODEC loopback test... pass
-- Asking for clock rate 32.000000 MHz...
-- Actually got clock rate 32.000000 MHz.
-- Performing timer loopback test... pass
-- Setting master clock rate selection to 'automatic'.
  _____________________________________________________
 /
|       Device: B-Series Device
|     _______________________________________________________
|    /
|   |       Mboard: B200
|   |   revision: 4
|   |   product: 1
|   |   serial: F54xxx
|   |   FW Version: 7.0
|   |   FPGA Version: 4.0
```

```
|    |
|    |   Time sources: none, internal, external, gpsdo
|    |   Clock sources: internal, external, gpsdo
|    |   Sensors: ref_locked
|    |     ____________________________________________________
|    |    /
|    |    |       RX DSP: 0
|    |    |   Freq range: -16.000 to 16.000 MHz
|    |     ____________________________________________________
|    |    /
|    |    |       RX Dboard: A
|    |    |     ______________________________________________________
|    |    |    /
|    |    |    |       RX Frontend: A
|    |    |    |   Name: FE-RX2
|    |    |    |   Antennas: TX/RX, RX2
|    |    |    |   Sensors:
|    |    |    |   Freq range: 50.000 to 6000.000 MHz
|    |    |    |   Gain range PGA: 0.0 to 73.0 step 1.0 dB
|    |    |    |   Connection Type: IQ
|    |    |    |   Uses LO offset: No
|    |    |     ______________________________________________________
|    |    |    /
|    |    |    |       RX Codec: A
|    |    |    |   Name: B200 RX dual ADC
|    |    |    |   Gain Elements: None
|    |     ____________________________________________________
|    |    /
|    |    |       TX DSP: 0
|    |    |   Freq range: -16.000 to 16.000 MHz
|    |     ____________________________________________________
|    |    /
|    |    |       TX Dboard: A
|    |    |     ______________________________________________________
|    |    |    /
|    |    |    |       TX Frontend: A
|    |    |    |   Name: FE-TX2
|    |    |    |   Antennas: TX/RX
|    |    |    |   Sensors:
|    |    |    |   Freq range: 50.000 to 6000.000 MHz
|    |    |    |   Gain range PGA: 0.0 to 89.8 step 0.2 dB
|    |    |    |   Connection Type: IQ
|    |    |    |   Uses LO offset: No
|    |    |     ______________________________________________________
|    |    |    /
|    |    |    |       TX Codec: A
|    |    |    |   Name: B200 TX dual DAC
|    |    |    |   Gain Elements: None
```

Compiling

To compile openLTE the following commands must be entered:

```
$ wget https://sourceforge.net/projects/openlte/files/openlte_v00
   -20-02.tgz/download
$ mv download openlte_v00-20-02.tgz
$ tar xzf openlte_v00-20-02.tgz
$ cd openlte_v00-20-02
$ mkdir build
$ cd build
$ sudo cmake ../
$ sudo make
$ sudo make install
```

2.3 Ettus N210

The USRP N210 is a network based variant of the USRP B210 version. Unfortunately we couldn't get it to work with srsLTE/UE. The communication with the device was working, but the srsLTE example tools did not show any data. This is a result of hard-coded parameters for B210 in the srsLTE library. The srsLTE developers currently don't have access to N210 hardware and therefore can't adapt the different parameters in their software.

To configure a new IP address of a N210 device you have to attach it directly to a arbitrary host systems Gigabit Ethernet interface and send the following command. After this you have to power cycle the device to make the changes work.

```
#set new ip (power cycle)
sudo /usr/lib/uhd/utils/usrp2_recovery.py --ifc=eth0 --new-ip=X.X.X.X
```

To update the image manually, you can use „uhd_image_loader“ program. It is sufficient to do this only if one of the programs tells you to.

```
#load new image
sudo uhd_image_loader --args="type=usrp2,addr=X.X.X.X"
```

2.4 Wireshark

Wireshark must be configured for reading srsUEs „.pcap“-files. It can do this by utilizing its mac-lte-framed dissector. Go to „Edit → Preferences → Protocols → DLT_USER → Edit“ and add a new entry with DLT=147 and the payload protocol „mac-lte-framed“.[2]

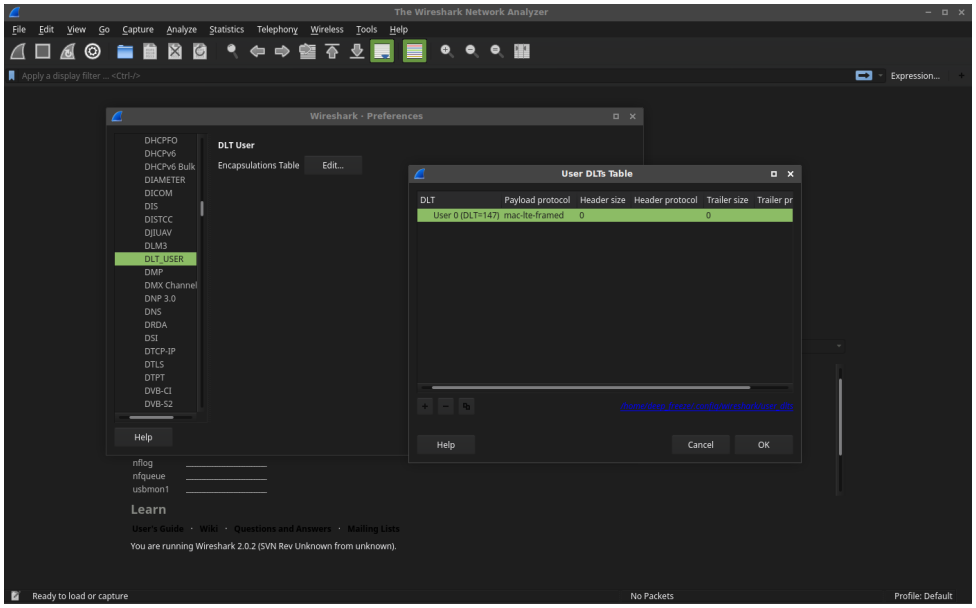

Abbildung 2.4.1: Wireshark

[2] https://wiki.wireshark.org/MAC-LTE

Kapitel 3

Analyse

3.1 Anwendungsprogamme

Zur Analyse von LTE Netzwerken stellen srsUE/srsLTE und openLTE einige nützliche Programme zur Verfügung.

srsLTE: cell_search

Das Kommandozeilenprogramm cell_search sucht das angegebene Frequenzband nach Mobilfunkzellen ab. Man findet das Programm unter folgendem Pfad: „srsLTE/build/srslte/examples/“.

```
Usage: ./cell_search [agsendtvb] -b band
-a RF args [Default ]
-g RF gain [Default 70.00 dB]
-s earfcn_start [Default All]
-e earfcn_end [Default All]
-n nof_frames_total [Default 100]
-v [set srslte_verbose to debug, default none]
```

Erläuterung der verwendeten Abkürzungen der Programmausgabe von cell_search:

- EARFCN: EUTRA Absolute radio-frequency channel number
- PSS: Primary synchronization signals [1]
- PRB: physical resource blocks [2]

[1] http://www.simpletechpost.com/2012/06/primary-and-secondary-synchronization.html
[2] http://www.teletopix.org/4g-lte/resource-block-physical-resource-block-in-lte/

srsLTE: cell_measure

Das Kommandozeilenprogramm cell_measure gibt für die angegebene Frequenz verschiedene Information zu Frequenz und Signalstärke aus. Voraussetzung ist, dass auf der Frequenz eine Funkzelle gefunden wird. Man findet das Programm unter folgendem Pfad: „srsLTE/build/srslte/examples/“.

```
Usage: ./cell_measurement [aglnv] -f rx_frequency (in Hz)
-a RF args [Default ]
-g RF RX gain [Default -1.00 dB]
-l Force N_id_2 [Default best]
-n nof_subframes [Default -1]
-v [set srslte_verbose to debug, default none]
```

Erläuterung der verwendeten Abkürzungen der Programmausgabe von cell_measure:

- PSR: Peak to side-lobe ratio
- MIB: Master Information Block
- SFN: System Frame Number
- SIB1: System Information Block 1

  Contains information regarding whether or not UE is allowed to access the LTE cell. It also defines the scheduling of the other SIBs. carries cell ID, MCC, MNC, TAC, SIB mapping. [3]
- CFO: carrier frequency offset
- SFO: Signal frequency offset
- RSSI: Received Signal Strength Indicator [4]
- RSRP Referenz Signal Received Power [5]
- RSRQ: Reference Signal Received Quality
- SNR: signal-to-noise ratio

openLTE: LTE_fdd_dl_scan

Das Kommandozeilenprogramm LTE_fdd_dl_scan sucht ebenfalls, wie das aus der srsLTE Bibliothek bekannte cell_search, das angegebene Frequenzband nach Mobilfunkzellen ab. Zusätzlich zum Frequenzband kann entweder eine einzelne Kanalnummer oder eine Liste von Kanalnummern welche im Frequenzband existieren angegeben werden. Dadurch ist es möglich zuvor gefundene Zellen anhand ihrer Kanalnummer erneut zu analysieren ohne das gesamte Frequenzband durchsuchen zu müssen.

[3] http://www.rfwireless-world.com/Terminology/LTE-MIB-SIB-system-information-blocks.html
[4] http://www.lte-anbieter.info/technik/rssi.php
[5] http://www.lte-anbieter.info/technik/rsrp.php

LTE_fdd_dl_scan kann allerdings nicht nur Mobilfunkzellen finden. Es kann auch von Mobilfunkzellen gesendete MIBs und SIBs empfangen und auswerten. Bei srsUE/srsLTE geht das lediglich über ein gesondertes Programm.

```
$ LTE_fdd_dl_scan
linux: GNU C++ version 4.9.2; Boost_105500; UHD_003.011.000.git-78-
   gf70dd85d

*** LTE FDD DL SCAN ***
Please connect to control port 20000
```

```
$ telnet localhost 20000
help
ok
***System Configuration Parameters***
  Read parameters using read <param> format
  Set parameters using write <param> <value> format
  Commands:
    start    - Starts scanning the dl_earfcn_list
    stop     - Stops the scan
    shutdown - Stops the scan and exists
    help     - Prints this screen
  Parameters:
    band = 1
    dl_earfcn_list = 25,26,27,...,575
    repeat = on
start
ok
```

Erläuterung der Programmausgabe von LTE_fdd_dl_scan:

- dl_earfcn: EUTRA Absolute radio-frequency channel number
- phys_cell_id: Physical Cell Identifier (PSS + 3*SSS)
- sfn: System Frame Number
- n_ant: Number of antennas
- phich_dur: PHICH Duration (normal or extended)
- phich_res: Number of PHICH groups (1/6, 1/2, 1, 2)

3.2 LTE Datenpakete aufzeichnen

Nachfolgend wird erläutert wie man vorzugehen hat, wenn man unverschlüsselte LTE-Datenpakete empfangen und dekodieren möchte. Mit srsLTE war es uns möglich SIB 1, 2

und 3 zu dekodieren. Weitere SIB konnten nicht empfangen werden, da srsLTE noch nicht alle SIB dekodieren kann oder da bestimmte SIB nicht vom Provider benutzt werden. Des Weiteren wurde der MIB empfangen und verschiedene Paging Datenpakete. Mit Hilfe von openLTE konnten wir zusätzlich SIB 5 und 6 dekodieren.

Der Standort des UE sollte so gewählt werden, dass die GPS Antenne freien Blick zum Himmel hat für die Zeitsynchronisation mit den GPS-Satelliten. Ohne die exakten Uhrzeit ist keine Synchronisation mit einer Funkzelle möglich. Die Programme von srsLTE/UE brechen die Ausführung ab und melden einen Synchronisationsfehler.

Das vorgehen lässt sich in folgenden Schritten zusammenfassen: Frequenzen der Funkzellen in der Umgebung suchen. Auf jeder Frequenz jeweils nach Datenpaketen lauschen.

3.2.1 Funkzellensuche

Zuerst möchte man erfahren, welche Funkzellen in der Umgebung sind und auf welchen Frequenzen diese senden. Mit den Programmen cell_search (srsLTE) und LTE_fdd_dl_scan (openLTE) wird dazu auf einem angegebenen Frequenzband nach Funkzellen gesucht. Am Ende des Suchdurchlaufs wird für jedes Band die gefundenen Frequenzen angegeben. Jede gefundene Frequenz stellt einen Funkmasten dar. In Deutschland werden die Frequenzbänder 3, 7, 8, 20 und 28 benutzt[6].

Beim Scannen der Frequenzbänder ist darauf zu achten, dass die entsprechende Antenne verwendet wird. Jede Antenne deckt nur einen vordefinierten Frequenzbereich ab. Von uns wurde die Antenne „VERT900 Antenna“ für die Messungen verwendet. Diese deckt einen Frequenzbereich von 824-960 MHz / 1710-1990 MHz ab[7].

Beispiel cell_search (srsLTE)

Auf Band 20 wird nach LTE Mobilfunkzellen gesucht:

```
$ sudo sudo ./cell_search -b 20
Found 1 cells
Found CELL 1815.0 MHz, EARFCN=1300, PHYID=180, 100 PRB, 2 ports, PSS
    power=-23.3 dBm
```

Am Ende der Programmausgabe ist zu sehen, dass eine Funkzelle gefunden wurde. Neben der Frequenz ist auch die Angabe der Signalstärke von Interesse. Ist die Signalstärke schlecht kann dies zu Problemen wie Verbindungsabbrüchen führen. Deshalb sollte der Standort des UE so gewählt werden, dass die Signalstärke ausreichend stark ist. In diesem Beispiel sehen wir, dass die Signalstärke mit -23.3 dBm ausreichend ist.

[6] http://www.lte-anbieter.info/ratgeber/frequenzen-lte.php
[7] https://www.ettus.com/product/details/VERT900

Beispiel LTE_fdd_dl_scan (openLTE)

Auf Band 20 wird nach LTE Mobilfunkzellen gesucht:

```
$ LTE_fdd_dl_scan
$ telnet localhost 20000
write band 20
start
info channel_found_begin freq=806000000 dl_earfcn=6300 freq_offset
   =991.153503 phys_call_id=95 sfn=172 n_ant=2 phich_dur=Normal
   phich_res=1 bandwidth=10
info sib1_decoded freq=806000000 dl_earfcn=6300 freq_offset
   =991.153503 phys_cell_id=95 sfn=184 mcc[0]=262 mnc[0]=02 network
   [0]=Vodafone resv_for_oper[0]=false tac=48035 cell_id=20664834
   cell_barred=false intra_freq_resel=allowed q_rx_lev_min=-126
   q_rx_lev_min_offset=0 band=20 si_win_len=40 si_periodicity[0]=32
   sib_mapping_info[0]=2,3 si_periodicity[1]=64 sib_mapping_info
   [1]=5,6,7 duplex_mode=fdd si_value_tag=14
...
info channel_found_end freq=806000000 dl_earfcn=6300 freq_offset
   =991.153503 phys_cell_id=95
```

Wie bereits erwähnt und in der Programmausgabe ersichtlich, werden auch die SIBs dekodiert. Es ist laut Entwickler möglich die dekodierten Informationen in einer Pcap-Datei zu sammeln. Das funktionierte jedoch zum Zeitpunkt der Projektdurchführung nicht und konnte in der zur Verfügung stehenden Zeit nicht behoben werden. Damit endet die Untersuchung der Ergebnisse von openLTE an dieser Stelle.

3.2.2 Funkzellenauswertung

Mit dem Programm srsUE kann auf einer Frequenz nach Daten gelauscht werden. Das Programm findet sich unter dem Pfad „srsUE/build/ue/src/ue“. Aus dem vorhergehenden Suchlauf mit cell_search entnehmen wir, dass eine LTE-Funkzelle auf der Frequenz 1815.0 MHz sendet. Um Datenpakete mitzulesen wird das Programm 'ue' mit folgenden Parametern aufgerufen:

```
$ sudo ./ue --rf.dl_freq 1815000000  --pcap.filename ./Scans/1815.0
   MHz.pcap ue.conf
```

Das Kommandozeilenprogramm wird durch eine Konfigurationsdatei gesteuert. Einzelne Konfigurationsparameter können jedoch auch mit Parametern überschrieben werden. Dies ist hier der Fall für „rf.dl_freq"(Downlink Frequenz) und "pcap.filename“ (Packet-Capture/Wireshark). In der Konfigurationsdatei ist es wichtig folgende Einstellung zu setzen:

```
[pcap]
enable = true
```

Weitere Parameter, wie zum Beispiel das Logging, können je nach Anwendungsfall entsprechend konfiguriert werden. Eine Angabe welches UE verwendet wird ist nicht notwendig, da die Software das über USB 3.0 angeschlossene Gerät selbstständig erkennt. In der Beispielkonfigurationsdatei findet sich zu jedem Parameter eine Erklärung (srsUE/-build/ue/src/). Man kann auch das Programm mit dem Parameter „–help“ starten, um die Hilfe aufzurufen.

Folgende Ausgabe erhält man neben der Pcap-Datei durch oben genannten Aufruf:

```
Setting frequency: DL=1815.0 Mhz, UL=1710.0 MHz
Searching for cell...
Found CELL ID: 180 CP: Normal  , CFO: 3.6 KHz.
Trying to decode MIB...
- Cell ID:          180
- Nof ports:        2
- CP:               Normal
- PRB:              100
- PHICH Length:     Normal
- PHICH Resources: 1/6
- SFN:              0
MIB received BW=20 MHz
Initializating cell configuration...
Setting Sampling frequency 23.04 MHz
SIB1 received, CellID=3841,  PLMN Id: MCC 262 MNC 1 PLMN Id: MCC 262
    MNC 1
SIB2 received
Random Access Transmission: seq=1, ra-rnti=2
Random Access Transmission: seq=2, ra-rnti=2
PCCH message received 61 bytes
```

Es wird angegeben, dass eine Funkzelle auf der Frequenz 1815.0 MHz (Downlink) erfolgreich gefunden wurde. Nachdem Erhalt des MIB (Master Information Blocks) kann begonnen werden nach SIB (System Information Block) Paketen zu horchen. Hier wurden SIB 1 und SIB 2 empfangen. Bei SIB 1 ist sofort ersichtlich, dass es sich um den Provider „Deutsche Telekom“ handelt. Der MCC (Mobile Country Code) gibt das Land an. Für Deutschland ist das die 262. Der MNC (Mobile Network Code) gibt den Provider an. Die 1 steht für die Telekom.

Zur weiteren Analyse wird nun die Pcap Datei „1815.0MHz.pcap“ mit Wireshark geöffnet. Dabei ist zu beachten das Wireshark so konfiguriert sein muss, dass LTE Pakete interpretiert werden können. Siehe dazu das Kapitel „Konfiguration - Wireshark“.

Die oben gezeigt Abbildung zeigt die Übersicht über die empfangenen Daten in Wireshark. Von besonderem Interesse war für dieses Projekt die Werte von MCC, MNC TAC und CellID auszulesen. Damit lässt sich mit Hilfe von Datenbanken wie OpenCellID.org oder

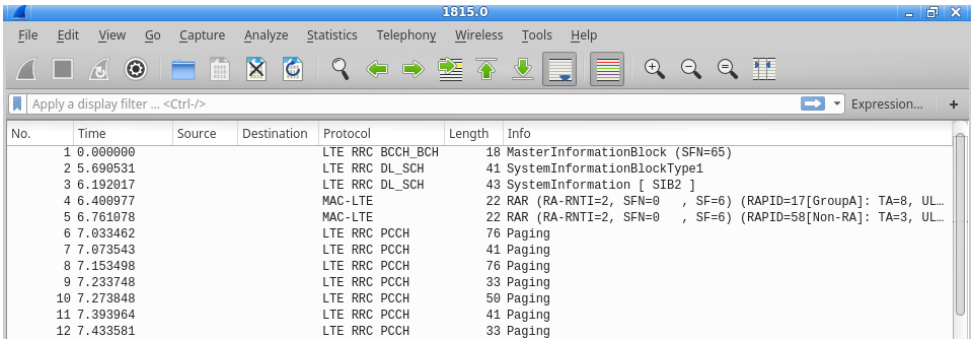

Abbildung 3.2.1: Wireshark 1815.0MHz.pcap

Google Geolocation API die Position des Funkmasten bestimmen. Diese Informationen sind im SIB 1 zu finden. Welches in der nachfolgenden Abbildung gezeigt wird.

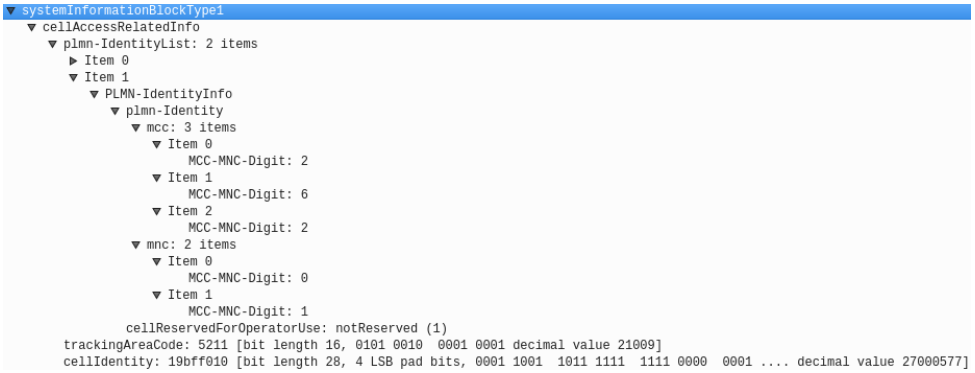

Abbildung 3.2.2: SIB 1

Folgende Werte können im SIB 1 ausgelesen werden:

- MCC: 262 -> Deutschland
- MNC: 1 -> Telekom
- TAC: 21009 (Der Tracking Area Code entspricht dem Location Area Code (LAC) bei GSM)
- CID: 27000577

Diese Werte können unter `http://opencellid.org` eingegeben werden, um den Standort zu ermitteln. Der nachfolgende GET-Request ergibt, dass der Standort des Mobilfunkmasten auf Breitengrad 48.105722 und Längengrad 11.535150 ist (München, Mittersendling). Dabei ist zu beachten wie die ausgelesenen Werte von SIB 1 an opencellid.org übergeben werden:

```
http://opencellid.org/#action=locations.cell&mcc=262&mnc=1&lac
    =21009&cellid=27000577
```

Auch die Paging Informationen sind von Interesse. Die Abbildung zeigt das in dem Paging Datenpaket die TIMSI kodiert ist.

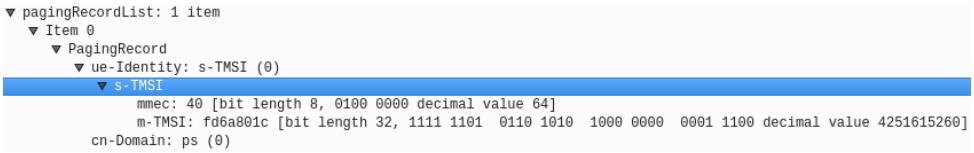

Abbildung 3.2.3: Paging TIMSI

3GPP Decoder

Der 3GPP Decoder ist ein Decoder der von 3GPP zur freien Verfügung bereitgestellt wird. Der 3GPP Decoder kann die meisten GSM, WCDMA und LTE Nachrichten sowie das IP-Protokoll über Wireshark dekodieren. Um eine Dekodierung durchzuführen wird die Nachricht als HEX-Stream eingegeben, sowie ein Protokoll ausgewählt. Dabei ist es wichtig, dass man im Dropdown-Menü einen Eintrag auswählt. Nach unserer Erfahrung ist es nicht wichtig das Richtige zu wählen, jedoch erleichtert es dem Dekoder schneller das Richtige zu wählen. Für eine erfolgreiche Dekodierung werden Administratorrechte benötigt. [3GP]

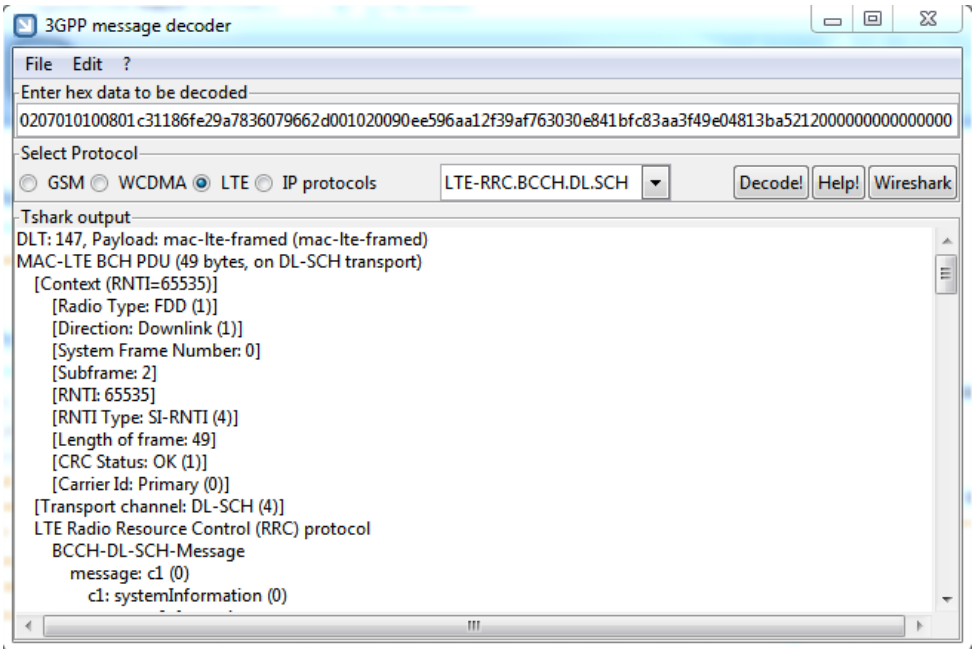

Abbildung 3.2.4: Zeigt den 3GPP Decoder mit einer decodierten Nachticht.

Vorteile:

- verwendbar über Kommandozeile
- dekodierter Text ist leicht durchsuchbar

Nachteile:

- ist nur für Windows verfügbar
- benötigt zusätzlich Wireshark

3.3 Master Information Block *(MIB)*

Das Auslesen der MIB ist der erste Schritt nach initialer Synchronisierung mit der Zelle. Wie in der Abbildung 3.4.1 zusehen ist wird in allen Radioframes als erstes ein MIB gesendet. Bei jeder System Frame Nummer die durch 4 teilbar ist wird eine neue MIB gesendet und bei den anderen Radioframes die alte MIB wiederholt. Die MIB enthält folgende Informationen:

- Bandbreite des Downlink-Kanals in Bezug auf die Ressourcenblöcke
- PHICH Konfiguration
- System Frame Nummer

3.4 System Information Block *(SIB)*

Mit Hilfe der SIBs teilt eine Zelle dem UE relevante Information mit, die dem UE helfen sich mit dem Netz zu verbinden und andere Informationen, wie zum Beispiel: INTRA-Frequenz, INTER-Frequenz und INTER-RAT Zellauswahl. In der nachfolgenden Tabelle 3.1 sind alle 19 SIBs zu finden, die für das LTE spezifiziert wurden. Jeder dieser SIB hat seine eigene Aufgabe, die von Zellen unterstützt werden müssen, deshalb treten nicht alle in einer Zelle auf, zum Beispiel SIB 18 wird für Device zu Device Kommunikation benötigt. Diese Funktionalität ist noch relativ neu und wird noch nicht von allen Zellen unterstützt. Um einen Verbindungsaufbau zu initiieren benötigt das UE die SIB1 und SIB2. Die SIBs werden im BCCH(Logical channel) -> DL-SCH(Transport channel) -> PDSCH(Physical channel) übertragen. Um den PDSCH zu lesen zu können benötigt man die Kontrollinformationen aus den PDCCH und hier kommt die MIB ins Spiel.

Da PDCCH, PHICH und PDFICH sich die Ressource in der Kontrollregion teilen, kann die UE durch das Wissen um die Konfiguration des PHICH und die fixe PDFICH Ressource, den Zugriff auf die PDCCH zu erhalten. Damit ist der UE möglich die PDSCH zu lesen und damit verbundene SIBs.

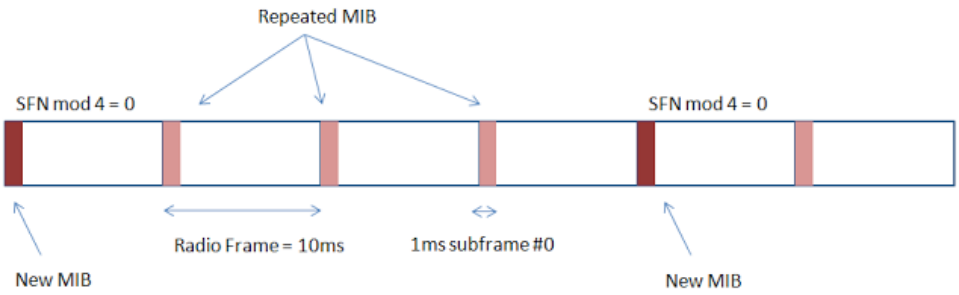

Abbildung 3.4.1: Zeigt das Vorkommen der MIBs im Radioframe. [sim]

3.4.1 Entdeckte SIBs

Im Rahmen unser Projekts konnten wir bei der Analyse der LTE-Zellen folgenden SIBs finden im Anhang A sind ein paar gefunden SIBs hinterlegt.:

Tabelle 3.1: SIBs

SIB	Beschreibung
SIB 1	Zellzugriffsparameter und Zeitplan für andere SIBs
SIB 2	Konfiguration von Common und Shared-Kanälen, inklusive RACH Konfiguration
SIB 3	Parameters für Zellen Re-Selektion
SIB 4	Information bezüglich Intra-Frequenz Nachbarzellen (E-UTRA)
SIB 5	Information bezüglich Inter-Frequenz Nachbarzellen (E-UTRA)
SIB 6	Information für Re-Selektion zu Inter-RAT (UTRAN Zellen)
SIB 7	Information für Re-Selektion zu Inter-RAT (GERAN Zellen)
SIB 8	Information für Re-Selektion zu Inter-RAT (CDMA2000)
SIB 9	Informationen im Zusammenhang mit Home eNodeB (FEMTOCELL)
SIB 10	ETWS Information (Primäre Benachrichtigung)
SIB 11	ETWS Information (Sekundäre Benachrichtigung)
SIB 12	Commercial Mobile Alert Service (CMAS) Information
SIB 13	Enthält die erforderlichen Informationen zu erlangen der MBMS-Steuerinformationen
SIB 14	Enthält die Extended Access Barring (EAB) Parametern
SIB 15	Enthält MBMS Service Area Identities (SAI)
SIB 16	GPS-Verwandte Informationen
SIB 17	WLAN Konfiguration für LTE-WLAN Zusammenarbeit
SIB 18	SID für LTE-D2D (Device-to-Device)
SIB 19	SID für LTE-D2D (Device-to-Device)

[Sata][ETS][www][Schb]

SIB 1 Diese SIB enthält die Daten für die Evaluation für den Zellzugriff und den Zeitplan für die andern SIBs. Dieser Frame wird im Subframe-Nummer 5 alle 80ms übertragen, wenn die SFN durch 8 teilbar ist. Es kann innerhalb von 80ms wiederholt werden. Dies geschieht im Subframe-Nummer 5 wenn SFN durch 2 dividierbar ist. Re-Selektion Parameter.[Satb]

SIB 2 Die SIB 2 beinhaltet die allgemeine Konfiguration für die Radioressourcen für die UEs sowie die Konfigurationen für Common und Shared Kanäle und RACH. Des weiteren Timers und UL Energie Kontrollen. Ohne SIB 2 könnte das UE kein *ATTACH* einleiten. Re-Selektion Parameter.[Satc]

SIB 3 Die SIB 3 behandelt ein Teil des Re-Selektions, hier bei werden Re-Selektions Informationen gemeinsam für Intra-Frequenz, Inter-Frequenz und Inter-RAT übermittelt. Dazu kommt noch Re-Selektions Information für Intra-Frequenz die nicht auf die benachbarten Zelle bezogen ist. Außerdem besitzt die SIB ein Feld in dem Inter-Frequenz und IRAT-Frequenz Messungen unter bestimmten Voraussetzungen anstoßen kann. Re-Selektion Parameter.[Satd]

SIB 5 Die SIB 5 dient ausschließlich nur für die Re-Selektion im Inter-Frequenz Bereich, das heißt es beinhaltet nur Informationen für andere E-UTRA-Frequenzen und benachbarte Inter-Frequenz Zellen. Re-Selektion Parameter.[Sate]

SIB 6 Die SIB 6 ist wie SIB 5, nur für Inter-RAT (UTRAN). Jedoch ohne Zellen spezifische Re-Selektion Parameter.[Satf]

Kapitel 4

Visualisierung

4.1 CellTracker

Der bisherige Prozess um LTE Funkzellen zu lokalisieren sieht wie folgt aus. Zuerst wird eine PCAP Datei mit Hilfe von Wireshark geöffnet und untersucht. Es werden die Informationen zur eindeutigen Identifizierung einer Funkzelle (MCC, MNC, TAC, CID) notiert und anschließend als Anfrage im Browser an eine der frei verfügbaren Funkzellendatenbanken gesendet. Als Ergebnis enthält man eine Karte auf der die Funkzelle als Marker eingetragen ist.

Um diesen Prozess zu automatisieren, haben wir eine Anwendung entwickelt, die alle nötigen Teilschritte beinhaltet. Ein Parser, der die PCAP Dateien einliest und die nötigen Informationen extrahiert, Schnittstellen zu den beiden größten Funkzellendatenbanken (Google und OpenCellID) um an die Koordinaten der Funkzellen zu gelangen, sowie eine Google Map Bibliothek zur grafischen Darstellung der Karte und Lokalisierung der gefundenen Funkzellen.

Der Vorteil unserer Anwendung ist, dass man mehrere PCAP Dateien auf einmal auswerten kann und alle gefundenen Ergebnisse auf der Karte vermerkt werden. Jeder Mobilfunkanbieter hat eine eigene Markerfarbe und ein eigenes Symbol, wodurch eine Zuordnung auf den ersten Blick möglich ist. Außerdem kann die Anwendung um weitere Parser und Funkzellendatenbanken erweitert werden, welche dann über ein Kommandozeilenargument ausgewählt werden können.

Implementierung

Zentrale Komponente der Anwendung ist eine JavaFX GUI, welche Kommandozeilenargumente zur Konfiguration entgegen nimmt und den gewünschten Parser, sowie die Schnittstelle zur Funkzellendatenbank erzeugt. Der Parser wiederum verarbeitet die gewählte PCAP Datei oder einen ganzen Ordner mit PCAP Dateien zu einer Liste aus Funkzellen. Diese Liste wird an die Schnittstelle der Funkzellendatenbank gesendet und um Koordinaten zur Funkzelle ergänzt. Im letzten Schritt wird für jede Funkzelle ein Marker und ein Informationsfenster auf einer Karte platziert, welches sich über einen Klick auf den Marker öffnet und die gesammelten Informationen der Funkzelle enthält.

Aufruf

Um den Anwendung starten zu können müssen bestimmte Argumente übergeben werden:

```
Usage: CellTracker.jar -p -f -l
-p Parser used to get list of cells
-f File to PCAP files/folder
-l Locator used to get cell coordinates
```

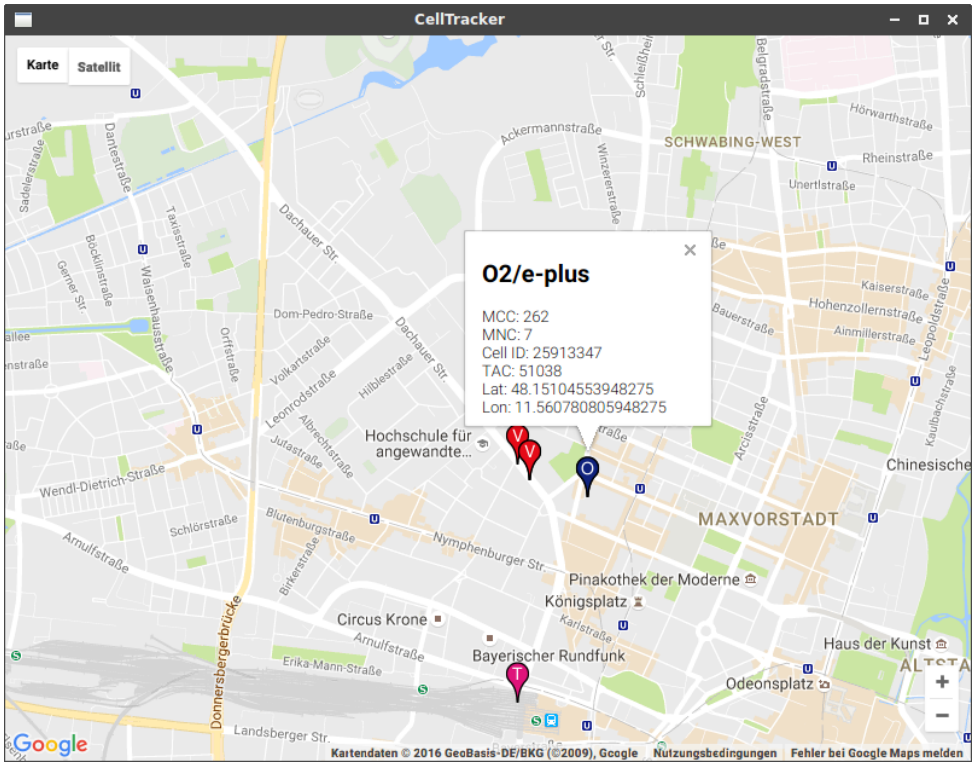

Abbildung 4.1.1: Zeigt gefundene Zellen in der Karte von Google Maps

Quellcode unter der GPL Lizenz auf GitHub veröffentlicht. Projektname: LTE-CellTracker. Link: `https://github.com/shreaker/LTE-CellTracker`

4.2 PCAP Parser

jNetPcap

Die Basis des Parsers bildet die jNetPcap-Bibliothek, welche als Java-Wrapper für die populäre libpcap-Bibliothek dient. Diese Software kann zur Erzeugung und Analyse von Netzwerkverkehr verwendet werden. jNetPcap ist dazu in der Lage aufgezeichnete Daten im „pcap“-Format von srsUE zu lesen und in einzelne Pakete aufzuschlüsseln.

Es war nicht möglich, die Paketinhalte mit jNetPcap zu analysieren, weshalb ein Dissector von Hand implementiert wurde. Zuerst müssen dabei die SIB1 Pakete von den restlichen

Daten gefiltert werden, was sich aber aufgrund ihres festen Headers als sehr einfach gestaltet. Die Nutzdaten im Paket selbst sind allerdings nicht anhand fester Bytegrenzen ausgerichtet, sondern orientieren sich auf Bitebene. Tabelle 4.1 zeigt die Daten am Anfang eines SIB1. Die erste Information ist die PLMN-Identity, welche aus einer variablen Anzahl von MCCs und MNCs besteht, wobei die MNCs ebenfalls von unterschiedlicher Länge sein können. Es wurde deshalb ein Ansatz gewählt, bei dem der ganze Block nacheinander gelesen wird und mit einem Zähler der aktuelle Offset vermerkt wird. Der nächste Eintrag ist der TAC, welcher eine feste Länge von 16 Bit besitzt und danach die CellIdentity mit 28 Bit. Sie liegen direkt im Anschluss an den PLMN-Block

PLMN-Identity	Consists of MCC and MNC. The first listed one is the Primary PLMN
trackingAreaCode	TAC which is common to the PLMN Identities
cellIdentity	Identifies a cell within the PLMN

Tabelle 4.1: SIB1 Daten

4.3 Lokalisierung von Zellen

Für die Standortsuche der LTE-Zellen, haben wir drei Wege gefunden, die Senderliste München, OpenCellID und Google Geolocation. Im Rahmen unserer CellTracker-Umsetzung haben wir zwei dieser Wege eingebunden, dabei haben wir festgestellt, dass nicht jede Zelle in jeder Zellenliste zu gefunden wird.

4.3.1 Senderlistemuc

Die Senderliste München ermittelt Funkzellen im Großraum München, hierbei werden visuell die Sendeanlagen gesucht und die Postion als Postadresse(PLZ, Straße, Hausnr.) gespeichert. Die Liste der Zellen können für die offline Auswertung heruntergeladen werden. [Scha]

4.3.2 OpenCellID

OpenCellID.org ist eine freie Datenbank für GSM-, UMTS- und LTE-Zellen. Sie bietet eine API-Schnittstelle um automatisiert Zellen abzufragen, sowie Messdaten zum Projekt einzureichen. Für einige Funktionen der API wird ein Schlüssel benötigt, den man kostenlos beantragen kann. Jedoch existiert für die Benutzung eine Richtlinie, diese beinhaltet das man für eine langfristige Benutzung der API nicht nur einseitig die Zellen abfragten sondern Messdaten von Zellen beisteuern muss. Im Rahmen unser Projekts haben wir nur die Funktion *Getting cell position* zur Positionsbestimmung unserer gefundenen Zellen benutzt, für eine langfristige Nutzung müssten wir die Richtlinie einhalten. Diese Methode wird über ein GET-Request realisiert, dabei kann das Antwortformat gewählt werden, entweder xml oder json.[Ope]

4.3.3 Google Geolocation

Die Google Geolocation API besteht aus einer einzigen POST-Request, die einen Json beinhaltet. Hierzu wird ebenfalls ein API-Schlüssel benötigt, denn man kostenlos beantragen kann. Bei der kostenlosen Version ist die Anzahl der Anfragen limitiert auf 2500 pro Tag und 50 pro Sekunde. Wir haben die Google Geolocation API neben der OpenCellID API in unser Projekt eingebunden, dabei haben wir mit der Google Geolocation API Zellen gefunden, die OpenCellID nicht hatte. [Goob][Gooa]

Kapitel 5

Fazit

In diesem Kapitel möchten wir unsere persönlichen Erfahrungen die wir im Laufe des Projekts gemacht haben teilen.

Die praktischen Erfahrung mit LTE haben zum Verständnis der Vorlesung Mobile Netze beigetragen und uns geholfen das erlernte Wissen einordnen und anwenden zu können.

Dadurch dass die verwendeten Open Source Werkzeuge/Frameworks noch sehr jung sind und nur wenige Entwickler daran beteiligt, war auch so gut wie keine Dokumentation vorhanden. Das stellte uns vor eine Herausforderung, da wir es gewohnt waren in die Dokumentation zu sehen wenn wir einmal nicht weiter wussten. Wir erlernten so den Umgang mit Mailinglisten und haben unsere Antworten dann auf anderem Wege erhalten. Das konnte dann aber auch schon einmal mehrere Tage in Anspruch nehmen. Geduld war gefragt.

Als wir das Wissen zusammen getragen hatten um die Software in Betrieb zu nehmen, war die Freude über die ersten Messergebnisse umso größer. Schließlich war das kein leicht umkämpfter Sieg. Angefangen bei der Auswahl des geeigneten Betriebssystems, den bestehenden Abhängigkeiten der einzelnen Pakete bis hin zum stundenlangen kompilieren der Software, was anfangs oft mit einem Fehler endete.

Wenn wir das Projekt ein weiteres Mal umsetzen müssten, würden wir vieles anders machen. Wir würden mehr Vorüberlegungen treffen und unser Vorgehen genauer planen und besser strukturieren. Durch den gezielten Einsatz der Mailinglisten wären wir schneller in der Lage an erste Ergebnisse zu gelangen und hätten mehr Zeit zur Durchführung.

Alles in allem war das Projekt für uns ein Erfolg und hat dazu beigetragen dass wir uns weiterhin mit den verwendeten Komponenten auseinander setzen möchten.

Literaturverzeichnis

[3GP] 3GPP: *3GPP decoder.* – http://3gppdecoder.free.fr/?q=node/1; 19. September 2016.

[anb16] ANBIETER.INFO lte: *LTE Frequenzen*, 2016. http://www.lte-anbieter.info/ratgeber/frequenzen-lte.php

[ETS] ETSI: *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (3GPP TS 36.331 version 11.1.0 Release 11).* – http://www.etsi.org/deliver/etsi_ts/136300_136399/136331/11.01.00_60/ts_136331v110100p.pdf;

[Gooa] GOOGLE: *Google Maps Geocoding API Usage Limits.* – https://developers.google.com/maps/documentation/geocoding/usage-limits?hl=de;

[Goob] GOOGLE: *Die Google Maps Geolocation API.* – https://developers.google.com/maps/documentation/geolocation/intro?hl=de;

[Ope] OPENCELLID: *API.* – http://wiki.opencellid.org/wiki/API;

[Sata] SATPATHY, Arijit: *All about SIB's in LTE.* – http://lteinwireless.blogspot.de/2011/06/all-about-sibs-in-lte.html;

[Satb] SATPATHY, Arijit: *SIB1 in LTE.* – http://lteinwireless.blogspot.de/2011/06/sib1-in-lte.html;

[Satc] SATPATHY, Arijit: *SIB2 in LTE.* – http://lteinwireless.blogspot.de/2012/12/sib2-in-lte.html;

[Satd] SATPATHY, Arijit: *SIB3 in LTE.* – http://lteinwireless.blogspot.de/2011/07/sib3-in-lte.html;

[Sate] SATPATHY, Arijit: *SIB5 in LTE.* – http://lteinwireless.blogspot.de/2011/08/sib5-in-lte.html;

[Satf] SATPATHY, Arijit: *SIB6 in LTE.* – http://lteinwireless.blogspot.de/2011/07/sib6-in-lte.html;

[Scha] SCHARL, Philipp: *Senderliste München.* – http://www.senderlistemuc.de/;

[Schb] SCHWARZ, Rohde : *LTE-Advanced (3GPP Rel.12)Technology Introduction White Paper.* – https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_

application/application_notes/1ma252/1MA252_2e_LTE_Rel12_technology.pdf;

[sim] SIMPLETECHPOST: *Master Information Block (MIB) in LTE*. – http://www.simpletechpost.com/2012/06/master-information-block-mib-in-lte.html;

[sim16] *All about Wired and Wireless Technology*. http://www.simpletechpost.com/2012/06/rimary-and-secondary-synchronization.html. Version: 2016

[sma16] SMARTCHECKER.DE: *Mobilfunkdatenraten*, 2016. http://www.smartchecker.de/ratgeber/lte-umts-und-co-die-ubertragungsstandards/4178

[www] WWW.SHARETECHNOTE.COM: *LTE Quick Reference - SIB(System Information Block)*. – http://www.sharetechnote.com/html/Handbook_LTE_SIB.html;

Anhang A

Erhaltene SIB 1-3,5,6

Abbildung A.0.1: SIB 1, 5 und 6

SIB 1

PLMN-Identity	Consists of MCC and MNC. The first listed one is the Primary PLMN	
	MCC	262
	MNC	2
p-Max	Value applicable for the cell. If absent the UE applies the maximum power according to the UE capability. If eNB configures the value more than the value supported by the UE then UE will set the max value aupported by the UE capability. Example UE Catagory 3 supports max 23 db	
cellReservedForOperatorUse	As defined by operator (Reserved/Not_reserved)	not reserved
trackingAreaCode	TAC which is common to the PLMN Identities	bba3
cellIdentity	Identifies a cell within the PLMN	141df030
cellBarred	If Barred then UE is not allowed to camp on the cell	notBarred
intraFreqReselection	If enabled, UE will be able to perform Cell-reselection to INTRA-frequency cells	allowed
q_RxLevMin	Minimum required RX level in the cell	
q_RxLevMinOffset	Actual value Qrxlevminoffset = IE value * 2 [dB]. only applied when a cell is evaluated for cell selection as a result of a periodic search for a higher priority PLMN while camped normally in a VPLMN [5]. During this periodic search for higher priority PLMN the UE may check the S criteria of a cell using parameter values stored from a different cell of this higher priority PLMN. Affects the minimum required Rx level in the cell.	
freqBandIndicator	indicates the E-UTRA operating band	
schedulingInfoList	information regarding the presence of SIB type; other than SIB1	
si_Periodicity	Periodicity of the SI-message in radio frames (SI will be transmitted within the specified radio frame)	
sib_MappingInfo	carries the List of the SIBs mapped. SIB2 is always present in the first element of schedulingInfoList	
si_WindowLength	specifies that a SIB should be transmitted somewhere within the specified window length. Value is in ms. This window starts at the starting sub-frame of the mentioned si_periodicity. SIBs can be received in any of the sub-frame as mentioned in the WindowLength.	
systemInfoValueTag	indicates if a change has occurred in the SI messages. UEs may use systemInfoValueTag, e.g. upon return from out of coverage, to verify if the previously stored SI messages are still valid. Additionally, the UE considers stored system information to be invalid after 3 hours from the moment it was successfully confirmed as valid, unless specified otherwise. Common for all SIBs other than MIB, SIB1, SIB10, SIB11 and SIB12.	

SIB 2

ac-BarringInfo	Access Class Barring configuration	
radioResourceConfig	used to specify common radio resource configurations in the system information and in the mobility control information	
numberOfRA_Preambles	Number of non-dedicated random access preambles	n16 (3)
preamblesGroupAConfig_exist	Provides the configuration for preamble grouping. If the field is not signalled, the size of the random access preambles group A is equal to numberOfRA-Preambles	
powerRampingParameters		
powerRampingStep	Power ramping factor	dB2 (1)
preambleInitialReceivedTargetPower	Initial preamble power	dBm-104 (8)
ra_SupervisionInfo		
preambleTransMax	Maximum number of preamble transmission	n10 (6)
ra_ResponseWindowSize	Duration of the RA response window. Value in subframes. Value sf2 corresponds to 2 subframes	sf10 (7)
mac_ContentionResolutionTimer	Timer for contention resolution. Value in subframes. Value sf8 corresponds to 8 subframes	sf64 (7)
maxHARQ_Msg3Tx 4	Maximum number of Msg3 HARQ transmissions, used for contention based random access	5
bcch_Config		
modificationPeriodCoeff	Actual modification period, expressed in number of radio frames= modificationPeriodCoeff * defaultPagingCycle. n2 corresponds to value 2	n4 (1)

pcch_Config		
defaultPagingCycle	Default paging cycle, used to derive 'T'. Value rf32 corresponds to 32 radio frames	rf64 (1)
nB	is used as one of parameters to derive the Paging Frame and Paging Occasion	twoT (1)
prach_Config	used to specify the PRACH configuration in the system information and in the mobility control information	
rootSequenceIndex		650
prach_ConfigIndex	**mentions the:** Preamble format: 0-4 (For frame structure 1 preamble format is 0-3 and for frame structure 2 it is 0-4) SFN: whether it will be EVEN no. frmae OR any frame subframe number: carrier the subframe no. within the SFN (Look for the table in TS36.211 - Table 5.7.1-2)	35
highSpeedFlag	TRUE corresponds to Restricted set and FALSE to Unrestricted set	FALSCH
zeroCorrelationZoneConfig	used for Preamble generation	14
pdsch_Config	used to specify the common and the UE specific PDSCH configuration	
	referenceSignalPower	15dbm
pusch_Config	used to specify the common PUSCH configuration and the reference signal configuration for PUSCH and PUCCH	
pucch_Config	used to specify the common and the UE specific PUCCH configuration	
soundingRS_UL_Config	used to specify the uplink Sounding RS configuration	release: NULL
uplinkPowerControl	used to specify parameters for uplink power control in the system information and in the dedicated signalling	
ue_TimersAndConstants	Timer values	
freqInfo	UL carrier frequency and bandwidth	
timeAlignmentTimerCommon	used to control how long the UE is considered uplink time aligned. Value in subframes	sf1920 (3)

SIB 3

cellReselectionInfoCommon	Cell re-selection information common for cells	
q_Hyst	This specifies the hysteresis value for cell re-selection ranking criteria. The specified value is added to the serving cell RSRP measurement.	dB4 (4)
speedStateReselectionPars	Speed dependent reselection parameters. If this field is absent then mobility State Parameters is not present. If q_hystSF is present then it is added to q_hyst. Carrier (-) value so as to reduce the ranking of the serving cell and allows cellreselection to occur easily.	
cellReselectionServingFreqInfo	Information for Cell re-selection to inter-frequency and inter-RAT cells	
s_NonIntraSearch	#used to trigger Interfrequency and IRATfrequency measurements for cell reselection when: - TARGET Interfreq. has lower or equal priority - IRAT freq. has a lower priority # specifies the Srxlev threshold (in dB) for E-UTRAN inter-frequency and inter-RAT measurements. #If the field s-NonIntraSearchP is present, the UE applies the value of s-NonIntraSearchP. # Otherwise if neither s-NonIntraSearch nor s-NonIntraSearchP is present, the UE applies the (default) value of infinity for SnonIntraSearchP.	8db (4)
threshServingLow	This specifies the Srxlev threshold (in dB) below which the serving cell must fall before reselecting towards a lower priority RAT/ frequency. Value in between 0-31dB. Actual value=signalled value*2	6db (3)
cellReselectionPriority	explains the absolute priority of the concerned serving cell frequency /set of frequencies (GERAN)/ bandclass (CDMA2000), as used by the cell reselection procedure. Corresponds with parameter "priority". Value is between 0-7 where 0 means: lowest priority.	5
intraFreqCellReselectionInfo	Cell re-selection information common for intra-frequency cells	-126dBm **(**-63**)**
q_RxLevMin	Minimum required RSRP for cell reselection. value -70 - 22 dbm, Actual value=signalled value*2	-126dBm **(**-63**)**
s_IntraSearch	# used to trigger intrafreq. measurements # specifies the Srxlev threshold (in dB) for intra-frequency measurements. # If the field s-IntraSearchP is present, the UE applies the value of s-IntraSearchP instead. Otherwise if neither 09s-IntraSearch nor s-IntraSearchP is present, the UE applies the (default) value of infinity for SIntraSearchP.	58dB (29)

presenceAntennaPort1	is used to09 indicate whether all the neighbouring cells use 09Antenna Port 1. When set to TRUE, the UE may assume that at least two cell-specific antenna ports are used in all neighbouring cells.	
neighCellConfig	used to provide the information related to MBSFN and TDD UL/DL configuration of neighbour cells.	No MBSFN subframes are in all neighbours
t_ReselectionEUTRA	# specifies the cell reselection timer value # defines time to trigger for cell reselection # The parameter can be set per E-UTRAN frequency	1s
p_Max	# max. allowed UL transmit power for intra-frequency neighbouring E-UTRA cells. If absent the UE applies the maximum power according to the UE capability	