University POLITEHNICA of Bucharest

Faculty of Automatic Control and Computers,
Computer Science and Engineering Department

BACHELOR THESIS

Validating and Profiling Physical Layer in 4G Base Stations

Scientific Adviser:

Conf. Dr. Ing. Răzvan Rughiniș
As. Drd. Ing. Daniel Rosner
Ing. Mihnea Ionescu

Author:

Mihai Bărbulescu

Bucharest, 2014

Universitatea POLITEHNICA din București

Facultatea de Automatică și Calculatoare,
Departamentul de Calculatoare

LUCRARE DE DIPLOMĂ

Validarea și monitorizarea nivelului fizic în stațiile de bază 4G

Conducător Științific:
Conf. Dr. Ing. Răzvan Rughiniș
As. Drd. Ing. Daniel Rosner
Ing. Mihnea Ionescu

Autor:
Mihai Bărbulescu

București, 2014

First and foremost, I have to thank my colleagues from Freescale Semiconductor Romania, S.l. Dr. Ing. Andrei Alexandru Enescu, Mr. Mihnea Ionescu, Mr. Cristian Şavlovschi and Mr. Victor Popescu. Without their assistance and dedicated involvement in every step throughout the process, this paper would have never been accomplished.

I would also like to thank to my thesis supervisors, As. Drd. Daniel Rosner and Conf. Dr. Ing. Răzvan Rughiniş, for their help and patience.

Abstract

A base station is a very complex equipment, containing several hardware and software modules, such as service protocol layers, operating systems, digital signal processors, general purpose processors, hardware accelerators etc.

This paper focuses on presenting a generic software solution for testing and monitoring how physical layer (Layer 1) behaves and also if it operates in conformance to standard specifications. The main purpose of the testing solution is to isolate Layer 1 from the other OSI (Open Systems Interconnection model) Layers and other software modules, in order to eliminate additional levels of complexity that they might cause, thus using a black-box approach.

Also, once a test session ends, profiling is needed for obtaining Layer 1 system information, such as processor usage or memory load and overview of the radio communication with a dedicated test mobile. Such information should be aggregated for an entire test session, which contains multiple test cases, for fast evaluation of the overall system performance and possible causes for poor performance of the wireless communication.

The case study of this testing and profiling framework is the LTE Physical Layer software solution deployed on Freescale® BSC913x small cells solution.

Keywords: communication systems, base station, small cells, BlackBox testing, Physical Layer, system analysis, profiling.

Contents

List of Figures

Notations and Abbreviations

3GPP	3rd Generation Partnership Project
ACK	Acknowledgment
API	Application Programming Interface
AWGN	Additive White Gaussian Noise
BCH	Broadcast Channel
BER	Bit Error Rate
BSP	Board Support Package
CQI	Channel Quality Indicator
CRC	Cyclic Redundancy Check
DCI	Downlink Control Information
DFT	Discrete Fourier Transform
DLSCH	Downlink Shared Channel
DNS	Domain Name Service
DSP	Digital Signal Processor
DUT	Device Under Test
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
EPC	Enhanced Packet Core
FAPI	Femto Forum Application Platform Interface
FDD	Frequency Division Duplex
FFT	Fast Fourier Transform
HARQ	Hybrid Automatic Repeat Request
IPC	Inter Processor Communication
LOS	Line of Sight
MAC	Media Access Control
MCH	Multicast Channel
MIB	Master Information Block
MIMO	Multiple Input Multiple Output
MME	Mobility Management Entity
NACK	Not Acknowledgment
NLOS	Non Line of Sight
OS	Operating System

OSI	Open Systems Interconnection model
P-GW	Packet Data Network Gateway
PCH	Paging Channel
PCRF	Policy and Charging Rules Function
PDU	Protocol Data Unit
PHICH	Physical HARQ Indication Channel
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RACH	Random Access Channel
RLC	Radio Link Control
RNTI	Radio Network Temporary Identifier
S-GW	Serving Gateway
SF	Subframe Number
SFN	System Frame Number
SIB	System Information Block
SNR	Signal to Noise Ratio
SRS	Sounding Reference Signal
TCP	Transmission Control Protocol
TDD	Time Division Duplex
ULSCH	Uplink Shared Channel

Chapter 1

Introduction

1.1 Context and Objectives

Context

The wireless communication technologies are in a continuous rise and there is a quick growth of the interest in radio access technologies for providing mobile service such as voice, data and video streaming. One important element in enabling mobile devices to have Internet access and possibility for sending and receiving calls is a base station, which enables radio access for the mobile devices.

In an enterprise context, the commercial software written for a base station can come from different providers, Layer 1 solution can come from one provider, Layer 2-3 solution (MAC and Network) can come from other third parties, OS and BSP could also come from a different sources etc. Every component of a base station has it's own requirements and challenges, thus arises the need to have a testing solution which can isolate one component from another.

Testers should have access to providing stimulus messages for ensuring proper working according to standard and client requirements of the Physical Layer Software. [MB14]

Objectives

This thesis presents a **generic automated testing framework for validating and profiling the LTE Physical Layer (Layer 1) software** deployed on Freescale® BSC913x (BSC9131® and BSC9132®) platforms which target the small cells solutions.

Objective of this testing framework is to provide remote access to an emulated higher layer logic developed on the device under test (DUT), BSC913x platforms, to a third party test mobile, which emulates the behavior of regular subscribers in a mobile network and offers access to controlling messages sent by any layer and to other additional modules required for automation (databases, additional PCs, scripts etc.).

In the end, users of the testing solution would only use a simple web interface for triggering execution of multiple test scenarios which are mapped to various scripts that control the base station and test mobile. Main purpose of this testing framework is to find faulty behavior of the Physical Layer software and the root cause of the faulty behavior, which might be: invalid higher layer configuration messages sent to physical layer, hardware issues or limitations or errors in Physical Layer software (such as invalid memory access or module integration interaction not entirely covered).

Moreover, as multiple tests can generate a large amount of logs, a set of scripts parsing content of this logs is required in order to generate **aggregated statistics** for easier debugging and easier determination of the root cause of the faulty behavior.

1.2 Motivation for BlackBox Testing

Although a full protocol (or full stack) testing is more close to a real life scenario, it is not suitable for detecting errors at the physical layer. How can one know for sure that a failed attachment of a mobile terminal to the network has root cause in a problem at the physical layer and not a problem at the network layer?

In order to validate a physical layer software solution, running on an embedded platform, it is necessary to strip Higher Layer entities to their minimal functionality, sometimes, even discarding them, if they are not necessary in the context of Layer 1 validation. This thesis describes a testing infrastructure for the Freescale's Layer 1 software.

BlackBox Testing consists of a custom Layer 2 emulator, a third party test mobile, which also offers directly access to Layer 2 and other additional dedicated equipments, such as a Test Execution PC for running test cases, a Web Server for displaying test results and trigger jobs containing multiple test cases, a storage for all the logs generated by the tests and a database to store the results. The motivation behind building an emulated Layer 2 is to discard Higher Layer entities, in order do better distinguish Layer 1 faulty behavior.

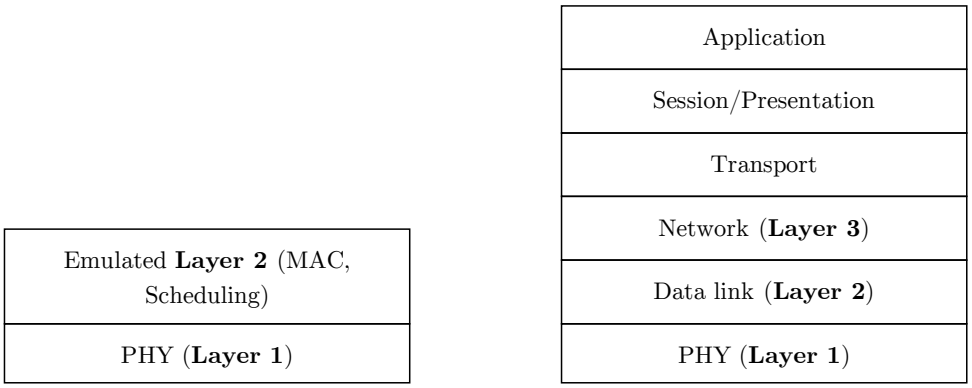


Figure 1.1: BlackBox Testing (left) versus Full Stack Testing (right)

Author's contribution was mainly on implementing testing infrastructure: having both solutions Freescale Layer 1 Software and emulated Layer 2 precompiled, what support should be added in either sides in order to achieve full automation. Briefly, this testing infrastructure (detailed in chapter 3) consists of:

- A web interface used to trigger execution of tests and used for displaying results
- A database to store test results and a job queue
- A test execution PC on which the user places his test scenarios (if he wants to have custom scenarios, and not use the predefined ones from the web interface)
- A private network in which the test execution PC, the UE emulator (a dedicated, third party test mobile) and the board with the emulated Layer 2 are placed for easy remote access. From any PC connected to this network anyone can reset, power on/off or send specific commands to the equipments. The web interface contains links to test scripts which do all these tasks in background, transparent for the user.

The final result is that users only have to click a button in the web interface to run the tests. They only care about the inputs that should be provided to the emulated Layer 2 and to the dedicated test mobile (not a mobile phone or a tablet). Probable causes of failure in a test are, consequently, reduced to:

- The custom Layer 2 solution
- The testing environment (or setup)
- Hardware issues
- Software issues in Layer 1 software solution

- Test mobile issues

1.3 Motivation for a Trace Visualizer

A test case ran with BlackBox can generate multiple logs, such as: the standard output of the board, test execution log (standard output of a test case), traffic capture generated by a packet sniffer, emulated Layer 2 standard output etc. A test case generates, in average, about 10-15 MB of log files. Usually a test session contains about 100 test cases, but this number varies depending on requests from developers or clients of running different test cases with different parameters. This means that at the end of a test session, one has to study about 1-1.5 GB of logs. Is all this data relevant for the developer or the client? Can either of them easily find the result he was looking for?

The Trace Visualizer is a set of scripts which focuses on logs generated by BlackBox Testing that offer a comprehensive view about the system (memory load, processor load for a given job, hardware accelerator load etc.) and also functional information, such as Layer 2 scheduling, retransmissions, transport block size and coding, packet loss, CRC (Cyclic Redundancy Check) errors etc.

Author's contribution was mainly in developing support for studying functional information, based especially on extracting messages exchanged in the communication between Layer 1 and the emulated Layer 2, which follows a standard, called Femto Forum Application Platform Interface (FAPI), described in detail in chapter 3.

Next chapter introduces some basic concepts about radio communication and LTE network radio access and protocols, highlighting:

- What parameters should one follow if measurements of radio communication performance are required and what errors can be simulated in a test and what solutions are currently used in base stations or in mobile terminals
- Why isolation from other parts that could have faulty behavior is important in testing only the Physical Layer
- What are the most important concepts that helps the user of testing solution who reads logs to find fast a root cause of a test failure or the root cause of an inconsistent behavior.
- What are the concepts used in the Trace Visualizer solution to filter results from logs and display them as pictures

Chapter 2

Overview of Radio Channel and LTE Network

2.1 Radio Channel and Physical Layer Challenges

This section contains a brief baseband theory, which highlights the most important parameters that should be extracted from a base station regarding the radio link in order to distinguish the hardware and software issues of the base station from the limitations induced by the wireless channel. Purpose of the below theory is to offer the reader a perspective about the most important concepts of wireless communication that should be known when reading large amount of data in logs generated by a testing solution of a base station and deciding which parameters offer relevant information, such as: channel quality or channel estimation, challenges in wireless communication and solutions used to attenuate different problems.

Content is based on the online course *"Advanced 3G and 4G Wireless Mobile Communications"* held by prof. Aditya K. Jagannatham [Jag13], lectures 1-12, Andrea Goldsmith's book on wireless communications [Gol05] and Jochen Schiller's book on mobile communications [Sch03], in which a summary of the most important concepts relevant for this thesis is made.

2.1.1 Multipath propagation of radio waves

Multi-propagation path refers to the fact that the UE (or the mobile station) will receive multiple radio waves, one of the most severe radio channel impairments. This is because in the environment there are trees, buildings, or moving objects (and also movement of the UE) which cause the apparition of reflected waves, which are called scatter components.

The direct wave, or the direct path from the base station (considered transmitter) to the UE (considered receiver) is called LOS (Line of Sight) . The scatter components are called NLOS (Non Line of Sight) :

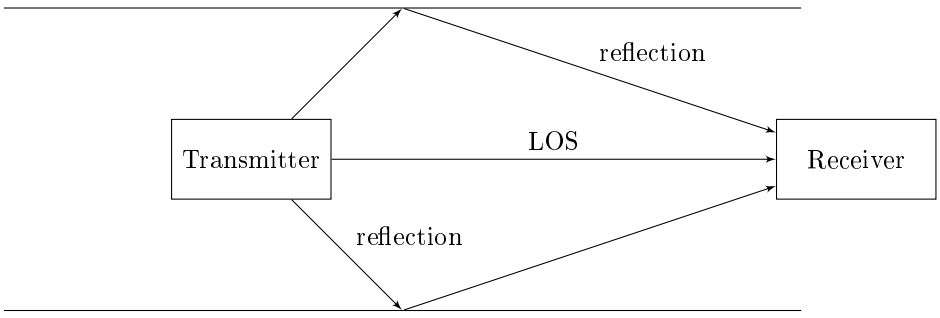


Figure 2.1: Line of Sight and reflections of wireless signal

Radio signals propagate just as light does: they follow a straight line - LOS - besides gravitational effects.

The effect of the multipath propagation is called **delay spread** the original signal is spread due to different delays of parts of the signal. It is important to note that this is not an effect of the possible movements of the sender or receiver. Typical values for delay spread are:

- 1 μs in indoor environment (Normal cyclic prefix [1] of 5 μs in LTE)
- 3 μs in city
- 15 μs in rural environment (GSM tolerates a delay spread of 16 μs, in LTE there is Extended cyclic prefix having a value of 17 μs, used for large cells, that are deployed in rural areas)

The effects of this delay spread on the signals representing the transmitted data are:

1. A short impulse will be smeared out into a broader impulse, or rather into several weaker impulses. A single transmitted impulse will result in many weaker impulses at the receiver. Each path has a different attenuation and, the individual received pulses have different power due to difference in distance traveled or nature of reflectors. The weaker impulses may appear as noise.
2. **Inter-symbol interference:** considering that every impulse is a symbol and that one or more bits of information represent one symbol: if sender has separated impulses, the receiver will have interference of impulses (overlap in time). Each impulse represents a symbol and one ore more symbols represent information. Inter-symbol interference refers to the energy needed to receive one symbol overlaps over the adjacent symbol. If the symbol rate is higher, this effect is more pronounced. Because of this effect, the bandwidth of a radio channel with mulipath propagation is limited, thus resulting in errors at the receiver. [Sch03, p. 40]

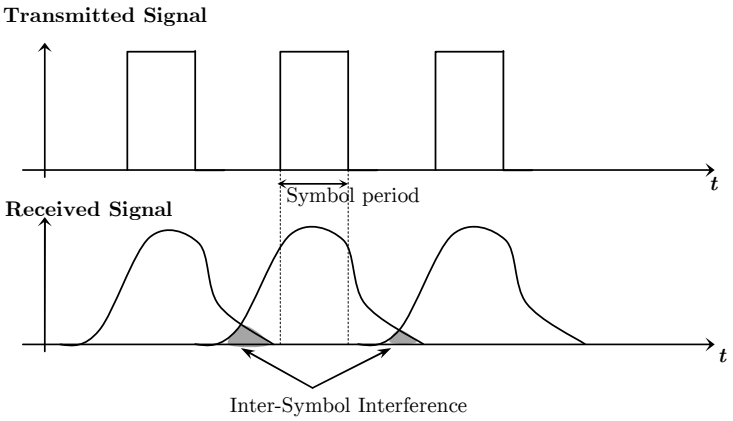


Figure 2.2: Inter-Symbol Interference

3. If the sender or the receiver will be moving, then another effect influencing radio transmission occurs: Doppler Shift. [Sch03, p. 40]

In LTE, in frequency domain, the subcarriers use a 15 kHz spacing for uplink and downlink traffic, which is considered a good balance for the overhead introduced by the cyclic prefix used to mitigate inter-symbol interference and Doppler Shifts.

In conclusion, we can say that multipath propagation limits the maximum bandwidth (property of medium) due to inter-symbol interference and moving transceivers cause additional problems due to varying channel characteristics.

In order to be able to mitigate these undesired effects the channel needs to be modeled. If the receiver has information about delays of different paths, it can compensate the distortion, by programming an equalizer, for instance. Section 2.1.2.7 describes some equalization techniques that can be used at the receiver.

[1] Cyclic Prefix is a guard period transmitted before each symbol (the number of bits represented by a particular phase, frequency or amplitude of a digital signal) in each subcarrier. LTE has three types of cyclic prefix with different value: short, normal and extended.

2.1.2 Mathematical models of wireless communications

In order to see exactly how multipath propagation affects wireless transmission, a mathematical model is needed. This could be extremely useful in order to determine a **channel estimation**, because if the receiver knows some values of the delays of the different paths it can compensate for the distortion caused by the channel.

We can model the Wireless Communication (or the Wireless Propagation Environment) as a system with impulse response $h(t)$:

in which:

- $x(t)$ - transmitted (TX) signal
- $h(t)$ - wireless channel (a linear time variant system)
- $y(t)$ - received (RX) signal

2.1.2.1 Multipah Propagation Model

In a multipath propagation environment, each path produces a copy of the original signal, which is attenuated by the scatter components. The attenuation is simply a scaling of the signal. There is also a delay, modeled as impulse function. Each path will be:

$$p_i = a_i \delta(t - \tau_i) \tag{2.1}$$

Radio channel is a multipath environment, so if there are L paths, the transfer function in time $h(t)$ will be the sum of the multipath components:

$$h(t) = \sum_{i=0}^{L-1} p_i = \sum_{i=0}^{L-1} a_i \delta(t - \tau_i) \tag{2.2}$$

Let $i = 0$ be the LOS (Line of Sight). The other components of the wireless channel will be the NLOS.

2.1.2.2 Wireless Signal Transmitted by the Base Station

The **wireless signal transmitted by the base station** is a passband signal that is transmitted at a carrier frequency (denoted f_c):

$$s(t) = Re\{s_b(t) e^{j2\pi f_c t}\} \tag{2.3}$$

where $s_b(t)$ is the complex baseband equivalent of the passband signal $s(t)$. The carrier frequency is allocated to that signal by the network operator.

The **received wireless signal at UE** will be a convolution between the transmitted signal and the channel (or transfer function) in complex baseband representation:

$$y(t) = s(t) * h(t) \tag{2.4}$$

The UE will receive from each path:

$$y_i(t) = Re\{a_i s_b(t - \tau_i) e^{j2\pi f_c (t - \tau_i)}\} \tag{2.5}$$

and from the summation of all L paths we obtain:

$$y(t) = \sum_{i=0}^{L-1} y_i(t) = Re\left\{\left\{\sum_{i=0}^{L-1} a_i s_b(t-\tau_i)e^{-j2\pi f_c \tau_i}\right\} e^{j2\pi f_c t}\right\} \tag{2.6}$$

This signal has two components:

- Complex baseband received signal: $y_b(t) = \sum_{i=0}^{L-1} a_i s_b(t-\tau_i)e^{-j2\pi f_c \tau_i}$ - which says what attenuation, delay and phase the signal has.
- Carrier term: $e^{j2\pi f_c t}$

2.1.2.3 Narrowband signal. Interference

A **narrowband transmitted signal** has a bandwidth which will never exceed its channel's approximate maximum bandwidth in a significant way. For such signal we can say that: $s_b(t-\tau_i) \approx s_b(t)$. Thus, the received signal will be the transmitted baseband signal multiplied (scaled) by complex phase factor (or complex coefficient).

$$y_b(t) = s_b(t) \sum_{i=0}^{L-1} a_i e^{j2\pi f_c \tau_i} \tag{2.7}$$

Depending on values of attenuations $(a_i, i = \overline{0, L-1})$ and delays $(\tau_i, i = \overline{0, L-1})$ these different complex numbers can result in a constructive or destructive interference.

- Let $L = 2, a_0 = 1, a_1 = 1, \tau_0 = 0, \tau_1 = \frac{1}{2f_c}$. The complex coefficient factor will be $h = \sum_{i=0}^{1} a_i e^{j2\pi f_c \tau_i} = 1 + e^{-\pi j} = 0$. So for two paths having the same attenuation but different delays the received signal is 0. There is no reception!
- Let $L = 2, a_0 = 1, a_1 = 1, \tau_0 = 0, \tau_1 = \frac{1}{f_c}$. Then $h = \sum_{i=0}^{1} a_i e^{j2\pi f_c \tau_i} = 2s_b(t)$. The two paths add up constructively in phase and the amplitude of the received signal is doubled.

2.1.2.4 Fading. Statistics

It is important to know how the signal's power varies due to multipath propagation (multipath induced fading) or due to obstacles which shadow the radio waves (**shadow fading**).

Fading is variation of the attenuation affecting a signal over certain propagation media. Fading varies with many components: time, frequency, geographical position, moving objects (which cause Doppler shifts), number of paths. Hence, you will generally see fading modeled using random variables.

In an ideal system (no constructive or destructive interference) we have a single path of propagation (which is exactly the LOS) so generally the received signal is exactly the input $y_b(t) = s_b(t)$.

In a wireless channel we saw that, for a narrowband signal, $y_b(t) = hs_b(t)$. Here, h represents the complex fading coefficient, which depended on attenuations and delays of the paths:

$$h = \sum_{i=0}^{L-1} a_i e^{j2\pi f_c \tau_i} = x + jy = ae^{j\phi} \tag{2.8}$$

In order to statistically describe the complex coefficient, it will be assumed that x, y are Gaussian in nature. Both are a sum of large number of random components (as mentioned above) depending on the scenario. Fading will be thus modeled as a normal (Gaussian) distribution, as fading varies with multiple components in a scenario and many of them are independent.

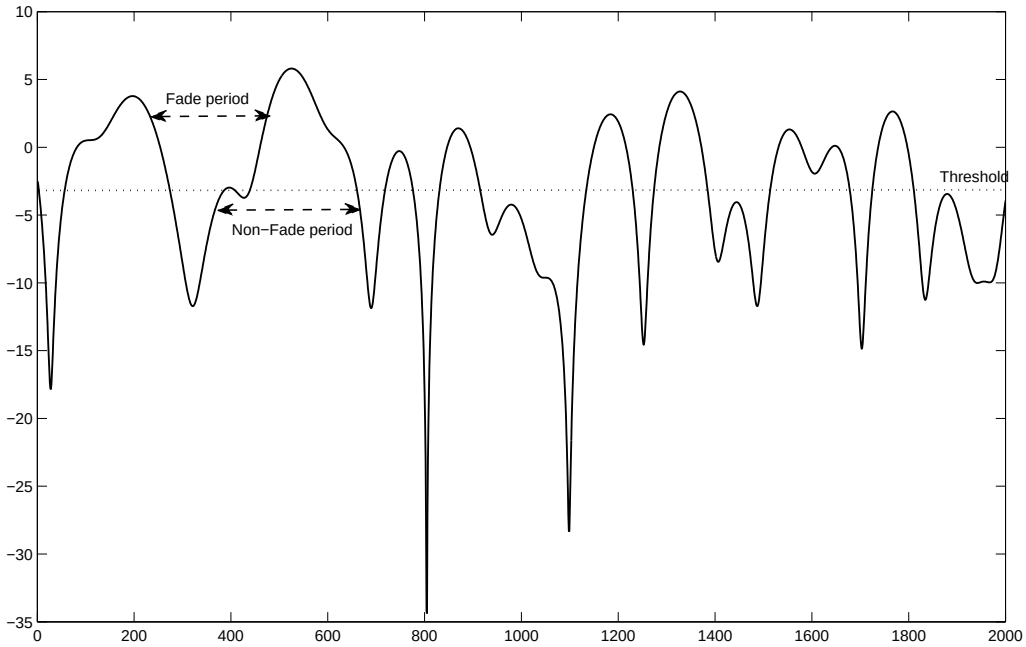


Figure 2.3: Rayleigh Fading Plot

In equation 2.8 a denotes the magnitude and ϕ denotes the phase. It will be assumed that x and y are Gaussian distributed with $N(0, 1/2)$ and also that x and y are independent random variables.

The probability density functions will be:

$$f_X(x) = \frac{1}{\sqrt{\pi}} e^{-x^2} \tag{2.9a}$$

$$f_Y(y) = \frac{1}{\sqrt{\pi}} e^{-y^2} \tag{2.9b}$$

The joint distribution:

$$f_{X,Y}(x, y) = f_X(x) f_Y(y) = \frac{1}{\pi} e^{-(x^2+y^2)} \tag{2.10}$$

But interest should be in the dependence of magnitude and phase of the complex coefficient factor, because the magnitude provides information about the gain (the power, a^2) between transmitter and receiver. It is important to know when a **deep fade** occurs (when the received signal is 0) that is when the received signal can't be distinguished from noise. Mathematically, deep fade occurs if received components have opposite phases. Using polar coordinates:

$$x = a \cos \phi \tag{2.11a}$$

$$y = a \sin \phi \tag{2.11b}$$

it is obtained:

$$f_{A,\phi}(a, \phi) = \frac{1}{\pi} e^{-a^2} \begin{vmatrix} \frac{\partial x}{\partial a} & \frac{\partial y}{\partial a} \\ \frac{\partial x}{\partial \phi} & \frac{\partial y}{\partial \phi} \end{vmatrix} = \frac{1}{\pi} e^{-a^2} \begin{vmatrix} \cos \phi & \sin \phi \\ -a \sin \phi & a \cos \phi \end{vmatrix} = \frac{a}{\pi} e^{-a^2}$$

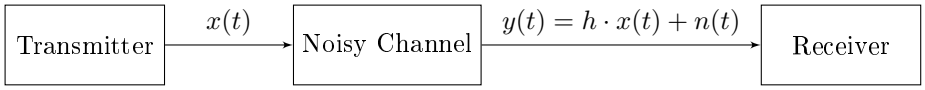
As interest is in the statistical properties of the gain of the wireless channel (how the factor a behaves in our system), marginal distribution with respect to a, representing the amplitude of our wireless channel, should be computed:

$$f_A(a) = \int_{-\pi}^{\pi} f_{A,\phi}(a,\phi)d\phi = \int_{-\pi}^{\pi} \frac{a}{\pi} e^{-a^2} d\phi = 2ae^{-a^2} \tag{2.12}$$

The above result is the **Rayleigh Fading Distribution**. This helps in finding the probability that a transmitted signal from a base station to the user equipment (or mobile station) is worse than say 30 dB, which is extremely useful in estimating the data rate (in bits per second).

2.1.2.5 Bit Error Rate

Bit Error Rate (BER) is one way to measure the performance of a communication environment. It contains information about the number of erroneously received bits. Alongside the effects induced by fading, one of the causes of BER may be the noise at the receiver. Thus, in our wireless channel model $y(t) = hx(t), h = ae^{j\phi}$ it is necessary to consider the noise: $y(t) = h \cdot x(t) + n(t)$.



By knowing that in nature we have multiple random processes that might cause noise at the receiver, that noise generally has uniform power across the band and using central limit theorem we can consider that we have an AWGN (Additive White Gaussian Noise) noise. Thus $n \simeq N(0, \sigma_n^2)$, where σ_n denotes the noise power.

Let B the bandwidth of the channel and P the power required to transmit the information. We have:

- Received signal's power: $P|h|^2 = Pa^2$
- Received SNR (Signal to Noise Ratio): $\dfrac{Pa^2}{\sigma_n^2}$
- Capacity of the channel (Shannon): $C = B\log_2(1 + SNR)$

If we want to transmit bit 0 with power P at the receiver we will have $y = -\sqrt{P} + n$. A bit error occurs if $y > 0 \Rightarrow n > \sqrt{P}$. So we need to compute the probability of this event:

$$P(n > \sqrt{P}) = \int_{\sqrt{P}}^{\infty} \frac{1}{2\pi\sigma_n^2} e^{-\frac{x^2}{2\sigma_n^2}} dx \tag{2.13}$$

Bit Error Rate represents the cummulative distribution function of the standard Gaussian random variable. A deep fade will occur if the noise power is higher than the received signal strength: $\sigma_n^2 > |h|^2 P \Rightarrow a^2 < \dfrac{\sigma_n^2}{P} \Rightarrow a^2 < \dfrac{1}{SNR}$.

Now probability of a deep fading can be determined:

$$a < \sqrt{\frac{1}{SNR}} = \int_0^{\frac{1}{\sqrt{SNR}}} f_A(a)da = \frac{1}{SNR} \tag{2.14}$$

A good approach in improving the performance of the wireless channel is using **diversity** in order to fight with the fading. RX diveristy reduces power and decreases BER. One such technique, used in LTE, is MIMO , which employs multiple antennas on both the receiver and transmitter to constructively use the multi-path effects that always exist to transmit additional data, rather than causing interference.

2.1.2.6 Delay Spread

Multiple antennas are used by the receiver in order to increase the received signal strength. Also, a base station could have multiple antennas for directing a transmission power to a particular receiver, for increasing the signal strength. For a given distance and a proper combination of signals and SNR at the receiver, data rate can increase significantly, if the number of antennas is optimal.

Let $h(\tau) = \sum_{i=0}^{L-1} a_i \delta(\tau - \tau_i)$ be a wireless fading channel for which every path has an attenuation (a_i) and a delay (τ_i). One such channel can be considered as one user. So in a multi-user environment we have multiple $h(\tau)$.

The gain (arriving power) associated with each path of the received signal at one user will be:

$$\phi(\tau) = |h(\tau)|^2 = \sum_{i=0}^{L-1} |a_i|^2 \delta(\tau - \tau_i) \tag{2.15}$$

Denote the gain associated with the i-th path: $g_i = |a_i|^2$. Due to the scattering of the transmitted signal, the receiver has multiple signals received: multiple copies arrive over an time interval. The time spread is called **delay spread**, usually denoted σ_τ. Final result should be a power delay profile, which is useful in evaluating the power of the received signal: given a multipath propagation environment and the delays we want to know the power in dB of the received signal.

In 3G and 4G systems the delay spread is typically around 1-3 μs (value typical for outdoor environments). The indoor delay spread has values of 10-50 ns.

2.1.2.7 Coherence Bandwidth

Another important concept in wireless communications is the **Coherence bandwidth**, which is a statistical measure of the frequencies which indicate if the channel can be considered *flat* - all components from the spectrum are passed by the channel with equal gain and phase. Coherence bandwidth is a property of the multipath environment (a property of the channel) for a given delay spread.

This parameter provides information about the distortion of a transmitted signal.

Given a multipath delay profile from a wireless channel $h(\tau) = \sum_{i=0}^{L-1} a_i \delta(\tau - \tau_i)$ the spectrum of this profile can be evaluated as:

$$H(f) = \int_0^\infty h(\tau) e^{-j2\pi f \tau} d\tau = \sum_{i=0}^{L-1} a_i e^{-j2\pi f \tau_i} \tag{2.16}$$

Coherence bandwidth provides information about a signal with bandwidth less than or equal to the coherence bandwidth will have no distortion at the receiver. If the transmitted signal has a bandwidth greater than the coherence bandwidth then some attenuated parts will be outside the coherence bandwidth and so the receiver will experience a *distorted signal*. Let B_S the bandwidth of the signal and B_C the coherency bandwidth:

- $B_S < B_C \Rightarrow$ No distortion. The wireless channel flat-fading (no inter-symbol interference occurs).
- $B_S > B_C \Rightarrow$ There is distortion can be seen in the edges of the signal (in the central part everything is all right). This is called frequency selective distortion.

Of particular interest is evaluating the points of significant change in the frequency response of the wireless channel - $H(0)$ against $H\left(\frac{1}{4\tau_i}\right)$. Thus, the coherence bandwidth for our multipath system can be approximated to the value: $B_C = \frac{2}{4\sigma_\tau}$, in which σ_τ represents the root mean square delay spread. This value is used in practice in 2G, 3G and 4G systems.

The conclusion that can be drawn is that as delay spread increases the coherency bandwidth shrinks.

For a delay spread of 1 μs (the typical value in outdoor channels) the coherency bandwidth is 500 kHz.

- 2G GSM has bandwidth of 200 kHz. This means that it is a flat fading system.
- 3G WCDMA, which is a wideband spread spectrum system, having a relatively large bandwidth value, of 5 MHz, which is much greater than the coherency bandwidth equal to 500 kHz. In this system it is encountered frequency selective fading, resulting in inter-symbol interference. Specific mechanisms are to be implemented at the receiver to mitigate these phenomena. The solution is called **equalizing**, a technique of compensating the distortion at the receiver, and it is shown in figure 2.4 [EDS11, p .45-46]

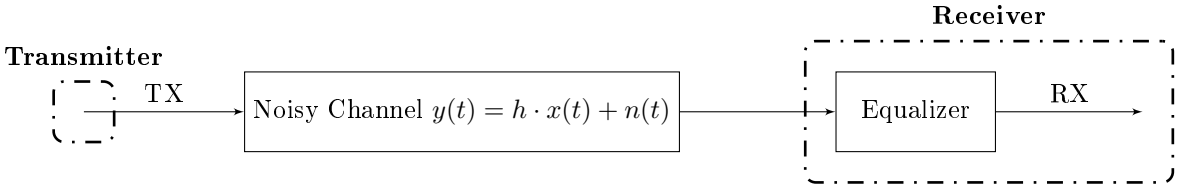


Figure 2.4: Equalizer placed at the receiver

- If the coherence bandwidth is increased to even bigger values, 20 MHz, as it is done in LTE, the high-performance equalization at the receiver side becomes a problem. The complexity of a receiver equipment should not be too high. A workaround is applying less optimal equalization techniques. A long term solution is using transmission modes or schemes that enable a higher throughput in the radio link, such as multi-carrier transmission - an overall wideband signal transmitted as multitude of multiplexed narrowband signals. The great achievement is bandwidth increase without having high SNR due to frequency selective distortion. A major disadvantage of multi-carrier transmissions is the variations in the strength of the transmitted signal. As mobile terminals should have low power consumption, another solution could be a single-carrier transmission scheme, especially for uplink. Reader can find extensive information about these topics in [EDS11, p. F15-71].

The conclusion that can be drawn from this section is that wireless radio channel is a medium in which it is very difficult and challenging to achieve high data rates because of noise, interference and other impediments which are variable in time. Movement of the UE could also introduce unpredictability in channel estimation. [Gol05] To sum up, different implementations of wireless communications (such as GSM, WCDMA or LTE) are heavily based on concepts and take into account the limitations that might be induced. Usually, a tester has three major tasks and the testing solution offers a generic framework, so that any of these tasks can be done:

- **Conformance testing** - use specialized equipments (such as Vector Signal Generator or Analyzer) for:
  - Study Fading, BER or other specific physical channels
  - Apply AWGN or induce high BER in a communication
  - Apply radio channel models and noise using theory described in this section
- **Coverage testing** - ensure that 3GPP and third parties standards are met, targeting functional validation
- **Load testing** - find the hardware and software limitations of the base station

2.2 LTE Network - General Concepts

In this section focus will be on some key elements of LTE Network that were implemented in the testing solution and also to highlight points which are relevant for this thesis.

The most important requirement that Long Term Evolution Network (on which work started in 2004) should meet is peak data rates: 100 Mbps for downlink and 50 Mbps for uplink for 20 MHz spectrum allocation, assuming two receive antennas and one transmit antenna at the mobile terminal. Also, latency in LTE in the communication between base station and UE has to be less than 5 ms.

LTE uses two duplexing techniques to separate uplink and downlink transmissions between the terminal and the base station: Frequency division duplex (FDD) and time division duplex (TDD) , as shown in figure 2.5 [ST07, p. 2].

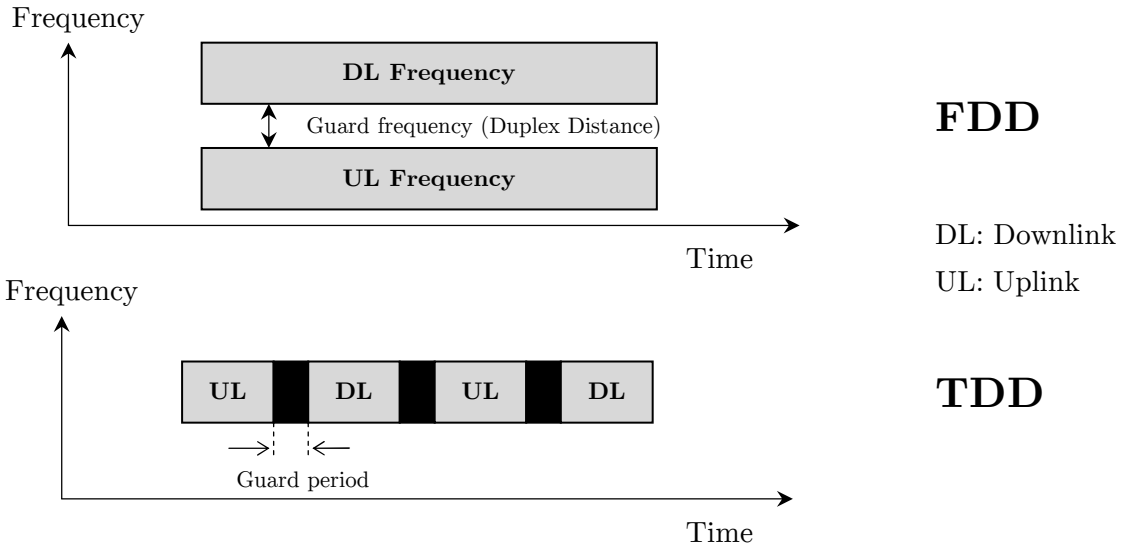


Figure 2.5: FDD and TDD used for duplexing

Another key feature of LTE is the modulation scheme (used to carry the digital data over analog waveforms): it can only be QPSK (Quadrature Phase Shift Keying) , 16QAM (Quadrature Amplitude Modulation) or 64QAM. This is a consequence of the requirement of high data rates within a fixed transmission bandwidth: more bits of information should be transmitted in one modulation symbol:

- QPSK allows up to 2 bits of information per symbol
- 16QAM allows up to 4 bits of information per symbol
- 64QAM allows up to 6 bits of information per symbol

The reason for using these modulation schemes is that they provide a higher bandwidth utilization, resulting in high data rates for a given bandwidth. But there is a trade-off: for a given SNR only a certain combination of modulation and coding rate of the radio channel (only a certain MCS is optimal) is optimal, meaning that it delivers highest usage of the bandwidth and consequently highest data rate for that bandwidth.

The LTE radio-access network uses an architecture containing a single type of node, responsible for all radio-related functions in the cells - the eNodeB. Note that this is a logical node. Moreover a base station is a possible implementation of, but not the same as, an eNodeB [EDS11, p. 110-111]. The architectural diagram of the LTE network is shown in figure 2.6. Briefly, it consists of:

- **E-UTRAN** - Evolved Universal Terrestrial Radio Access Network , which represents the air interface, which is basically made out of the eNodeBs
- **EPC** - Enhanced Packet Core - the evolved core network usually found in GSM, which has:
  - **MME** - Mobility Management Entity - a node responsible for handling idle to active transitions, security, connection or release of bearers to a terminal.

- **S-GW** - Serving Gateway - a node which connects EPC and E-UTRAN, acting as an anchor for terminals moving from one eNodeB to the other
- **Packet Data Network Gateway** (P-GW) is the node responsible for connecting the EPC to the Internet
- **PCRF** - Policy and Charging Rules Function - a database used for quality of service handling and billing
- **HSS** - Home Subscriber Service - a database with information about the subscribers, used in the billing process

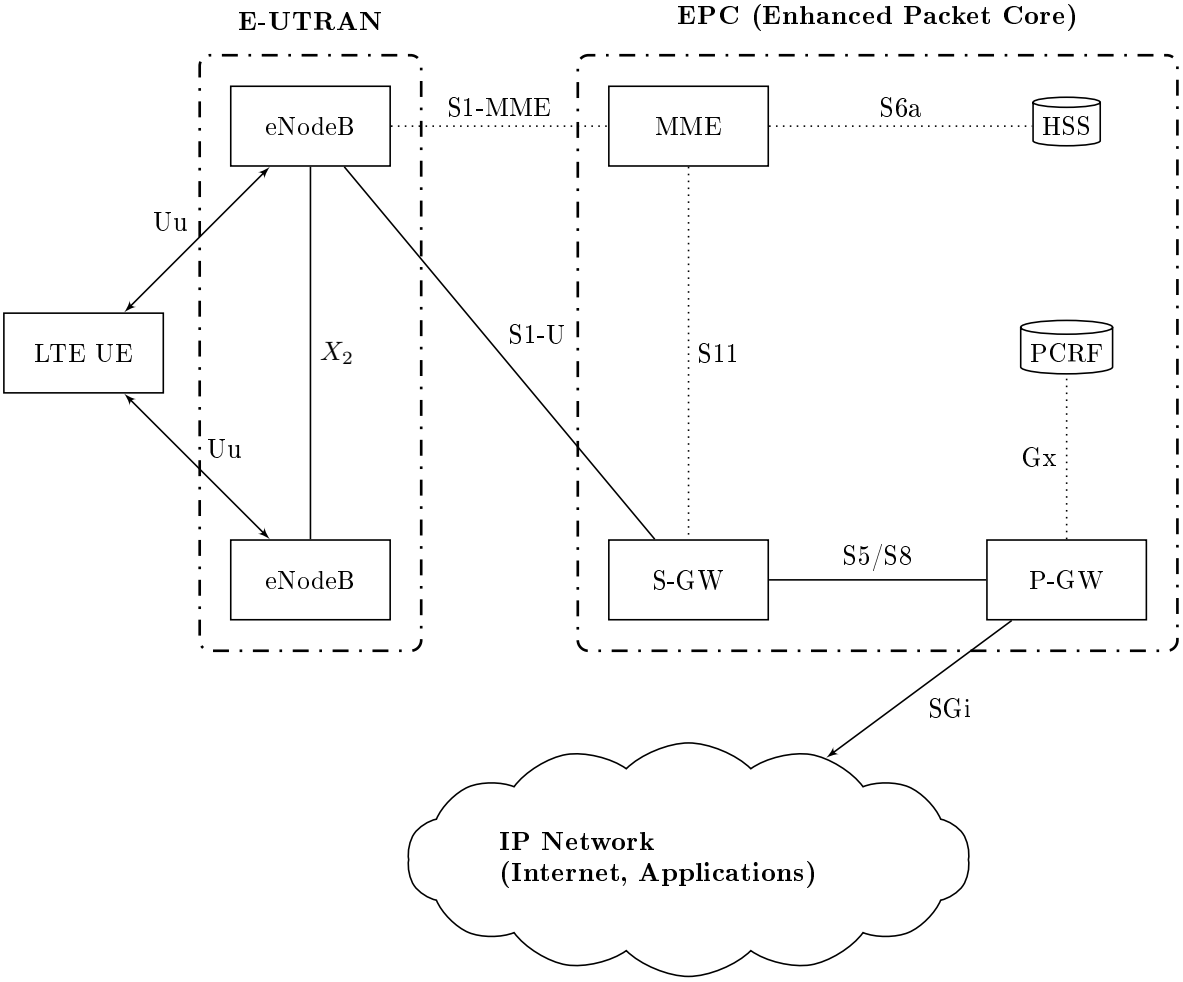


Figure 2.6: LTE Network

Once the Layer 3 packages come to Layer 2 their headers are compressed by PDCP, in order to reduce the number of bits transmitted over the air, then passed to RLC (Radio Link Control). In this point the data processing consists of segmentation, retransmission handling and duplicate detection. For an UE there is one RLC entity per radio bearer. Once RLC processing is done, the RLC protocol data units are passed to MAC sublayer through logical channels for handling multiplexing, retransmissions (using techniques such as HARQ - Hybrid Automatic Repeat Request, described in section 2.3, present in both UE and eNodeB) and uplink/downlink scheduling of users, for which the eNodeB is responsible. This protocol chaining can be seen in figure 2.7 and more detailed in figure 2.11. Any of these steps (detailed in [EDS11, p. 111-121]) can be error prone, and focus is only on finding faulty behavior of Layer 1 (Physical Layer). Thus, a custom Scheduler and/or MAC entity, in which RLC, PDCP and RRC functions and processing are stripped or even eliminated, is required.

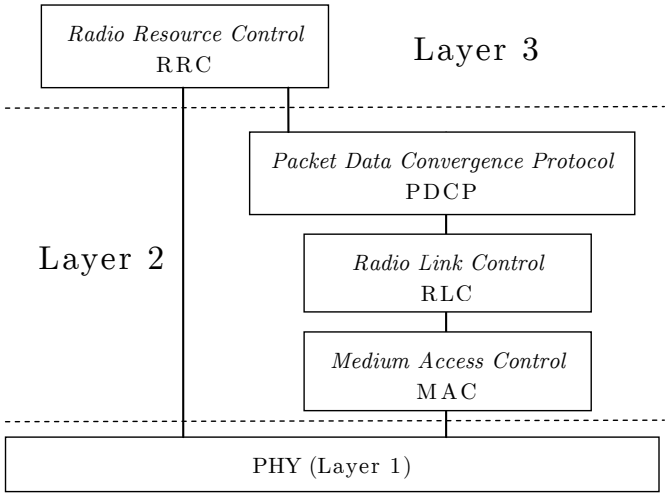


Figure 2.7: LTE protocol architecture in Layers 1, 2 and 3 [MB14]

2.3 Brief Overview of HARQ Retransmission Protocol

The Hybrid Automatic Repeat Request (HARQ) is a combination between the classical ARQ (Automatic Repeat Request) encountered, for instance, in Ethernet at data link level and forward error correction. For each mobile terminal there is one HARQ entity defined. When receiving data, the receiver will attempt to decode it and send an ACK to the transmitter whether data was correctly received. HARQ uses soft combining, which means that the receiver combines the multiple received signals from previous transmission attempts (so the receiver should know how to differentiate between initial transmission and retransmission). This is the reason for the existence of a one-bit flag, called NDI (New Data Indication) which highlights if data is from current transmission or it is from a new transmission.

In a FDD implementation, there are up to eight HARQ processes (eight *stop-and-wait* processes which run in parallel), so that the transmitter does not have to wait for ACK/NACK and transmits data to another HARQ process. Thus, the easy implementation of a *stop-and-wait* protocol is maintained, while transmission is not being fragmented by the blocking wait for an ACK or a NACK. HARQ entity in Layer 2 resides in the scheduler and it is responsible for controlling all these eight processes. This entity manages both uplink and downlink retransmissions and new packet generation. Scheduler regularly generates DCI (Downlink Control Information), based on feedback collected from physical layer running on both eNodeB and UE, containing information about the errors in receiving packets and new transmissions of packets. Usually, new transmissions have priority.

Figure 2.8 shows how NDI toggling mechanism can be used to know exactly when retransmissions occur.

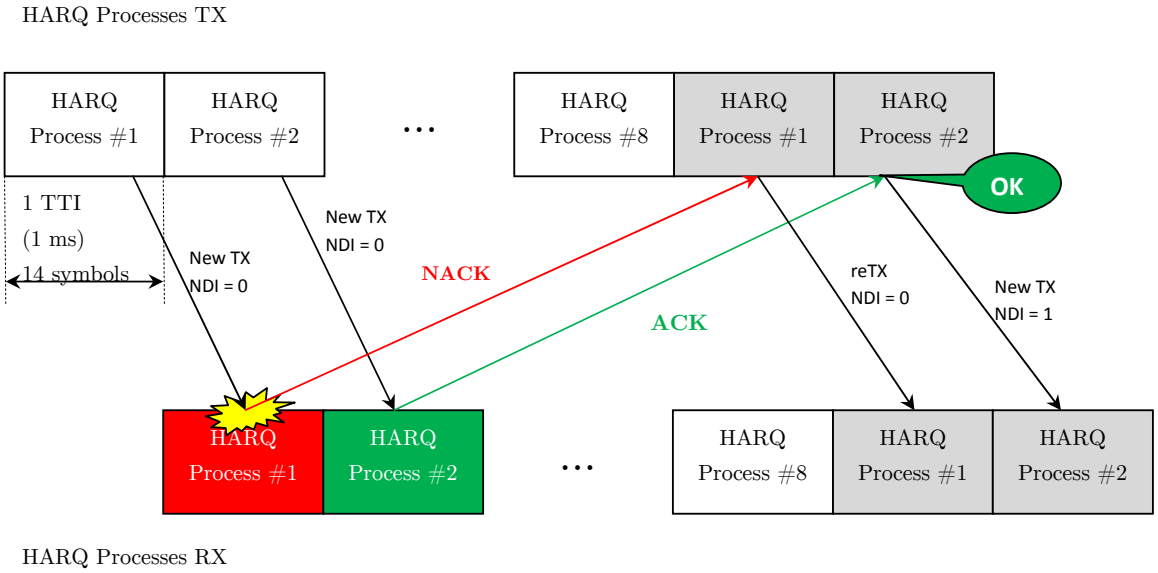


Figure 2.8: HARQ retransmission and NDI toggling. Original image is [dTdC11]

2.4 Motivation for a Custom Layer 2 in Testing Environment

In a full-protocol testing the pass or fail decision can only be taken based on the final results of specific procedures, such as UE attachment to network, described in figure 2.9 or obtaining a specific throughput or data rate. In this case Layer 1 behavior is transparent for the test outcome. [MB14, p. 1]

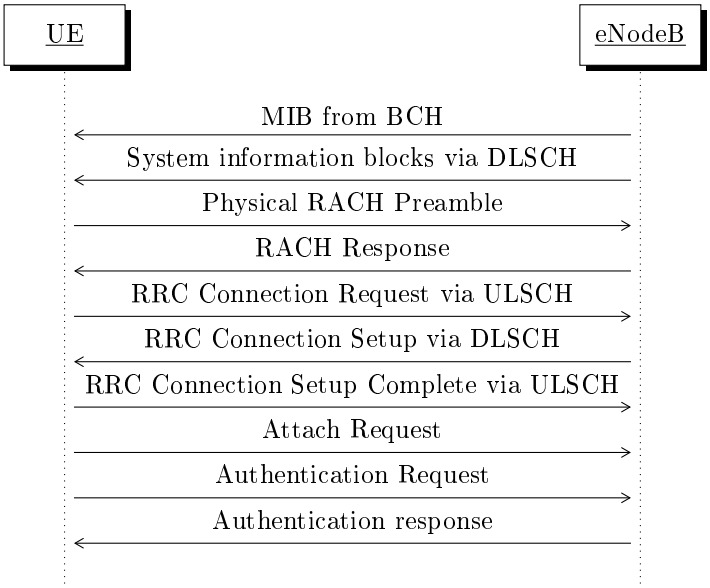


Figure 2.9: UE attach to network in LTE

Also, in the context of Layer 1 validation, there is a necessity to exclude any higher layer logic. For instance, retransmissions can also occur due to a congestion detected by TCP at Layer 4, which is a consequence of high network load, but not necessarily an indicator of Layer 1 software problem.

Thus, in the **BlackBox Testing** solution an emulated Layer 2 is used , which isolates the testing of Layer 1 from issues that might occur at higher layers or in LTE network. The user has full access to the Layer 2 API and has the possibility to define a custom Layer 2 scenario by sending specific messages (physical layer control messages) to Layer 1. One can imagine the Layer 1 acting as a finite state machine, to which

if you send different messages, it gets from one state to another, where there are different jobs running or scheduled in a queue to run, according to the parameters contained in the message, as shown in figure 2.10, in which the states could be:

- IDLE - Physical Layer is ready to be configured for a certain scenario or deployment
- L1 Configured - Physical Layer is configured for a certain scenario or deployment or is ready for reconfiguration
- L1 Running - in every TTI (Transmission Time Interval) equal to 1 *ms*, Physical Layer receives downlink and uplink messages that contain requests to configure all operations to be done within that TTI.

FAPI standard interface between Layer 2 and Layer 1, described in section 3.2, uses this model of Layer 1 Software as a simple finite state machine (another state called Network Monitor Mode or Network Listening Mode might be added, in which physical layer is ready to listen for specific radio signals for channel estimation or measurements) in which the implementation of Layer 1 software is stateless and all information regarding an UE is stored in Layer 2. [Aus13, p. 18]

The user can define a suite of messages in a test script to bring Layer 1 through different states, and send different downlink messages (from base station to the UE) or emulate uplink messages sent, as it will be further detailed, by the UE to the base station. Once the upper layers are stripped, for a given scenario there might be only four points of failure:

- Layer 1 does not respect the finite state machine, or problems occur in downlink/uplink jobs when processing the parameters received or to be sent to the UE.
- Hardware issues of the base station
- Invalid scenario imagined by the tester
- Testing Environment Problems

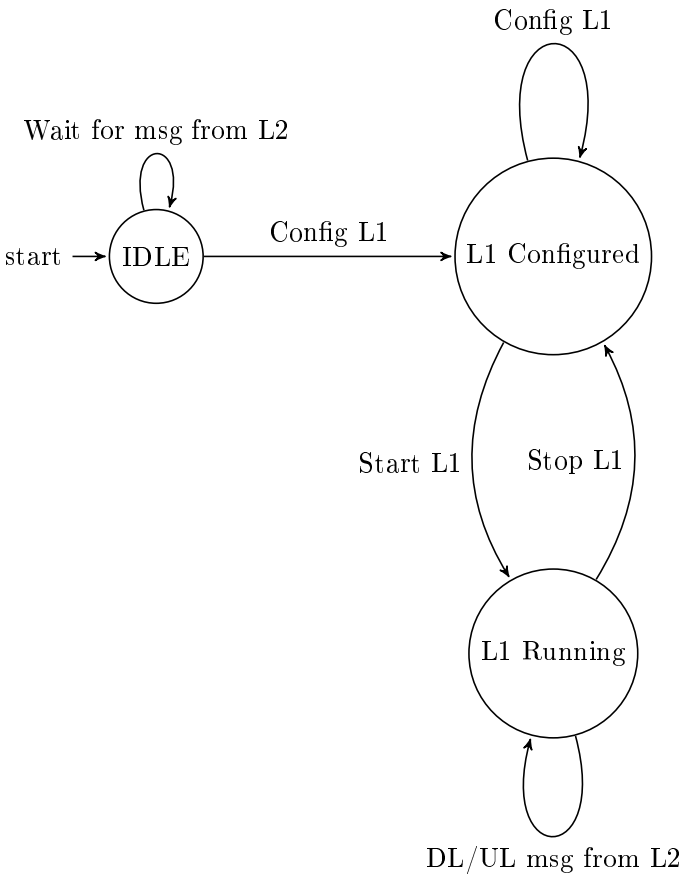


Figure 2.10: Layer 1 Finite State Machine

Also, in the emulated Layer 2, there is no need for interaction with PDCP or IP, since it is assumed that the multiplexing of RLC PDUs and the attachment of the MAC header are already done. For Layer 1 data payload is presented as transport blocks irrespective of their informational content.

By using full-protocol testing, one can have access only to the application layer data payload in the OSI Stack. A traffic made by an user on his mobile phone will pass through all network stack layers, and at every layer a header will be added, to which the tester has no control. In figure 2.11 it is shown a flow commonly encountered in an Internet exchange encountered in LTE at the eNodeB (in a downlink scenario), from Layer 3 to Physical Layer, with the jobs of each protocol.

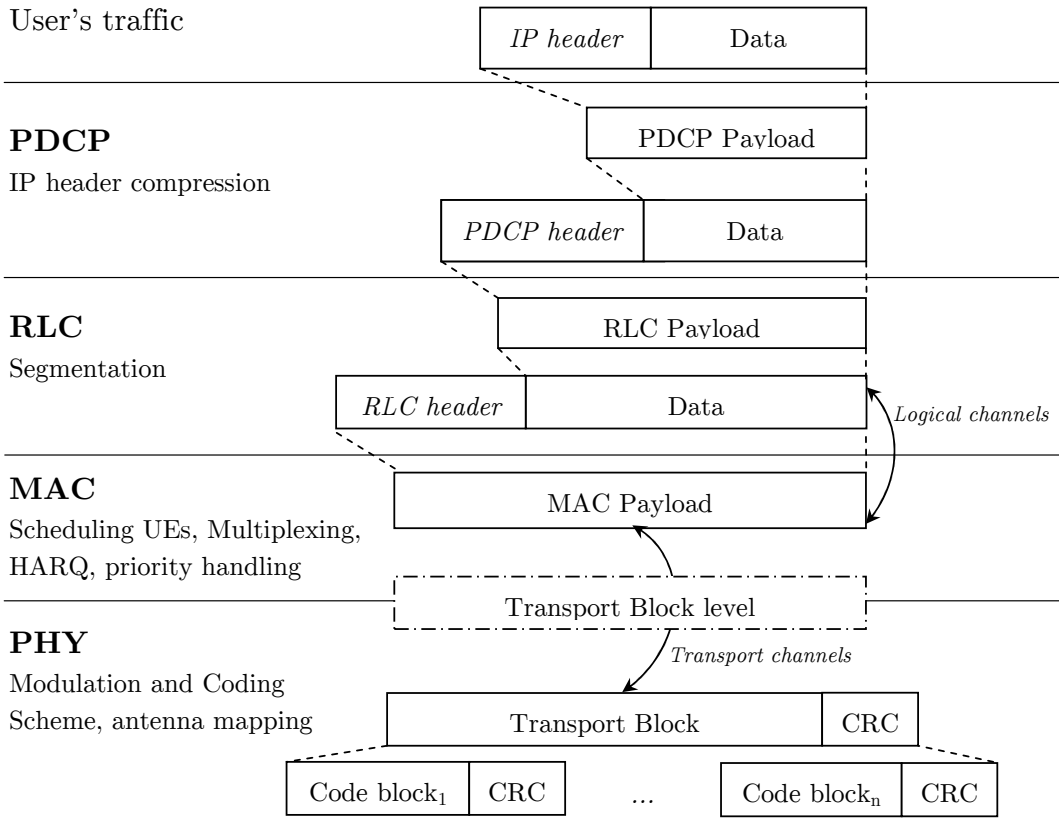


Figure 2.11: LTE flow in Layers 1, 2 and 3

In BlackBox testing solution, testers and users have full access to the content of MAC payload (header and raw data) and when defining a scenario/test case as a script he could directly send the content of transport blocks which result from MAC. These blocks are to be sent to Physical Layer for coding and further processing, hence removing the need for IP, RRC, PDCP and their possible errors. Consider transport blocks as a set of resource blocks with the same modulation and coding scheme (MCS) that have to be applied by the Physical Layer. Furthermore, the tester has full access to the content, size and the MCS of a transport block, meanwhile in a full-protocol testing the size of a transport block was in a deep relationship with the data rate selected by link adaption or the resource scheduling done by MAC.

For a proper analysis post test, the tester is able to see in the log files what messages were actually sent by BlackBox to Layer 1, the content of transport blocks received by Layer 1, the timing and what jobs were done in Layer 1 when receiving a transport block with specific content. Layer 1 will use the transport channels to send its response to MAC. Automated pass/fail decisions are taken based on statistics made by the custom Layer 2, using the data from transport channels or by the test script. Transport channels used by Layer 1 to provide services to Layer 2 are [EDS11, p. 116]:

- **BCH** (Broadcast Channel) - contains system information about the network collected from all users in current cell. BCH transmits a block called MIB (Master Information Block) which contains :
  - Cell's downlink bandwidth, measured in number of resource blocks (see appendix A for more details about resource blocks. Purpose of resource blocks in downlink processing chain will be

detailed in section 4.2)
  - Information about the configuration of HARQ Indication, for the UE to know how retransmissions are done in current cell
  - SFN (System Frame Number) - a number used by the eNodeB and the UE to synchronize in time. If LTE clock would be compared with a ordinary clock, then LTE's hands are SFN and subframe (SF). SFN represents the hand ticking at the 10 *ms* interval and has values between 0 and 1023. The other hand of LTE's clock, SF, ticks every 1 *ms* (every one TTI) and has values between 0 and 9. UE and eNodeB have to set SFN and SF to the same values before starting to communicate. From now on, a timing axis will be measured in SFN and SF, and not in seconds.

- **DLSCH** (Downlink Shared Channel) - the main channel used for transmitting data from the eNodeB to the UE. On DLSCH are also found the SIBs (System Information Blocks) [1] , which are an extension of system information sent as MIB via BCH, which help the UE to access the cell and perform cell reselection or if the UE is allowed to attach to that particular cell.
- **ULSCH** (Uplink Shared Channel) - the main channel used for transmitting user's traffic to the eNodeB.
- **PCH** (Paging Channel) - paging is the network's way to notify the UE that it has some data for him. Paging happens when the UE is idle, as UE in sleep mode only scans the PCH, in order to save battery. PCH only sends at predefined moments of time some data, so the UE will only wake up to scan the network at these particular moments.
- **MCH** (Multicast Channel)
- **RACH** (Random Access Channel) - although this channel does not contain transport blocks, it is considered a transport channel. It is an uplink channel used by the UE in order to gain access to the network from the eNodeB and initial synchronization with it. Figure 2.9 shows RACH role in UE attach to network. Usually, after power on, the LTE UE does the following things:
  1. Frequency search
  2. Time and frame synchronization with the base station
  3. Decode MIB came via BCH
  4. Decode SIB came via DLSCH
  5. Initial RACH procedure for entering the network

The next chapter will describe the setup used for BlackBox testing of Layer 1, using the approach described in this section, deployed on Freescale® BSC913x platforms, and how data is extracted from the base station, for a faster search of the root cause of a faulty behavior.

[1] In 3GPP TS36.331 version 9.3.0, there are defined 13 SIBs for LTE and also there are described the actions required for the UE upon receiving specific SIB via DLSCH

Chapter 3

Testing and Profiling Setup

In this chapter is presented the setup used with the emulated Layer 2 implemented and with dedicated test mobile (a third party UE emulator), which stripes the functionality of higher layers, and the main points from which data could be extracted for system analysis.

3.1 Freescale® BSC913x Architecture

The system on a chip BSC913x platforms (block diagram of BSC9132 is shown in figure 3.1) consist of:

- e500 cores built on Power Architecture® with 256 KB shared L2 cache
- StarCore® DSP SC3850 with 512 KB private L2 cache
- Multi Accelerator Platform Engine, MAPLE-B2F, for baseband processing, which has support for acceleration of signal processing operations (FFT , DFT , Turbo Coding), fast matrix inversion operations and CRC algorithms

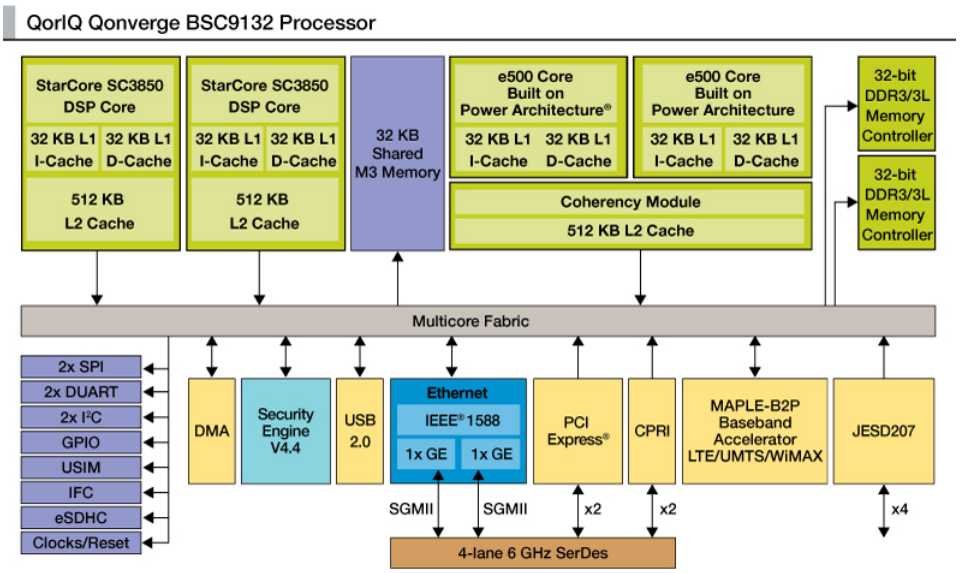


Figure 3.1: Freescale® BSC9132 architecture

The emulated Layer 2 software is deployed on the general purpose Power Architecture processor and the device under test (DUT) is the Layer 1 software solution deployed on StarCore® DSP which uses MAPLE-B2F for faster processing. The two software solutions communicate via a proprietary Inter Processor Communication (IPC) protocol. The interfacing between Layer 2 and Layer 1 is done using Femto Forum Application

Platform Interface (FAPI) [1], which is a common interface between multiple vendors of Layer 1 and Layer 2 software solutions. The main purpose of a standardized interface is to reduce time to market. The proprietary mechanism IPC is used for transporting FAPI and the message exchange between Layer 1 (on StarCore) and Layer 2 (on Power Architecture).

Testers do not have access to neither Power Architecture nor the DSP, as their focus is on LTE scenarios. They only have access to a command prompt, with possibility of writing test scripts and individual commands for automation, which communicates via a network socket with the emulated Layer 2, and sends serialized FAPI commands.

3.2 FAPI Interface for Communication Between Layer 1 and Layer 2

As it was mentioned in 3.1, FAPI is an interface between multiple vendors of Layer 1 and Layer 2 software solutions.

All FAPI messages which come from the communication between Layer 1 and Layer 2 offer valuable information about the uplink and downlink traffic and radio parameters. Usually, there are three major types of FAPI messages, as defined in [For10]:

- `.request`: messages which come from Layer 2 to Layer 1, as command messages
- `.response`: messages sent from Layer 1 to Layer 2, as a response to a `.request` message
- `.indication`: messages which come asynchronously from Layer 1 to Layer 2 in case of special events that occur in Layer 1. For instance, the message `ERROR.indication` will come from Layer 1 if Layer 2 sends a start request to Layer 1 when Layer 1 is already started and configured (remember the finite state machine from figure 2.10)

For example, there is a CQI (Channel Quality Indicator) reporting mechanism used by the emulated Layer 2 to determine the quality of downlink radio channel. The reporting of CQI is done in two possible ways:

- During a "simulation" of a RRC connection (see figure 2.9), the *"emulated"* Layer 2 will ask the UE for periodic CQI reports and use them for internal statistics.
- A downlink control channel is used to ask the UE for aperiodic CQI reports.

Continuing the example with channel quality indication, in order to start a CQI reporting, the *"emulated"* Layer 2 sends a control message to (FAPI standard defines for aperiodic CQI `HI_DCI.request` - HARQ Indication Downlink Control Information Request). If the information regarding CQI is stored in the MAC Layer, then Layer 1 should receive an uplink configuration message, in order to send the proper indication to upper layer with channel information received from the UE, as shown in figure 3.2. The uplink configuration request message (`UL_Config.request`) can contain multiple types of PDUs whether the UE is scheduled to transmit data, to transmit a CQI report, an acknowledgment etc.

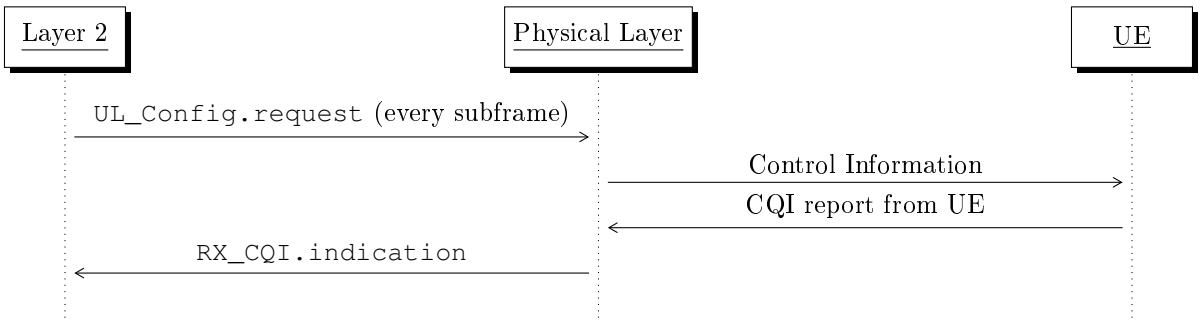


Figure 3.2: Simplified CQI Reporting Mechanism as it is defined in FAPI [For10]

As mentioned above, all these FAPI messages are sent from the Power Architecture over the network and a sniffer (such as Wireshark) is started on tester's PC for capturing these packets. Once such communication is

[1] http://www.smallcellforum.org/resources-technical-papers

finished, a packet capture file (`.pcap`) is generated and then the Trace Visualizer takes this file as input and starts a dissection. If one is interested in monitoring downlink SNR, the `RX_CQI.indication` contains, for every SFN and SF a number of PDUs which contain the RNTI (Radio Network Temporary Identifier) which identifies the UE and another field that contain SNR information with values from -64 dB to 63.5 dB (with 0.5 step) and timing advance information: how long does it take for the signal to reach the UE from the base station, and it is used to determine the distance between the UE and the base station thus providing grounds for localization.

Continuing our example, what it was described above was the downlink channel quality estimation. The quality of uplink radio channel is determined by using sounding reference signal (SRS) mechanism , described in figure 3.3. In this case, the Layer 1 Trace Viewer should inspect the `SRS.indication`, as it contains also timing advanced information and number of resource blocks allocated for each RNTI (user) which are reported at a specific SFN and SF, and each resource block reports a SNR value from -64 dB to 63.5 dB.

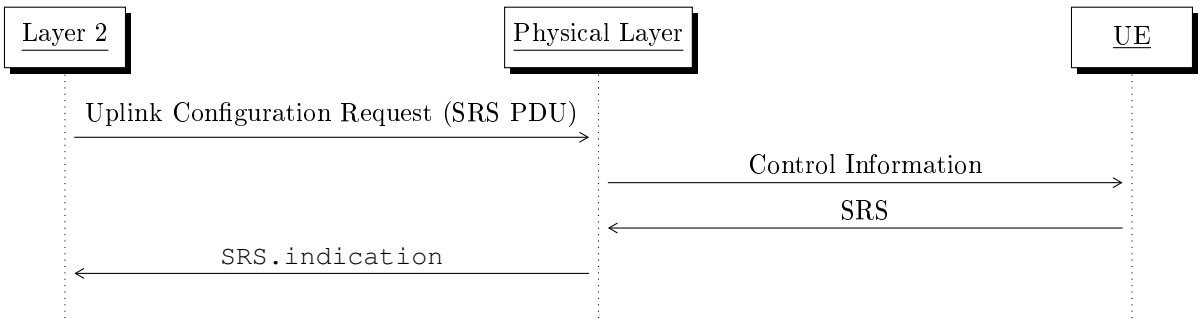


Figure 3.3: Simplified SRS procedure as it is defined in FAPI [For10]

Based on SNR information extracted as discussed above, one can decide what MCS can be used in different channel conditions, as discussed 2.1, making use also of a CRC error statistic.

3.3 BlackBox Setup Environment and Modules

In order to have a full isolation from higher layers, as discussed above, testers should also have access to a test mobile, a dedicated equipment which emulates one or multiple UEs, which should offer a layered approach to testing the physical layer of the base station and verbose logging mechanism. Although testers have access to a graphical user interface of the test mobile, for automation there is also a specific API defined which enables control of the test mobile from a script or from the command prompt.Thus, a script is a sequence of FAPI commands which are to be sent to the Layer 2 and multiple configuration commands which have to be in sync with the FAPI commands to the test mobile, as well as commands to both Layer 2 and test mobile are sent via sockets, as it is shown in the figure 3.4. All equipments can be accessed remotely by anyone who has access in the private network and can be remotely powered on or off or rebooted from any computer in the network. To sum up, the testing environment is described in figure 3.4 and it is made out of:

- **Base Station** - containing the DUT (Layer 1 solution) and the emulated Layer 2, which has minimal Layer 2 logic (for instance resource scheduling requests handling on both uplink and downlink, HARQ retransmissions, logical channels multiplexing) support for extracting statistics useful for taking pass/fail decision in a test scenario (such as BER or CRC, which come from Layer 1).
- **Test Mobile** - a dedicated equipment, third party UE emulator
- **Database** - used to store test results, job queue and location of log files for post test analysis.
- **Test Execution PC** - a computer containing two important agents for full automation of running multiple triggered tests:
  - **Polling Agent** - regularly queries the database, looking in the job queue table, to see if it has new jobs. A new job is started only after the current running job stopped.
  - **Job Runner** - a module responsible for parsing the list of tests which was sent by the tester via the job queue in the database and starts scanning the disk on the Test Execution PC for the files which contain the definition of the scripts, mentioned in the list of tests. At the end of each test, the Job Runner is responsible for inserting in the database the test result, execution log and other logs generated during the test in the database.

- **Web Server** - used to render web pages containing detailed test status reports using data extracted from the database and offer links for downloading logs. Also, an interface for running pre-defined scenarios or custom scenarios is offered via web forms.

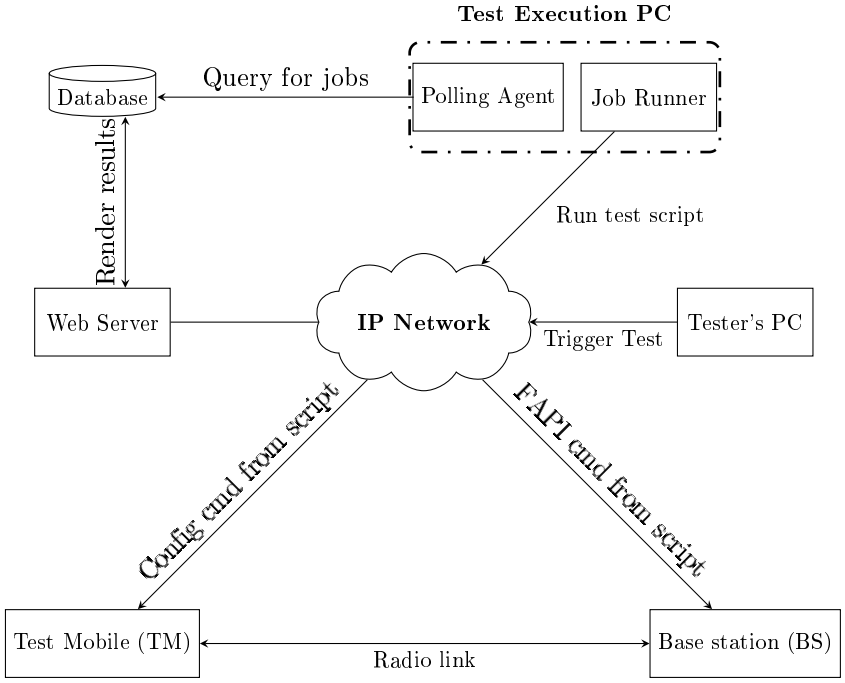


Figure 3.4: Testing infrastructure

A fully automated testing workflow, with current implementation has the following steps: the user (or the tester) clicks the *"run"* button in the web interface and his request is enqueued in the job queue in the database, with a predefined priority. If there is a current job running, the Polling Agent will put the current jobs on hold (leaving them in the job queue), else it sends the current job as an object to the Job Runner. The object received by the Job Runner contains the following fields:

- Job Name
- Class of Job (LTE Tests, WCDMA Tests etc.)
- Path to a text file containing a list of functions (the test cases) which are various scripts across the test exection PC

Job Runner will now start parsing the text file and if this step is successful, the job is started. This text file also contains some instruction for the Job Runner such as:

- On what board should the tests be done (on BSC9131 or on BSC9132)
- If the board must be rebooted at the beginning of each test case
- If the test mobile must be rebooted at the beginning of each test case
- If the equipments (board or test mobile) have to be rebooted before starting the execution of the test cases
- What band and what bandwidth should be used for these test cases (for a proper configuration of the radio-frequency cards of the board)

The test cases are defined as individual scripts on the test execution PC and the tester, for a proper automation without errors should place all the files on the PC, before running the tests. Once the scripts are present, the files sent to the Job Runner can contain various calls of these test cases with different parameter values (parameter sweep). A collection of test cases alongside input parameters and above mentioned session configuration is called a suite file. At the end of the test session, user will find in the web interface a table containing all the jobs and their test cases with full test status report: passed, failed or environment, start time, end time and link to logs found on the storage, for downloading them. The workflow is summarized in figure 3.5

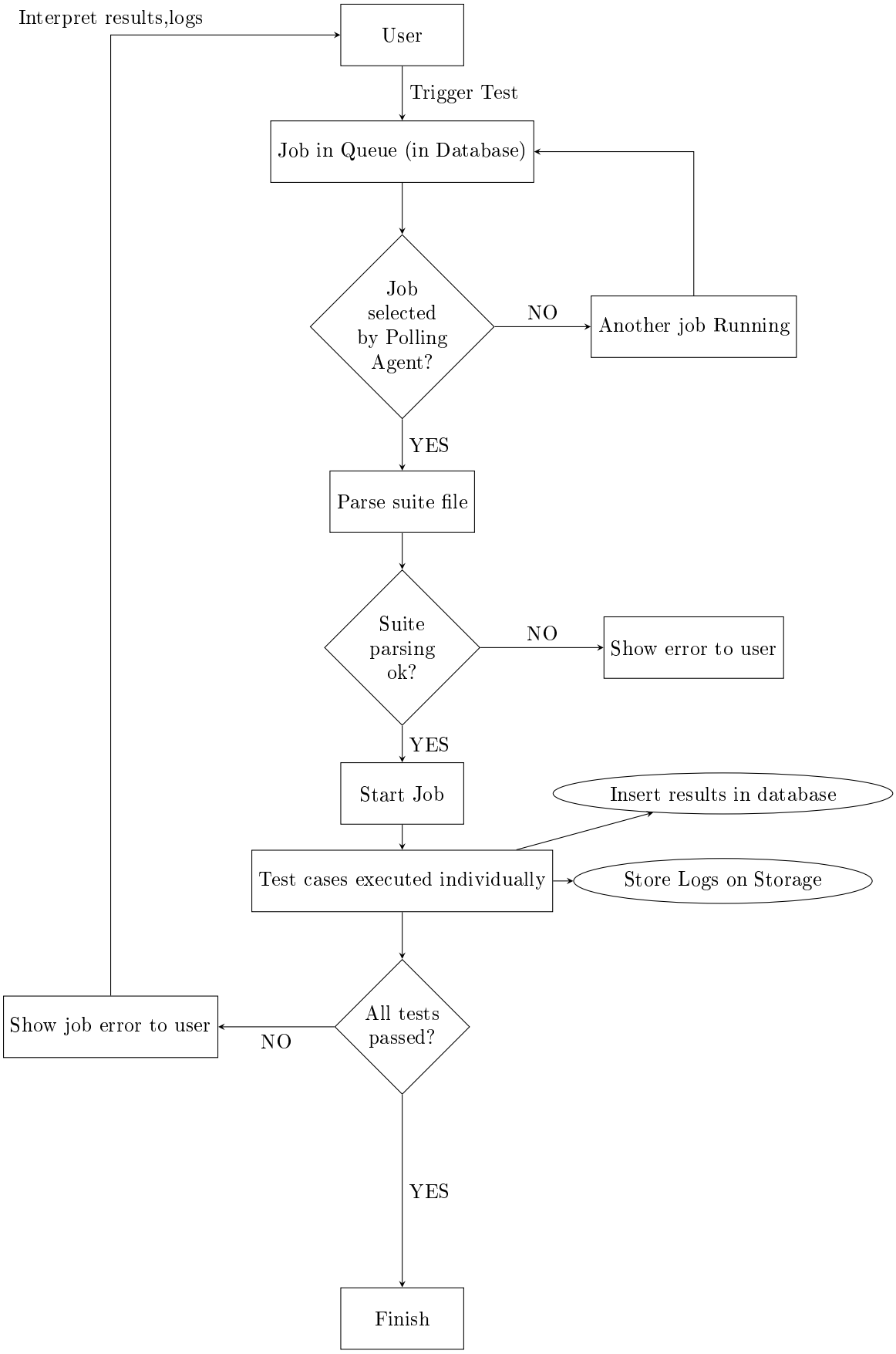


Figure 3.5: Testing algorithm - Automated steps

3.4 Trace Visualizer Deployment

The most relevant logs for monitoring the Physical Layer are those who contain information about the main points of failure in the base station side, where either hardware or software errors could occur:

- Power Architecture (Layer 2)
- StarCore DSP
- MAPLE

Other problems can have the root cause in the testing environment, for instance high latency in the network communication between test script and base station or forcibly closed connection due to network failures, or test mobile issues, but these are neither subject to our device under test nor to system analysis of the base station, as the Trace Visualizer has only full access to data from base station's system memory. Figure 3.6 highlights the communication between tester's PC, BSC913x platform and additional modules used for an automated testing environment. Modules highlighted with gray have been installed and configured by the author of this thesis.

On both, StarCore® and Power Architecture® there are two agents which have a specific print call that inserts information in system memory, in a proprietary compact bit stream format, which is then exported as a binary file (using functions implemented in BlackBox, that extract memory content) and then parsed by the Trace Visualizer.

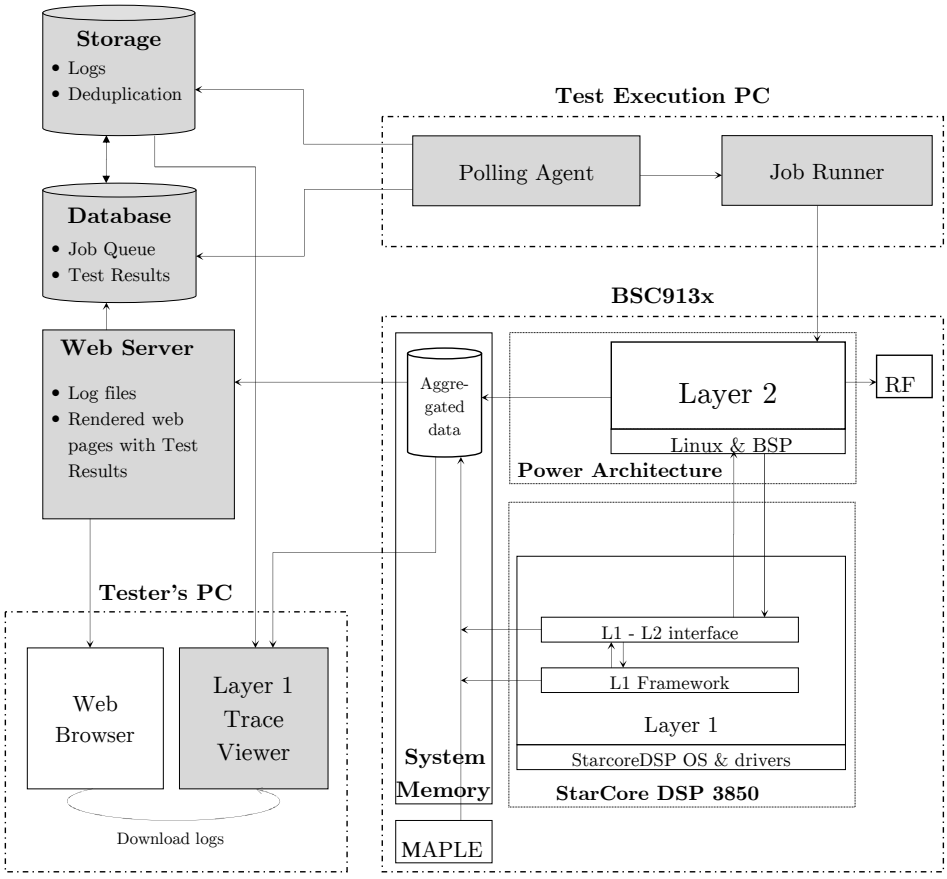


Figure 3.6: Testing infrastructure: communications between board and tester's PC

Following the workflow described in figure 3.5, once a test case is finished the tester receives multiple logs, but the most relevant from a Layer 1 monitoring perspective are:

- `L1Trace.dat` - a binary log file containing the dump of the system memory with all the data send by StarCore and MAPLE accelerator. The format of this file is a proprietary bit stream designed to encode maximum amount of system info with minimum overhead in data processing and transmission.

- `FAPI.pcap` - the capture made by the sniffer (e.g.: Wireshark) containing all FAPI messages sent during the message exchange between Layer 1 software and emulated Layer 2. For this thesis, focus was on this type of files. This file contains information about downlink and uplink transmission (contained in the FAPI message body) and has the structure described in figure 3.7, which is parsed in the Trace Visualizer

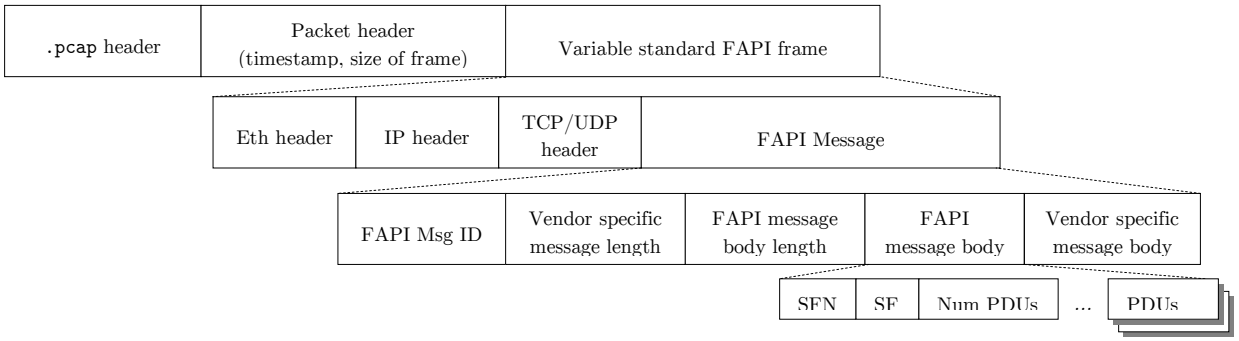


Figure 3.7: `.pcap` structure used for parsing

Other logs generated by a BlackBox test are:

- `bsc-serial.log` - text file containing the standard output captured from the serial interface of the board
- `execution.log` - text file containing Job Runner's standard output during the execution of the test case
- `env-debug.log` - text file containing messages useful for debugging the whole test environment and implementation

The Trace Visualizer takes as input the binary file with memory dump and the packet capture file and offers support for aggregated statistics and plots for:

- Number of DSP cycles consumed for each Layer 1 specific job
- General overview of MAPLE and DSP Load using statistics about job operations and cycles (e.g. operations used by StarCore DSP or MAPLE)
- Code efficiency: if Layer 1 developers are using the full advantages of the hardware or if one processing is completely inefficient done in the Layer 1 software (if an optimization can be done)
- Physical Layer radio parameters extracted from FAPI and correlation with hardware and software performance
- Number of Cache Hits and number of Cache Misses

Also, for debugging the communication between Layer 1 and Layer 2, the emulated Layer 2 from Power Architecture sends raw FAPI packets over the network and on the tester's PC it is necessary to start sniffing the network traffic (using tools such as Wireshark) in order to capture the packets and save them to a capture file. The FAPI packets sent over the network contain the FAPI configuration sent from the test script, the actual FAPI message sent by Layer 2 to Physical Layer, and the FAPI messages which came from Layer 1 to Layer 2.

Author's contribution and focus for this thesis were on the FAPI packet capture, especially for debugging the testing environment. In the next chapter a case study of downlink traffic testing, using BlackBox approach, what FAPI message flow should be exchanged between the emulated Layer 2, Layer 1 and the test mobile and what is the output of the test case using the automated steps described in figure 3.5. In the very last sections of the chapter there are some plots generated by the Trace Visualizer using the logs generated during the downlink traffic testing.

Chapter 4

Monitoring Downlink Traffic Generated using BlackBox Environment

This chapter provides further detail on how the environment described in sections 3.3, 3.4 and in figure 3.6 works, the expected inputs and outputs from both: a BlackBox test and the Trace Visualizer, using a downlink scenario as test case.

4.1 Scenario

Let's imagine the following real-life scenario: an user in a cell wants to load on his mobile phone a videoclip from YouTube, Vimeo, Trilulilu or other similar services, or he wants to watch a live stream.

The very first step the mobile phone does is exiting the idle state (it wakes up) and then staring a first handshake with the base station. Basically, the steps from figure 2.9 are involved in this first message exchange, but also the MME and HSS will be acknowledged about this request (this acknowledgment is usually used in the billing process by the network operators). The base station will send the initial message from the mobile phone to MME which will then send an update location request to HSS and then receive an acknowledgment from HSS.

Then, the base station will send to the serving gateway a request for creating a session for the mobile phone and default bearers will be activated. Once the serving gateway sends an acknowledgment, then the attach of the user is accepted.

Once the attach to the network is done, the application has the possibility to request the content of the webpage. Before opening a TCP connection, using the well known three way handshake with the video service, a DNS (Domain Name Service) lookup is done.

Many errors may occur in the above scenario, and we want to know for sure which of them can have the source in the Physical Layer. From a Layer 1 perspective, video streaming is a downlink scenario (on the uplink side we only find the request from the mobile phone to load the webpage)

This means that we want to know what does the base station transmits (what the PDUs and the transport blocks contain) to the user, if the transmission is done correctly, the transmission quality, identify probable cause for transmission errors and many other information.

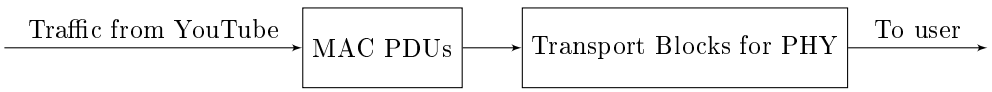


Figure 4.1: Downlink processing in eNodeB at L2/L1

The Physical Layer software offers services to MAC in the form of transport channels (DLSCH, BCH, MCH,

PCH), as described in figure 2.11 and it should basically end up in mapping the resource elements [1] in time and frequency for each user, according to the information received from Layer 2, from the flow described in figure 4.1. Using a 20 MHz bandwidth up to 100 resource blocks can be used.

4.2 Downlink processing chain in Physical Layer

This section briefly describes the processing which should be done in a Layer 1 software, outlining a more detailed view about the possible faulty points in the Layer 1 side. The steps described below are mapped to Layer 1 jobs and they are usually triggered when a downlink message comes from FAPI via IPC.

In downlink processing, in every TTI, corresponding to one subframe (which has 1ms), up to maximum two transport blocks are sent to the physical layer, in order to be transmitted over the radio channel. To each transport block there is a CRC attached for error detection. These transport blocks represent a sequence of the PDUs received from MAC layer containing user's traffic with any header attached from higher layers. [EDS11, p. 144-145] In a BlackBox scenario the content of the transport blocks is irrelevant, as it is directly given as input the content of the transport block. Downlink processing chain is summarized in figure 4.2.

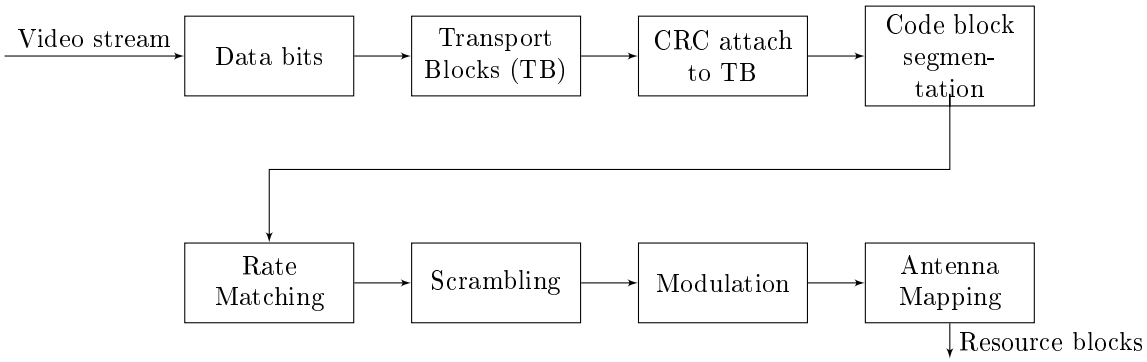


Figure 4.2: Downlink processing chain in Physical Layer

A large transport block (which was previously attached a CRC) is generally divided in multiple code blocks of maximum 6144 bits length [2], using a technique called Turbo Coding, which is used for forward error correction. Each such code block has a CRC attached. Thus, multiple redundancy bits result. Next step is Rate Matching, whose main task is to decide which of all the bits will be transmitted within a given TTI. Problem for Rate Matching can be stated like this: *"There are y redundant bits and x useful bits. On the air interface only z bits can be transmitted. What is the relationship between $x+y$ and z?"*. Rate Matching decides what redundancy bits should be discarded (or if $z > x+y$ more redundancy bits might be added in this step) and it operates on the entire transport block, not on a code block. The information about the final transport block size (with CRC attached) is provided to the UE as part of the scheduling response came from the scheduling request the UE made to the eNodeB. This information is transmitted over PDCCH (Physical Downlink Control Channel). The UE can use the scheduling response to determine the number of code blocks and their size.

Once Rate Matching is done, the next step is randomizing the interference using scrambling. This step is necessary in order to combat the effects of multipath propagation, as it is described in section 2.1. Moreover, the interference can occur due to other station's signals that are in the air. When scrambling is done, the data is modulated. In LTE, the most common modulation schemes are QPSK, 16QAM, 64QAM. The bits are thus transformed in symbols which carry two, four or respectively six bits. The physical layer has now to map the symbols corresponding to the transport blocks to the set of eNodeB's antenna ports [3] used for

[1] A resource element is the smallest physical resource in LTE. It is made out of one subcarrier during a symbol. Resource elements can be grouped in resource blocks. One resource block contains twelve subcarriers in frequency domain and can contain up to 84 resource elements if normal cyclic prefix is used. A brief overview of LTE resource grid is provided in appendix A

[2] 3GPP 36.212, section 5.1.2 states that *"The maximum code block size is $Z = 6144$"* http://www.3gpp.org/DynaReport/36212.htm

[3] 3GPP 36.211, section 5.2.1 states that: *"An antenna port is defined such that the channel over which a symbol on the antenna port is conveyed can be inferred from the channel over which another symbol on the same antenna port is conveyed. There is one resource grid per antenna port. The antenna ports used for transmission of a physical channel or signal depends on the number of antenna ports configured for the physical channel or signal."* http://www.3gpp.org/DynaReport/36211.htm

transmission. Finally, these symbols from each antenna port will be mapped to a set of resource blocks assigned by the MAC scheduler for the transmission. That is, for each antenna port the base station has to do a mapping to the time-frequency domain based on the information from MAC. More details about downlink physical layer processing can be found in [EDS11, ch. 10] and in 3GPP TS36.212, version 8.8.0, sections 5.3.2 and 5.3.3, release 8[1].

4.3 Downlink transmission testing

4.3.1 BlackBox message flow in a test case

In the tests done for validating the infrastructure of this thesis, support for correct transmission in downlink was enabled, and it has two major components: control information and user data. These components can be found in the communication between Layer 2 and Layer 1 in the following points:

- DLSCH - in which a random payload is sent from BlackBox (normally, here is found user's traffic, the data he receives from YouTube)
- DCI - Downlink Control Information , which carries the information that the UE should always decode. DCI contains information about what resource blocks carry the requested data from YouTube and what demodulation scheme should the UE use to decode the data. Consequently, DCI contains two information: uplink resource allocation, persistent and non-persistent, and parameters describing downlink data transmitted to a particular UE. The UE will have to decode DCI in order to know how to decode the data addressed to him contained in DLSCH. It is also important to know that DCI has various formats for the information in order to define the type of the resource allocation (in which way the scheduler allocates resource blocks for each transmission).
- BCH - which is sent regularly (at every 40 ms) by the base station, generally containing cell info as discussed in 2.2

A multi UE generic downlink test case, defined as a function in a script should have the following signature (additional variables can occur if scenario becomes complicated):

```
1 def MUE_downlink_generic(ue_per_tti, total_UEs, scheduling_enabled,CQI_enabled
    , transmission_mode, dl_mcs, num_resource_blocks_dl, force_retx, max_retx)
```

Multiple calls of this function, with different parameter values can be placed in a text file which is parsed by the Job Runner and thus both function definition and text file should reside on the Test Execution PC. The above function sends commands to the emulated Layer 2 in form of FAPI `DL_Config.request` messages, which contain two types of PDUs:

- BCH PDU
- DCI PDU with content depending on DCI format, associated to an incoming DLSCH PDU
- DLSCH PDU with data for a given UE (identified by RNTI)

In this test it is assumed that the test mobile properly decodes DCI. Also, within every `DL_Config.request` message a `TX.request` message is sent from Layer 2 to Layer 1. The downlink message flow would be similar to the one in the figure 4.3. Listing C.1 from appendix C contains an algorithm of a downlink test case which should be ran using BlackBox, following this message flow.

[1] http://www.etsi.org/deliver/etsi_ts/136200_136299/136212/08.08.00_60/ts_136212v080800p.pdf

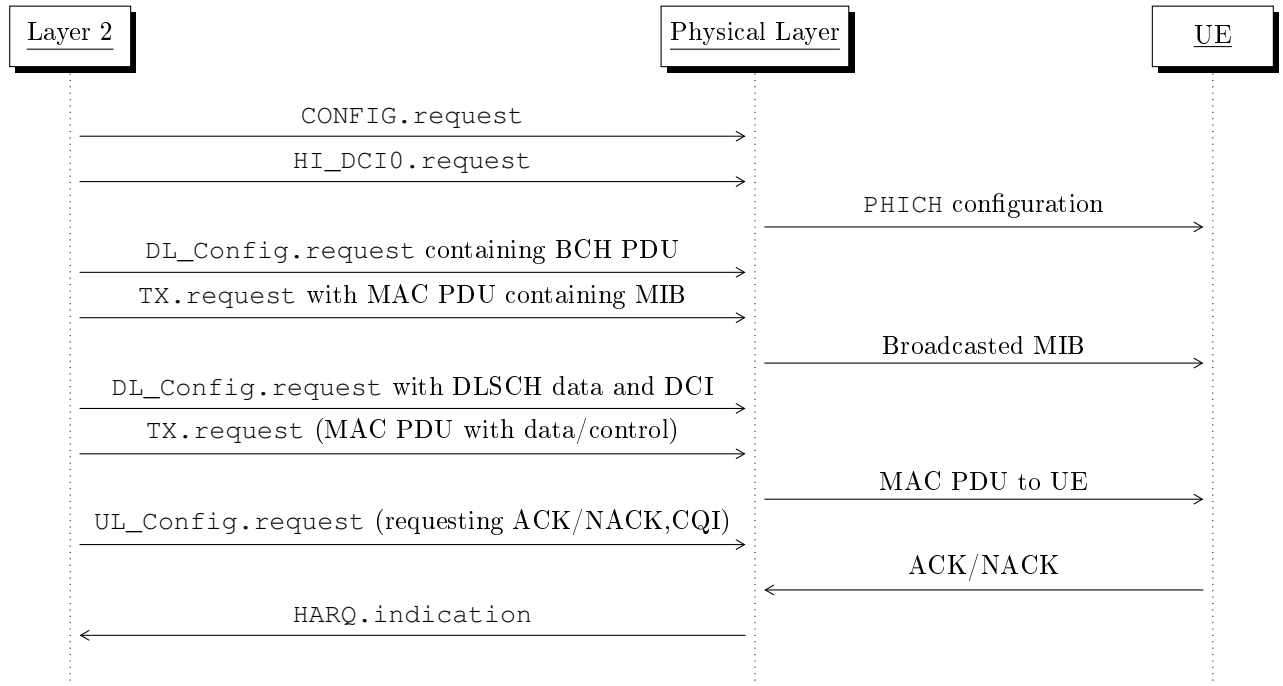


Figure 4.3: Downlink FAPI Message Flow for a TTI

The two initial messages from the BlackBox message flow described in figure 4.3 have the following purpose:

- **`CONFIG.request`** - used to specify cell parameters, such as duplexing mode used to separate downlink and uplink transmission (FDD or TDD) or the bandwidth, specified in number of resource blocks, for downlink or uplink, or signal's reference power for transmission. Revisiting figure 2.10 which described the Physical Layer as a finite state machine, this message could be considered that realizes the transition from the idle state to the configured state of the Physical Layer or reconfigures the Physical Layer, maintaining him in the same state.
- **`HI_DCI0.request`** - used by the emulated Layer 2 to send ACK or NACK responses for the MAC PDUs received from the UE on ULSCH. Then the Physical Layer informs the UE about the HARQ acknowledgments configuration in this cell and then in every first symbol of every subframe, the UE will find PHICH (Physical HARQ Indication Channel) with ACK or NACK. Information about PHICH will be regularly sent to the UE via BCH.

4.3.2 Extracting Data from Logs using Trace Visualizer

4.3.2.1 Radio Parameters Statistics from FAPI

The `UL_Config.request` can have a PDU depending on what information wants Layer 2 to receive from the UE (or what information is the UE scheduled to send):

- data and ACK/NACK (acknowledgment or not acknowledgment) response
- ACK/NACK response
- CQI report and ACK/NACK response

Layer 2 receives the ACK/NACK from UE in the form of `HARQ.indication` message from physical layer.

All FAPI messages, such as `DL_Config.request`, `TX.request` (which contain information about downlink communication) or `UL_Config.request` (which contains information about the uplink communication) are found in a packet capture file, called `FAPI.pcap`, which results from capturing, using Wireshark, the communication between Layer 2 running on Power Architecture and Layer 1 running on StarCore DSP.

Listing 4.1 shows raw output of the Trace Visualizer with values of data extracted from a `DL_Config.request` after parsing the packet capture file with FAPI messages `FAPI.pcap`. Listing 4.2 shows the values of one DLSCH PDU and one DCI PDU.

```
dl_info{1041} =
       sfnsf: 6242
    num_pdus: 8
    the_pdus: {8x1 cell}
```

Listing 4.1: Downlink Information extracted from `DL_Config.request`

```
dci_dl_pdu =
                pdu_type: 0
            pdu_type_str: 'DCI_DL_PDU'
                pdu_size: 24
              dci_format: 0
               cce_index: 31
                    rnti: 100
       aggregation_level: 1
     res_allocation_type: 0
                  mcs_1: 28
    redundancy_version_1: 0
              rb_coding: 491520
    new_data_indicator_1: 1
        harq_process_num: 2
                     tpc: 1
     dl_assignment_index: 0
                tx_power: 6000
              rnti_type: 1
                padding: 0

dlsch_dl_pdu =
                    pdu_type: 3
                pdu_type_str: 'DLSCH_PDU'
                    pdu_size: 36
                dlsch_pdu_len: 1479
                    pdu_index: 0
                         rnti: 100
          res_allocation_type: 0
          vrb_assignment_flag: 0
                    rb_coding: 491520
                          mcs: 6
           redundancy_version: 0
             transport_blocks: 1
     tb_to_codeword_swap_flag: 0
          transmission_scheme: 1
                num_of_layers: 2
              num_of_subband: 0
         ue_catagory_capacity: 3
                           pa: 4
    delta_power_offset_aindex: 0
                       n_gap: 0
                       n_prb: 0
                     tx_mode: 1
                     padding: 0
           num_rb_per_subband: 0
                num_bf_vector: 0
```

Listing 4.2: Sample DLSCH and DCI PDUs from a `DL_Config.request` message

Information about the transport block size can be extracted from `TX.request` messages, and it is stored in form of `tag_len`. A `TX.request` message can contain multiple tags. Listing 4.3 shows sample content of a `TX.request` message.

```
tx_request_structure =
        pdu_len: 16
      pdu_index: 0
    num_of_tlvs: 1
       the_tlvs: {[1x1 struct]}
tlv =
        tag: 1
    tag_len: 1479
      value: 1.349769216000000e+009
```

Listing 4.3: `TX.request` sample values

Previous examples showed sample content for one message. Layer 2 and Layer 1 exchange multiple FAPI messages in time (measured in SFN and SF) as figure 4.4 with the content described above. As it is impossible to print (or to dump to a file) the structures as they were displayed in listings 4.1, 4.2 or 4.3, the Trace Visualizer dumps a table (which further can be used for plots of data per RNTI, that is per one user) with only the most relevant information for a tester or for a Layer 1 software developer interested in debugging the downlink processing. A sample excerpt from a downlink table dumped by the Trace Visualizer is shown in listing 4.4. Columns contain information extracted from DLSCH and DCI PDUs and contain respectively:

- Timing information: System Frame Number (SFN) and Subframe (SF)
- For each user, identified by RNTI:
  - Transmission Scheme (what type of MIMO was used) and Transmission Mode
  - Modulation type (can be only QPSK, 16QAM or 64QAM)
  - MCS (Modulation and Coding Scheme) for each transport block - by knowing this value, the modulation type, we can use table 7.1.7.1 - 1 from 3GPP (see appendix B) and using transport block size one can determine the number of physical resource blocks used for the transmission in that TTI.
  - Resource Allocation Type (information about how the scheduler at Layer 2 allocates resource blocks for current transmission)
  - Resource Block Coding - What encoding was used for the resource blocks. This value depends on the Resource Allocation Type and should match value sent in the DCI PDU which allocated current grant for transmission.
  - If retransmission occurred, by using the values of `HARQ_Process_Number` and `NDI` (New Data Indication)
  - Transport Block Size (or length, in bits), which indicates the number of bits that can be transmitted in that TTI (or in that SF)

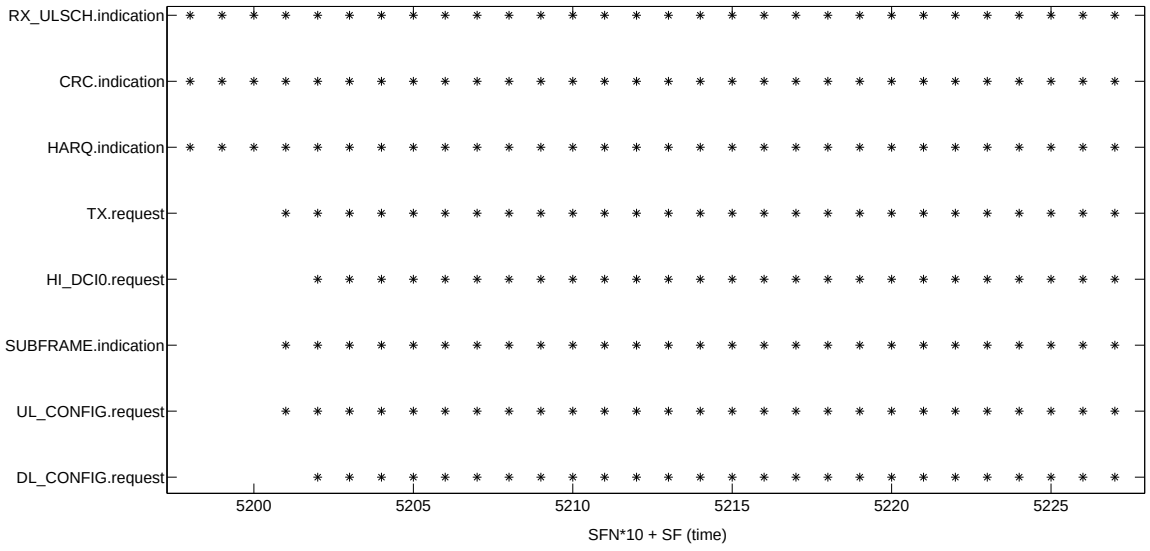


Figure 4.4: FAPI Messages exchanged between Layer 2 and Layer 1 in a BlackBox test

```
# Downlink Map from FAPI PCAP
#
# RNTI SFN/SF Modulation MCS TxScheme TxMode RAT RB_Coding HARQ_Process_Number
    NDI TB Size
@
...
92  9344    6    18   1   1   0   262144  0   0   277
93  9344    6    18   1   1   0   131072  0   0   277
94  9344    6    18   1   1   0   65536   0   0   277
95  9344    6    18   1   1   0   32768   0   0   277
96  9345    6    18   1   1   0   262144  1   1   277
97  9345    6    18   1   1   0   131072  1   1   277
...
```

Listing 4.4: Sample Downlink Traffic Map generated by Trace Visualizer

To conclude, by knowing the resource block coding (which is a bit map in which the resource blocks used are set to 1) and the resource allocation type we can determine the number of physical resource blocks N_{PRB}. Once N_{PRB} is known, by knowing the MCS extracted from a packet capture, one can use table 7.1.7.2.1-1 from 3GPP (an excerpt is provided in appendix B) to determine the transport block size, denoted TBS. This value can be used for the following things:

- Compare with the Transport Block Size extracted from `TX.request` and decide if coverage is sane
- Use TBS to compute the downlink peak data rate by knowing the transmission mode and transmission scheme. For instance if 4x4 MIMO is used, peak data rate will be $4 \cdot TBS$ bits per second. Hence, one can obtain the maximum theoretical speed in a 4G connection in a given spectrum allocation.

Usually, the above extracted parameters, using a BlackBox test, are prior known (as tester's have direct access to input such values), but a Layer 1 software developer, who has no previous knowledge about what is defined in BlackBox scenario, and only clicks buttons in the web interface, should have some aggregated data from a packet capture file. Also these values can be used to debug the whole BlackBox environment, because if a tester gives as input, for instance, a FAPI `DL_Config.request` containing MCS 18 and 16QAM modulation ends up in the emulated Layer 2 or in Layer 1 with MCS 18 and 64QAM or MCS 26 and 16QAM modulation (which is an invalid combination), then there is an error in the environment and it should be fixed.

Also, if one is interested in knowing if retransmissions occurred, then it is important to know the HARQ Process number and the New Data Indication (NDI) whose value can only be 0 or 1. From the downlink map provided in listing 4.4 Trace Visualizer displayed the *"retransmission map"* from figure 4.5, in which blue is the transport block size, green is NDI and red is the HARQ process number. Using this map, one can easily determine the SFN and SF (moment in time) when a retransmission occurred, using figure 2.8 described in section 2.3. Also, this figure shows the number of users who where scheduled to transmit in a particular TTI. In a BlackBox test, the tester has full access to control this scheduling and that is why the plots from figure 4.5, which represent a downlink BlackBox test, have exactly four users transmitting in a TTI. In a full protocol test, the number of users who are scheduled to send or receive data at a moment in time is unknown.

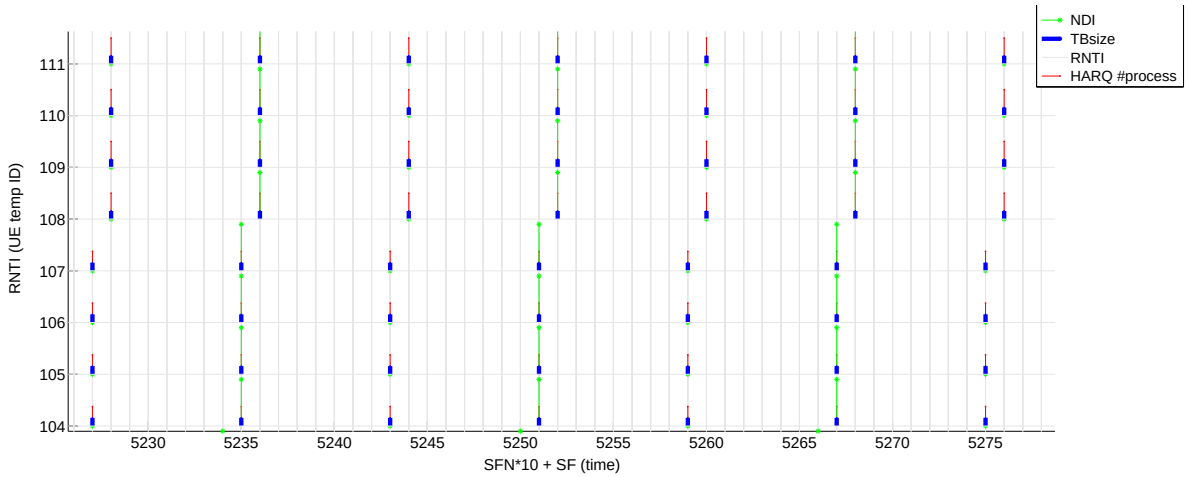


Figure 4.5: Retransmission map, using HARQ information from FAPI

4.3.2.2 Layer 1 Processing Statistics extracted from System Memory

The file parsed for finding information about Layer 1 Processing Chain done in StarCore DSP is the binary file `L1trace.dat` containing the dump of the system memory, with specific print calls from the Layer 1 software. The Trace Visualizer executes a parsing of this binary file and generates a specific buffer with readable text which can be then inserted in either plain text files, in a spreadsheet or can be used directly in the Trace Visualizer as vectors for plots. For this thesis, focus was on aggregation of statistics generated by the binary file parsing and correlation with radio parameters.

The Layer 1 Software executes multiple jobs (as functions) depending on messages came from Layer 2. One interesting thing to know is how many cycles were used for executing a job (minimum number used, average number of cycles during all the TTIs in the test case and the maximum number of cycles used). In figure 4.7 the time axis is measured in symbols, but it can be easily converted to TTIs (SFN and SF) knowing that in LTE in one subframe there are 14 symbols transmitted, in normal cyclic prefix mode. For our downlink processing chain we are interested in how many cycles were needed for `PDSCH EDF` and also the `JOB BUILDER`, which creates the Layer 1 jobs with parameters came via FAPI.

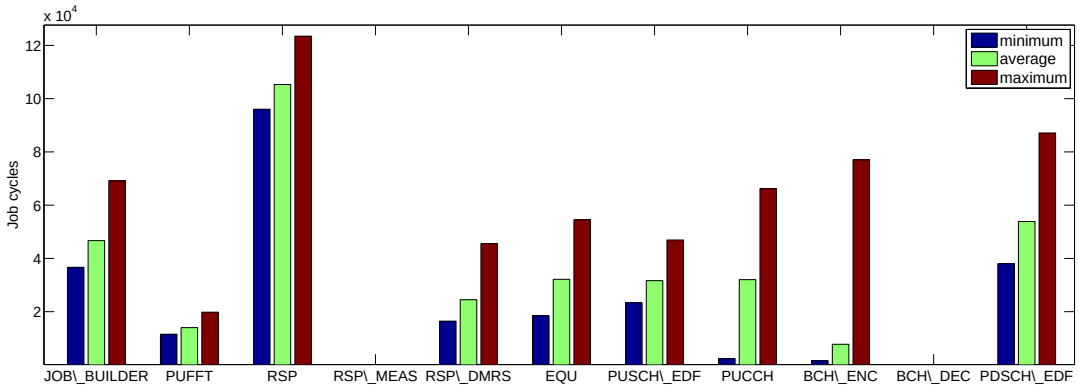


Figure 4.6: Layer 1 Jobs Number of Cycles Statistics

Also, for all the Layer 1 jobs, it is possible to draw the symbols that triggered a specific job in time. Figure 4.7 shows an excerpt of how job execution in time is plotted by the Trace Visualizer. This image can be used together with the image that shows what FAPI messages were exchanged between Layer 2 and Layer 1 (figure 4.4) and the fact that, for instance, a downlink job will be built only after one subframe from the moment that an `UL_Config.request` message was received, one can easily know what jobs were built in Layer 1 software and what FAPI message from Layer 2 triggered this execution. Table 4.1 shows the timing differences between Layer 1 log messages and FAPI packets, which are dynamically extracted for each test

case, and these differences are useful for making a synchronization between Layer 1 trace and FAPI trace (Δt denotes the difference in subframes between the moment a FAPI message came from Layer 2 and Layer 1 builds the job for that particular FAPI message).

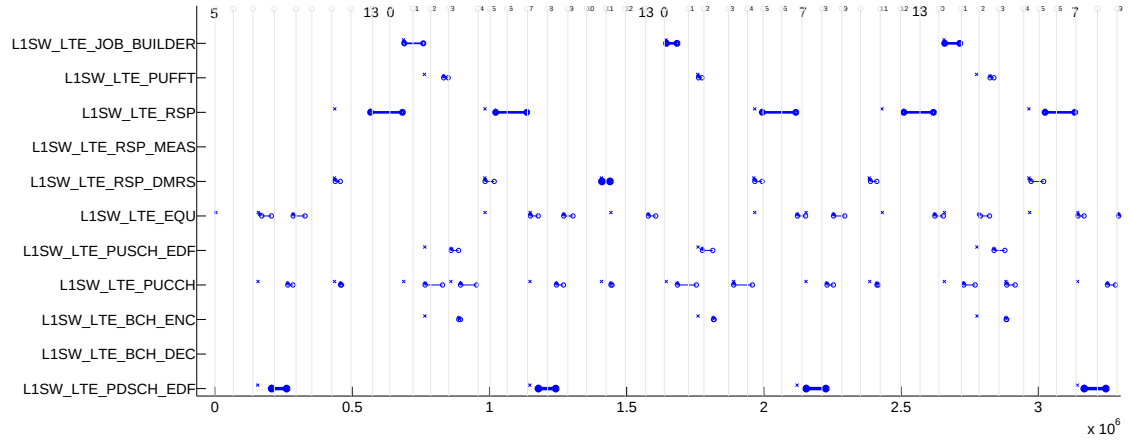


Figure 4.7: Layer 1 software Jobs Time Chart

FAPI Message	Δt
DL_CONFIG.request	0
UL_CONFIG.request	1
HI_DCI0.request	0
TX.request	0
HARQ.indication	4
CRC.indication	0

Table 4.1: Subframe difference between when a FAPI Message occurs and when a Layer 1 job is built

From the system memory, for the entire test case we can extract the memory load, and draw an evolution in time, for each symbol:

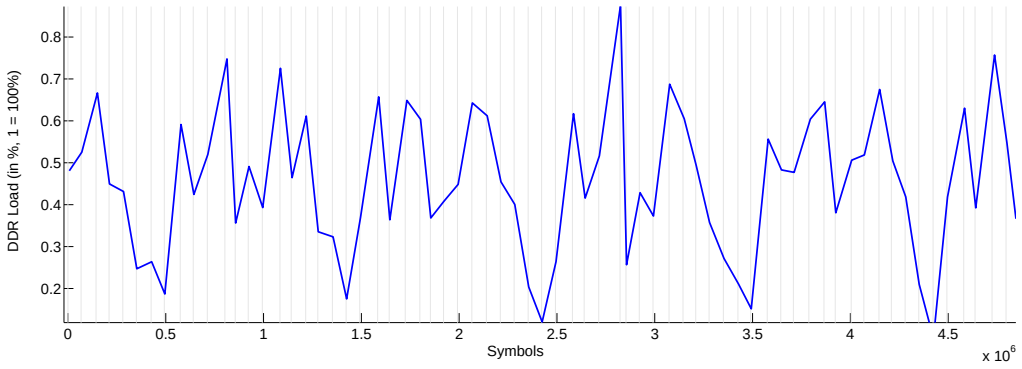


Figure 4.8: Layer 1 software Memory Load (percents) for a Test Case

Chapter 5

Conclusion

This thesis presented a project which is extremely useful in finding Layer 1 faulty behavior and thus helping the developers of Freescale Layer 1 software for small cells base station in easier integration with other third parties who provide higher layer solutions and easier debugging and profiling.

Major achievements in development of this project are:

- Development of a generic automated testing environment and framework.
- Use a custom Layer 2, which emulates the functionality found at this level for a faster determination of Layer 1 errors and performance.
- Deployment of additional modules, useful for a fully remote access to hardware equipments (base station and test mobile) and automation (Web Server, Test Execution PC, Database, Storage for Logs). By using this approach, any developer who has access to Freescale network can easily access a web interface and trigger jobs to test his own solution.
- Detailed information about a faulty behavior: anyone interested in knowing more precisely the root cause of a failure of a test using BlackBox environment has full access to verbose logs and packet capture, displayed in the web interface for download, and on an external storage for bulk analysis.
- By using the Trace Visualizer, a Layer 1 developer can activate various flags to display plots of the points that he is interested to monitor offline. Consequently, he does not have to read gigabytes of verbose logs, which might contain irrelevant information for him. This tool is capable of aggregating data from multiple logs, so one could directly input folders from the external storage where logs are stored, without the need to browse the web interface and manually download logs for providing input. Again, it is only necessary just to have grants for reading permissions in an access control list.

Chapter 6

Further Development

The project has two important known limitations:

- Although the testing environment is fully automated, user has to do two manually steps, which require some knowledge about the setup:
  1. Click in the web interface to trigger a test. This results in running a set of pre-defined test cases. If the developer has a custom scenario, he has to manually write the script and place it on the Test Execution PC.
  2. Have previously installed dependencies for Trace Visualizer in order to view the plots. For solving this step, it is necessary to integrate with BlackBox Testing Solution and once a test case is done, plots should be dumped as images and/or as `.eps` files (or any other user friendly format) for displaying in documents and reports.
- Logs parsed by the Trace Visualizer are generated during a test case. This means that we are monitoring Layer 1 software parameters and radio parameters offline (after the test was done). It would be great to develop support for real time analysis, similar to `top` utility from Linux with two windows: one for hardware parameters and one for radio parameters.

Appendices

Appendix A

Overview of LTE Resource Grid

A.1 Terms definition

- **Subcarrier** is the smallest division of LTE spectrum which carries data
- **Resource element** is considered a modulated subcarrier (during one symbol). This is the smallest logical unit of the LTE spectrum.
- **Physical Resource Block** is the smallest unit used by the scheduler at Layer 2. It consists of 12 resource elements which are adjacent on the frequency axis (in time-frequency representation, as provided in figure A.1)

A.2 Known facts

- One subcarrier (or one resource element) has a bandwidth equal to $\Delta f = 15kHz$ in normal and extended cyclic prefix mode.
- For extended cyclic prefix mode it is defined a special value for subcarrier spacing $\Delta f = 7.5kHz$. Table A.1 shows the cyclic prefix configuration, number of subcarriers for a resource block (denoted N_{sc}^{RB}) and number of downlink symbols (denoted N_{sym}^{DL}).

Configuration	Bandwidth	N_{sc}^{RB}	N_{sym}^{DL}
Normal cyclic prefix	$\Delta f = 15kHz$	12	7
Extended cyclic prefix	$\Delta f = 15kHz$	24	6
	$\Delta f = 7.5kHz$		3

Table A.1: Cyclic prefix configuration, 3GPP 36.211, section 6.2.3

- If one physical resource block has 12 resource elements in frequency domain, then it has a bandwidth equal to $12 \cdot 15 = 180kHz$. In time domain, a resource block is equal to one slot of length $0.5ms$.
- Transmission bandwidth is defined as the number of active resource blocks. If number of resource blocks is increased then the bandwidth increases. Table A.2 shows the maximum number of resource blocks that can be used for transmission for a given channel bandwidth.

Channel Bandwidth (MHz)	N_{RB}	Maximum occupied bandwidth ($180kHz \cdot N_{RB}$, in MHz)
1.4	6	1.08
3	15	2.7
5	25	4.5
10	50	9
15	75	13.5
20	100	18

Table A.2: Transmission Bandwidth Configuration

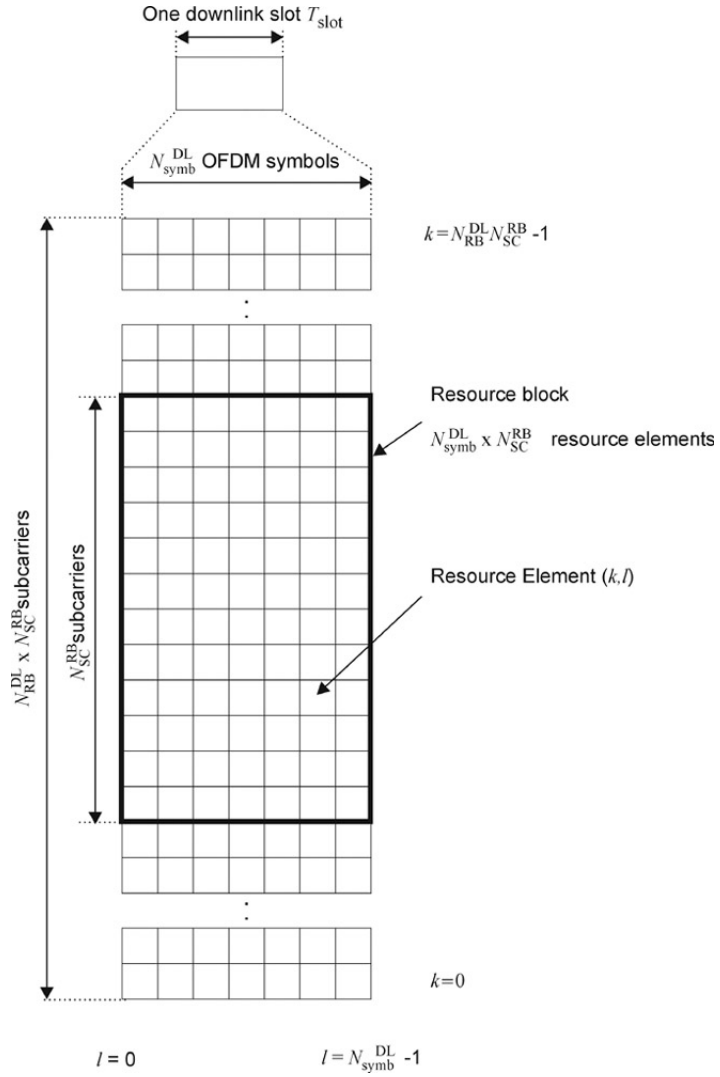


Figure A.1: Downlink LTE Resource Grid Overview, 3GPP 36.211 [3GP11a] section 6.2.2

Appendix B

3GPP Tables for Determining Transport Block Size and Physical Resource Blocks

Table 7.1.7.1-1 from 3GPP TS 36.213 [3GP11b] version 11.2.0 Release 11: Modulation and TBS (Transport Block Size) index table for PDSCH (Physical Downlink Shared Channel)

Table 7.1.7.1-1: Modulation and TBS index table for PDSCH

MCS Index I_{MCS}	Modulation Order Q_m	TBS Index I_{TBS}
0	2	0
1	2	1
2	2	2
3	2	3
4	2	4
5	2	5
6	2	6
7	2	7
8	2	8
9	2	9
10	4	9
11	4	10
12	4	11
13	4	12
14	4	13
15	4	14
16	4	15
17	6	15
18	6	16
19	6	17
20	6	18
21	6	19
22	6	20
23	6	21
24	6	22
25	6	23
26	6	24
27	6	25
28	6	26
29	2	reserved
30	4	
31	6	

By knowing the I_{TBS} and number of physical resource blocks N_{PRB} from the previous table, we can determine the transport block size in bits, using the following look-up table:

Table 7.1.7.2.1-1: Transport block size table (dimension 27×110)

I_{TBS}	N_{PRB}									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	16	32	56	88	120	152	176	208	224	256
1	24	56	88	144	176	208	224	256	328	344
2	32	72	144	176	208	256	296	328	376	424
3	40	104	176	208	256	328	392	440	504	568
4	56	120	208	256	328	408	488	552	632	696
5	72	144	224	328	424	504	600	680	776	872
6	328	176	256	392	504	600	712	808	936	1032
7	104	224	328	472	584	712	840	968	1096	1224
8	120	256	392	536	680	808	968	1096	1256	1384
9	136	296	456	616	776	936	1096	1256	1416	1544
10	144	328	504	680	872	1032	1224	1384	1544	1736
11	176	376	584	776	1000	1192	1384	1608	1800	2024
12	208	440	680	904	1128	1352	1608	1800	2024	2280
13	224	488	744	1000	1256	1544	1800	2024	2280	2536
14	256	552	840	1128	1416	1736	1992	2280	2600	2856
15	280	600	904	1224	1544	1800	2152	2472	2728	3112
16	328	632	968	1288	1608	1928	2280	2600	2984	3240
17	336	696	1064	1416	1800	2152	2536	2856	3240	3624
18	376	776	1160	1544	1992	2344	2792	3112	3624	4008
19	408	840	1288	1736	2152	2600	2984	3496	3880	4264
20	440	904	1384	1864	2344	2792	3240	3752	4136	4584
21	488	1000	1480	1992	2472	2984	3496	4008	4584	4968
22	520	1064	1608	2152	2664	3240	3752	4264	4776	5352
23	552	1128	1736	2280	2856	3496	4008	4584	5160	5736
24	584	1192	1800	2408	2984	3624	4264	4968	5544	5992
25	616	1256	1864	2536	3112	3752	4392	5160	5736	6200
26	712	1480	2216	2984	3752	4392	5160	5992	6712	7480

Figure B.1: Table 7.1.7.2-1 from 3GPP TS36.213 version 11.2.0 Release 11 used for determining Transport Block Size

The full 27 × 100 table can be found in 3GPP TS 36.213 version 11.2.0 Release 11, section 7.1.7.2.

Appendix C

Downlink Test Case Algorithm for BlackBox

```
1
2  def MUE_downlink_generic(ue_per_tti,
3          total_UEs,
4          scheduling_enabled,
5          CQI_enabled,
6          transmission_mode,
7          dl_mcs,
8          modulation_type,
9          num_resource_blocks_dl,
10         force_retx, max_retx):
11
12         # Create new basestation and testmobile objects for current test case
13         bs = new basestation()
14         tm = new testmobile()
15
16         # Create a new FAPI object containing parameters for current test case
               on downlink and uplink
17         fapiConfig = new FAPIConfig()
18
19         fapiConfig.setBCH(cell_id,
20         rnti,
21         dl_mcs,
22          modulation_type,
23          num_rb,
24          rb_coding,
25          resource_allocation_type)
26
27         fapiConfig.setHIDCI0(cell_id,
28                 rnti,
29                 dl_mcs,
30                 modulation_type,
31                 num_rb,
32                 rb_coding,
33                 resource_allocation_type)
34
35         # Can be only QPSK, 16-QAM, 64-QAM
36         fapiConfig.setULModulation(modulation_type)
37
38         # mcs_idx index is in range 0 to 31, according to 3GPP 36.213, table
               7.1.7.1-1
```

```
39          ul_mcs = fapiConfig.setULMcs(mcs_idx)
40
41          fapiConfig.setULSCH_PDU(payload,
42                  payload_len,
43                  rnti,
44                  ul_mcs,
45                  modulation_type,
46                  num_rb,
47                  rb_coding,
48                  resource_allocation_type,
49                  ul_tx_mode)
50
51          bs.setHARQinfo(num_processes, ndi)
52
53          fapiConfig.addDCI(dci_format,
54           rnti,
55           dl_mcs,
56           cqi_params,
57           extra_params)
58
59          fapiConfig.addDLSCH(rnti,
60           dl_mcs,
61           modulation_type,
62           num_rb,
63           rb_coding,
64           resource_allocation_type,
65           dl_tx_mode)
66
67          bs.configureFAPI(fapiConfig)
68          bs.start()
69
70          # Start configuring test mobile
71          tm.init()
72          tm.configBCH(cell_id)
73          foreach ue in current cell:
74                  rnti = tm.setUE_context()
75                  tm.configULparams(ue, cell_id)
76                  tm.configHARQ(num_processes, ndi)
77                  for i in range(0, num_processes):
78                          tm.decodeDCI()
79                          tm.readDLSCHdata()
80                          tm.configureNoiseSimulation()
81
82          # Measure CRC error statistics provided by base station
83          bs.getCRCerrors()
84
85          # Measure HARQ retransmissions
86          bs.getHARQ_ReTxStats()
87
88          # Measure statistics from TestMobile
89          tm.getAllStats()
```

Listing C.1: Sample downlink test case using BlackBox

Bibliography

[3GP11a] 3GPP. TS36.211. http://www.3gpp.org/DynaReport/36211.htm, 2011.

[3GP11b] 3GPP. TS36.213. http://www.3gpp.org/DynaReport/36213.htm, 2011.

[Aus13] Ori Auslender. Small Cell Concept and LTE Small Cell Implementation on BSC9132. http://cache.freescale.com/files/training/doc/dwf/DWF13_EUF_NET_T1278.pdf, November 2013.

[dTdC11] Centre Tecnologic de Telecomunicacions de Catalunya. LTE-EPC Network Simulator (LENA). http://lena.cttc.es/manual/lte-design.html#harq, 2011.

[EDS11] Stefan Parkvall Erik Dahlman and Johan Skold. *4G LTE/LTE-Advanced for Mobile Boradband*. Academic Press, Oxford, United Kingdom, 2011.

[For10] Small Cell Forum. LTE eNodeb L1 API Definition. http://www.smallcellforum.org/smallcellforum_resources/pdfsend05.php?file=LTE%20eNB%20L1%20API%20Definition.pdf, 2010.

[Gol05] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, Cambridge, United Kingdom, 2005.

[Jag13] Prof. Aditya K. Jagannatham. Advanced 3G and 4G Wireless Mobile Communications. Online Video Course, http://nptel.ac.in/courses/117104099/, 2013.

[MB14] Mihnea Ionescu Mihai Barbulescu, Andrei-Alexandru Enescu. A Black Box Approach to Physical Layer Validation for 3G/4G Base Stations. *10th International Conference on Development and Application Systems, Suceava, Romania, May 15-17*, pages 161–164, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6842447, 2014.

[Sch03] Jochen Schiller. *Mobile Communications, Second Edition*. Addison Wesley, Addison Wesley, 2003.

[ST07] Javier Sanchez and Mamadou Thioune. *UMTS*. ISTE Ltd, United Kingdom, 2007.