Final thesis

Test Data Post-Processing and Analysis of LA

by

Paul Nedstrand & Razmus Lindgren

LITH-IDA-EX-2015/THESISNUMBER

December 19, 2014

Final thesis

Test Data Post-Processing and Analysis of LA

by

Paul Nedstrand & Razmus Lindgren

LITH-IDA-EX-2015/THESISNUMBER

December 19, 2014

Supervisor: Ola Leifler

Examiner: Johannes Schmidt

Abstract

§ This master thesis involves developing a lightweight analyser that produce statistics from the traffic data in the communication link between a UE e.g. a cell phone and a E-UTRAN base station (the air interface). The analyser tool will produce graphs with information about the correlation between two signal data in the channel (e.g throughput over interference). From the statistics produced by the analysis tool, the testing personell at Ericsson can more easily detect potential faulty behavior from the UE or eNodeB in a much faster way than they were able to do before. This tool will also be able to help ericsson to rewrite test cases to be more indepth than they are now. To show that that it is possible to some kind of analysis with this tool, we made our own analysis on the link adaptation target BLER, this way we could validate that our tool could handle large amount of data and also that you can compare different data with each other. To be able to show that our tool is usable for Ericsson IODT we offered IODT test engineers to do tests on Link adaptation and HARQ with our tool and evaluate it with their current methods.

Contents

Acknowledgements

We want to thank our supervisor Sonia Sanghari and Roland Sevegran. for giving us the opportunity to make this master thesis available to us and for helping us with material we needed to be able to do this thesis We also want to thank the whole testing team at Ericsson IODT, especially Peter Tureby, for the help we got while doing simulations and also for giving relevant information about the LTE system.

Abbreviations

CQI	Channel Quality Index.
SINR	Signal to Interference plus Noise Ratio
eNodeB	e-UTRA Node-B...
e-UTRA	evolved UMTS Terrestrial Radio Access
e-UTRAN	evolved UMTS Terrestrial Radio Access Network
UE	User Equipment
LTE	Long Term Evolution
MCS	modulation and coding scheme
LA	Link Adaptation,
3GPP	Third Generation Partnership Project
FEC	Forward Error Correction.
OFDM	Orthogonal Frequency DeMultiplexing.
OFDM Symbol	Orthogonal Frequency DeMultiplexing symbol
DL	Downlink
UL	Uplink
BLER	BLock Error Rate
UL	Uplink

Chapter 1

Introduction

1.1 Background

The 3GPP standard was created/introduced in 1998 [source], and is a collaboration between different network distributors.The most recent step in 3GPP is called LTE (long term evolution). This was started 2008. LTE is believed to be the next step to becoming 4G and is fulfilling the most requirenmets of it. Therefore it’s often called 3.9G.

1.2 Problem Formation

Ericsson does not currently have any tool or functionality to make analysis of layer 1 and layer 2 data for testing in a more indepth way. The way IODT works is that they manually look at log files for potential problem in the link between UE and EnodeB. These files often contain a lot of data and can be quite hard to analyze, it is then difficult to see and detect potential problems. The other way they study data is to look at it in real time in a command window, which goes with a very high speed, which makes it easy to miss potentially faulty behaviour. There is hence no way to make an in-depth post analysation of data to detect problems that can occur only a few times. How can we help Ericsson IODT to do these analyses in an efficient way and in a way that helps them to detect error and faulty behaviour? How can we help testers to look at data in the link between the UE and enodeB in a more presentable way than in a log file or in real time? There is also a need to look at and compare different UE's performance to each other or the performance on the same UE with different setups, but there is no current way to do this in a presentable way. How can we make this available for Ericsson?

1.3 Our task

Our task from Ericsson is to develop a lightweight analysis tool that simply produces statistics from data traffic between UE and eNodeB (the air interface), handle several input parameters and from these parameters produce data in form of graphs.
If we had time we would construct functionality to store graph data, so that one could be able to make comparisons with other UE's performance.

To be able to construct such a program we first needed to know:
What kind of data shall we handle and produce graphs from?
What quantities of data shall the program be able to handle?
What type of statistics are necessary? (max, min average, crossvalues correlations etc)
To be able to answer those questions we needed to:

- Study and understand the LTE air interface.
- Study how data traffic is stored and handled.
- Study how data traffic is collected.
- Study the which kind of relevant data traffic parameters that might be interesting to process.

- Evaluate a suitable tool for the processing of data, e.g. MATLAB or something similar

To validate that we had implemented our tool in a correct way we set up a set of goals:

- Can our tool provide functionality that is useful to the testers?
- Can a tester perform a set of tasks faster or easier with our tool than without it?
- Is our tool easy to use?
- Are we able to perform an analysis on LA or HARQ with our tool?

1.4 Related Work

Since LTE is still in a developing stage there is no similar work done in the LTE area. We talked to our supervisors and other employees at ericsson regarding similar tools, bet there are currently non of these.

1.5 Thesis Outline

This thesis is divided into the following Parts

- Chapter 2: Theoretical background of LTE covering all the basics of how LTE is constructed and how certain parts in LTE works.
- Chapter 3: The analysis tool. In this it is descirbed how the testing works, and the motivation behind functionality of it.
- Chapter 4: the analysis. In this chapter it is described how and why we did our analysis.
- Chapter 5: In this chapter we write about a conclusion and future work.

Chapter 2

Theoretical Background

In this chapter the theoretical background is described to fully understand this thesis.

2.1 Basic Overview of LTE air interface

In this section there will be a general overview of the basics of LTE interface.

2.2 LTE and e-UTRA

LTE stands for long term evolution and is the new standard in mobile networks, it is also known as 4G but has yet to fulfill some requirements to call it self that [source]. e-UTRA stands for evolved UMTS Terrestrial Radio Access Network and is radio access network

2.3 The Physical Layer of the LTE System

The layer we have been working in is called layer 1, also known as the physical layer. It is the description of how the signals are sent through the air, how they are scheduled (typ). All the parameters that you are able to look at is in this layer.

2.4 Layer1

2.5 Layer2

2.6 Link Adaptation

Link adaptation is a way to enhance the performance in wireless systems. Dependent on the channel condition the modulation scheme and code rate is changed. The better channel condition the higher modulation scheme and higher code rate is used and the worser channel condition the lower modulation scheme and higher code rate [source]?. The modulation scheme used in the LTE systems are QPSK, 16QAM and 64QAM [source]. The codes used is QPP (quadrature polynomial permutation) turbo code [source]?. When data is sent from a base station to a UE (DL) the UE will report a CQI (channel quality indicator) value to the base station indicating how good the channel is. CQI can take values from 0 to 15 (4 bits) where each number correspond to the SINR in table X, so 0 represent a very bad channel condition and 15 a very good one. Out from this value The eNB decides a MCS (modulation and coding scheme). For downlink MCS can take values between 0-28 and uplink 0-24 [source]. Each one of these values represent a modulation scheme and code rate, were MCS = 0 has the lowest code rate and lowest modulation order (QPSK). MCS = 28 in downlink and 24 in uplink has the highest code rate and the highest modulation order (64-QAM). After the MCS is set in the eNodeB it will keep track how many
In the uplink the same principles are used but instead

2.7 HARQ Algorithm

HARQ stands for hybrid automatic repeat request and is an algorithm to handle rightly and wrongly received data. HARQ is a combination of FEC (forward error correction) and ARQ correction control [source]. When data is sent from transmitter to receiver, the data might either be correct, incorrect and correctable or incorrect and incorrectable.
If the data that is sent is correct in the receiver, it will send an ACK bit which then the transmitter receives. ACK stands for acknowledge and means that the data that was sent was understood, and the transmitter can sent new data.
If data is incorrect but correctable the wrongly received bits will be corrected an ack bits will be sent. The number of of bits that can be corrected is dependent on how many code bits you have [source]. The transmitter can send new data
If the data is incorrect and uncorrectable the receiver will send an NACK (Not ACKnowledged) bit and the data will be retransmitted. (picture on harq ack, harq corrected bit ack och en nack)

Since there will be a delay between when data is transmitted and an ACK/NACK will be received, there are different HARQ processess in LTE that are running in parallell. The number of HARQ processes in the LTE standard is set to 8 [source (air interface boken)]. This way the data wont have to halt the data before a new transmission or retransmission but can send new data immideately and resend the wrongly received bits at a later stage.

(picture on the parallell harq process)

2.8 Modulation

A modulation scheme is a way to map digital bits to analog signals in wireless systems. it is a way to represent the bits in the air. There are different modulation schemes and the ones that are used in LTE are 4QAM (QPSK), 16QAM and 64QAM. The signals are modulated in the following way
(bild pa 4QAM 16QAM and 64QAM over I-phase and Q-phase). The I-phase is basically a Sinus wave and Q-phase is a Cosine wave. What the points on the two axises represent is the the power of the signal [source]. So in the QPSK case, what the different signals will be is:
A*sinus(f*t)
A*cos(f*t)
-A*sinus(f*t)
-A*cos(f*t)
So in this case it fits 2 bits in each signal. Therefore, the different possible signals are 00, 01, 10 and 11. In 16QAM and 64QAM each signal point is represented by 4 and 6 bits respectively.
The signals in LTE are modulated with an IQ-modulator and decoded with and IQ-Demodulator TODO(write stuff about IQ-Modulator/demodulator)

2.9 Coding and Code Rate

Coding is a way to create redundancy in the data that are send. The data will consist of real data bits and coded bits. This way you are able to correct bits that are incorrect in the receiver. The more coded bits you have in your message the more errors you can detect and correct, but the slower data rate you will have. The code rate states how many bits that are coded in a message. the Code rate is between 0 and 1 and is simply $\frac{\#realbits}{\#realbits + \#codedbits}$.

example: if we have a code rate of 0.73 we have 73% real bits in the message and 27% coded bits.

2.10 MCS

MCS is a key parameter when you talk about link adaptation. MCS stands for modulation and coding scheme and it describes how that data is modulated and coded as an analog wave. In LTE there are several selections of mcs's. There is (currently) 29 different MCS's in the downlink and 25 in the uplink [source]. in LTE each MCS is used in different channel condition. In the best signal conditions the data is sent with MCS 28 in DL and 24 in UL, and in the worst they both have 0. A list of how the data is modulated and coded is below.

2.11 BLER target

BLER stands for BLock Error Rate and is in this thesis referred as the % of the wrongly received blocks in the receiver (UE or eNodeB) [source]. In the eNodeB there is a so called BLER-target. It indicated how many wrongly received bits the receiver should have. One can argue that it should be best to have a bler target on 0% or as low as possible but it would be quite expensive.You would have to have as low MCS as possible to achive this goal and hence the throughput would decrease, the throughput is something that you want to optimize as much as possible. So there is a tradeoff between MCS and BLER if you want to have the highest throughput. The higher bler-target you have set in eNodeB the higher MCS you will have but the more retransmissions you will have and the lower BLER-target you have the lower MCS you will have.

2.12 SINR

SINR stands for Signal to inteference plus noise ratio. It is defined as the powerSignal/(powerInterference + powerNoise). This is a relevant parameter when analysing Link adaptation

2.13 CQI

Like SINR

2.14 MAC layer

This layer is the layer before the Physical layer and is the lower sublayer of layer 2. This layer controls harq and transmission and retransmission. It also controlles the scedueling of data [source].

Chapter 3

The Analysis Tool

Our analysis tool was developed for IODT at Ericsson. The purpose of the tool was to analyse data in layer 1, especially LA and HARQ. It has been developed so that IODT in an easier way can do an post analysis of the data traffic between the UE and eNodeB. Some questions we will answer is:

- Can IODT testers study and analyze the layer 1 data especially link adaptation or HARQ?
- Is it easier to get a better overview with our tool for large quantities of trace data than it was before?
- Is it easier to get a better overview with our tool for smaller quantities of trace data than it was before?

The tool is mainly developed to perform analysis of data over SINR and CQI, i.e. the channel condition. But it is also able to use all parameters (that are sent between the UE and eNodeB) as an axis in graphs so that the tester can validate any data he/she wants.

3.1 Motivation for choosing Logtool

In the beginning of the project we had to make a decision on how we would create our analysis tool. What we needed to do was produce a program that would:

- Read tracedata (from different kinds of sources)
- Rewrite it into a better format (for easier parameter data extraction)
- Analyse the data and plot graphs with data from the analysis

Since we didn’t have any advanced criteria on what our program should do we felt that it was best to build it upon an already existing project or in an

environment that already had the functionality to do all we needed.
We started to analyse available (already existing) tools by asking personel at Ericsson if they had any preferences or any tools that they knew that they already had licences for or tools that was already used in the lab testers environment. Unfortunately they didn't have any recommendations, so we started to search the web for good tools that would be able to produce graphs and analyse big sets of data. Our initial idea was Matlab since both of us have experience with Matlab and it contains extensive libraries for calculating signal data and for plotting graphs. Some of the other tools we looked at was [TODO add list of tools]. But mostly all programs seemed to do pretty much what Matlab did or less, and since we had experience with Matlab since earlier we felt that Matlab was our best choice.
We were still pretty open minded about using other softwares at this point, but started out with just writing code for displaying graphs. After about a week we found out from our supervisor that Ericsson did indeed already have a Ericsson-developed program called Logtool that the lab testers used for handling tracedata. It was written in java and the program was built upon the eclipse framework, all of which Razmus already had experience with since earlier. The team in charge of the project was in LinkÃ¶ping, and using Logtool would enable us to get tracedata in a good format without us having to write a single line of code to filter the unfiltered logfiles, plus our program wouldn't need any form of extra integration for Ericsson.
We therefore decided to develop our program as a plugin to Logtool.

3.2 Logtool Plugin

Logtool is a program that contains different analyzers that does individual analysis on raw trace data. Each analyser that the project uses is added as a plugin-project and not directely integrated into the project. We needed the raw trace data in a BB-filtered(BaseBand-filtered) format, and it existed a plugin which already did that. We could have written our project as a new plugin but both we and Ericsson were agreed on that it would be unnecessary to implement a new plugin which would need to BB-filter the data when that functionality already existed.

3.3 Description of the Tool

Ericsson first requested us to do a lightweight analyzer that would plot graphs in a basic interface. We included extra functionality and created a total of 3 different kinds of views (a separate window in eclipse) with unique functionality.

3.3.1 Basic Graph View

In the first view we plot graphs with specific values over SINR and CQI (CQI is for downlink and SINR is for uplink). These values we got as suggestions from Rickard Wahlgren and Jonas Wiorek two Ericsson employees at Kista that has a good insight in layer 1 and also came up with this thesis idea. The reason for why we look at SINR in uplink and CQI downlink is because the UE doesn't calculate SINR, it calculates CQI which can be translated as a SINR but it only contains 15 values. The graphs you can look at is Throughput/SINR, Throughput/CQI, PRB/CQI PRB/SINR BLER/CQI, BLER/SINR, SINR/(UL MCS) CQI/(DL MCS) ... These graph are presented in a two dimensional graph as showed in picture X. Where the data is on the Y-axis (vertical axis) and SINR / CQI / MCS is on the X-axis (horizontal axis).

Image X (Basic Graphs View Image)

3.3.2 Multiple Graph View

The second view contains the option to save and load graphs. If you have a set of data you should be able to compare them. There is a save button in the basic graph view. After you have saved your graphs you can run several other bb-filtrations. The user can then load all the saved datasets that you wish to compare. Each loaded dataset can be toggled on/off though a list in the right corner of the view.

Image X (Multiple Graphs View Image)

3.3.3 Advanced Graph View

The third view is a more advanced graph tab. In this view you are able to look at data in any form of graph where you choose what you want to look at in the X and Y-axis, and then plot your graph over those values. The user can also save the BB-filtered data to file and load it later.

Image X (Advanced Graphs View Image)

Chapter 4

The Analysis of BLER target

In this chapter the analysis of the BLER target is formulated, the reason behind the anlysis, how we did the analysis and a summary.

4.1 The reason for the analysis

The reason why we did this analysis is because we wanted to show that an analysis of something on lower layers are able to do and to verify that our tool works on a set of data. Since they didn’t have anything they wanted to analyze we came up with our own idea of a relevant analysis. This was to analyse the optimality of the BLER target at the enodeB.

4.2 explanation the analysis

In the EnodeB there is a BLER target for the data that is sent in downlink and uplink. This means that the data that is sending shall have an error rate at a specific level. This is called BLER target or target BLER [source]. This way you can change your modulation and coding scheme (mcs) until the BLER target is achieved. The definition for BLER is as explained earlier number of block NACKs / number of block ACKs (over the 100 previous blocks [source]). So there is a tradeoff between different BLER-targets. The higher BLER target you have the higher number of bits can be sent, but you will have more retransmissions. The lower BLER target you have the lower block sizes and lower retransmission.
In the analysis we will look at this BLER target rate, Which in ericssons eNdoeB’s are at 10%. Is this the optimal target rate if you want to achive the highest throughput? what is the throughput at 11% or 9%? to analyse this we do several Basebands traces with exact similar conditions. To do

this we used a script that an employee in Ericsson had done before, were we switch the SINR with the same time inteval. So the first traces we collected was on BLER 10% (which is default) were we switched the SINR from 20 dB to -20 dB, were each SINR is help in four seconds. The reason why this script is used is to both to achieve as similar environment but mostly to be able to compare the data over time.

4.3 How we collected data

What we will look at what the throughput is over time and the throughput over SINR, to see which the best BLER target. Are there any difference between the throughput and the SINR level with the different BLER target? So we first drew a trace with default settings, that is a BLER target at 10%. Then we drew traces with BLER target at 1%, 5%, 7%, 9%, 11%, 13%, 20%, 35%. The traces we drew we started from an SINR from 20 dB to -20 dB, were each SINR step is four seconds, we drew this with a script that changed SINR level every four second. The reason why this script is used is to both to achieve the same environment and to be able to compare the data over time which would be impossible to do if we didn’t have the same simulation time.

4.3.1 expected result

Thoughput over time

What we expected as result of the throughput is that the optimal BLER target in a throughput sense is 10% simply because it is used by the eNodeB’s, the curves close to it ie 9% and 11% should be close to that curve, the other one should worsen the further you come from 10% also the MCS values should should decrease after a little a while in time. These values should decrease linearly but with a delay. The delay of when the MCS changes should be longer the larger BLER target you have.

Throughput over SINR

It is not specified anywhere that blerTarget in a eNodeB should be 10%, but it is the most common value that’s used in eNodeBÂ´s so we made the assumption that, for a UE which ran the same simulation for different blerTargets then 10% would have the best thoughput.
We expected the result to be that the 10% curve would be the best over all the SINR. The other ones should be slightly worse or much worse

4.3.2 Simulated results

When we simulated the result we that the BLER target at 10% wasn’t in fact the best one. We saw that the 5% curve was better i.e. gave a higher throughput than the other ones, and that it was better at higher SINR value and slightly worse in bad channel conditions

4.3.3 Summary

With this analysis we have first of all shown that we are able to analyse data. We are able to analyse different trace data in the same window so that we can compare different log file with each other. We can compare different data over different parameters
We can also see that from this analysis that the optimal BLER target isn't fixed, it is varying. We can also see fast that the data is wierd in the beginning of the trace for some of the traces. We don't know why.

4.4 Testing of our tool

To validate that our work is useful right now and not only a potential future work of analysis we felt that we had to do a test with their previus method and compare it with what we can do with our tool. There is right now quite few test that need of a graphical representation is quite redundant. The most relevant we found is the tests of link adaptation.

4.4.1 selection of measurements

What we wanted to do is the best way we could measure how good our tool was. Since the way the testers work is that they manually look at the log file. So our comparison is with that. What we look at is
how much more time efficient would our tool be. Can we detect errors or faulty behavior with our tool (if there are any) how more accurate is our tool to the looking log files.

4.4.2 measurements result

what we could show with our tool is that it is much faster than to manually. To see that our tool is much better and faster and it works we think.

Chapter 5

Result

Here we explain the result of the analysis and that we made some testers do a test with our tool and answer a survey regarding the usability of the tool...

Chapter 6

Conclusion

What we have done in this master thesis is to develop an analysis tool for Ericsson. The purpose was to be able do in-depth analysis on traffic data in in the physical layer, which they were not able to do before. What we developed was a graphical tool that took in two sets of data that comes from a trace file and plot it as a graph. This way one can make more in-depth analysis of physical layer data, especially those of HARQ and LA. To show that an analysis is possible we did our own analysis on the BLER target which is closely linked to the Link adaptation algorithm [source]. From the result of this tool we can say that we can do ...

6.1 Things we didn't had time to do...

Here we desbribe task that we wanted to do but didn't have time to do.

6.2 Future work

In this area we have experienced that there are not a lot of similar tools. We had many ideas how to further develop this tool to something more useful tool. The drawback right now with the tool is that in able to run it you have to draw a trace save it to a raw, file and then read the file which may over all be quite time consuming. If you want to do a quick analysis or a faster one. you should be able to read the data in real time which logtool has support for and from that data produce graphs. This we felt were out of our scope for this thesis.

We also feel that it is necessary to adjust the test cases to suit our tool better. Right now to do a test is quite abstract, and you don't look in-depth so you don't get a good picture of what the data look like.

There is also a number of features we felt that we could implement.

One is that for each value on the X-axis, how many values were the average calculated from. One is also that there are only a few parameters that are linked to each other. You could then delete the number of irrelvant data on the other axis when when you have chosen a parameter on the first. One could also do more representation of data in the bbfilter columns. Right now we only take care of a few number of non-numbers parameters. One could do more special representations of these if you want to plot over that. There is also relevant to do something about the files that bbfilter translates. They are sometimes erroneous. You could do implement some functionality that corrects it.

References

[1] Teppei Ebihara, Hidekazu Taoka, Nobuhiko Miki, and Mamoru Sawahashi. Performance of Outer-Loop control for AMC based on mutual information in MIMO-ODFM downlink, 2012
[2] Tao Cui, Feng Lu, Anil Goteti, V. Sethuraman, S. P. Rao and P. Subrahmanya, Throughput Optimization in High Speed DownlinkPacket Access (HSDPA), 2009

Linköping University

Avdelning, Institution
Division, Department

DIVISIONSHORT,
Dept. of Computer and Information Science
581 83 Linköping

Datum
Date

December 19, 2014

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ ____________

Rapporttyp
Report category

☐ Licentiatavhandling
☒ Examensarbete
☐ C-uppsats
☐ D-uppsats
☐ Övrig rapport
☐ ____________

ISBN
ISBN

ISRN
LiU-Tek-Lic–2015:THESISYEARNUMBER

Serietitel och serienummer **ISSN**
Title of series, numbering -

Linköping Studies in Science and Technology

Thesis No. THESISNUMBER

URL för elektronisk version

http://XXX

Titel
Title

TITLEBIBPAGE

Författare
Author

Paul Nedstrand & Razmus Lindgren

Sammanfattning
Abstract

§ This master thesis involves developing a lightweight analyser that produce statistics from the traffic data in the communication link between a UE e.g. a cell phone and a E-UTRAN base station (the air interface). The analyser tool will produce graphs with information about the correlation between two signal data in the channel (e.g throughput over interference). From the statistics produced by the analysis tool, the testing personell at Ericsson can more easily detect potential faulty behavior from the UE or eNodeB in a much faster way than they were able to do before. This tool will also be able to help ericsson to rewrite test cases to be more indepth than they are now. To show that that it is possible to some kind of analysis with this tool, we made our own analysis on the link adaptation target BLER, this way we could validate that our tool could handle large amount of data and also that you can compare different data with each other. To be able to show that our tool is usable for Ericsson IODT we offered IODT test engineers to do tests on Link adaptation and HARQ with our tool and evaluate it with their current methods.

Nyckelord
Keywords KEYWORDS