



A study on the Optimal Placement of
the Decision-Making Entity in LTE
Mobile Networks

Master Thesis

Postgraduate programme “Technology Education & Digital Systems” – Digital
Communications and networks

Aristotelis Margaris, University Of Piraeus 2013-2014

Supervisor: Prof. Panagiotis Demestichas

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

*Dedicated to my girlfriend,
my parents
and my closest friends*

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Table of Contents

Table of Tables	8
Table of Figures	10
Acronym Analysis	12
Chapter Analysis.....	14
Περίληψη	16
Abstract.....	18
Chapter 1: Introduction	20
Chapter 2: Overview of the LTE technology.....	22
2.1 Performance Analysis of the LTE System	23
2.2 LTE Technological Features	24
2.2.1 The OFDMA multicarrier scheme	24
2.2.2 The Usage of Multiple Antenna's (MIMO).....	26
2.2.3 The implementation of Packet-Switched Radio Interface	27
2.3 The LTE System Architecture.....	28
2.3.1 Block Architecture Analysis of the EPS	28
2.3.2 Analysis of signaling procedures	31
2.4 LTE Policy enforcement and the ANDSF template.....	34
2.4.1 Implementation of the ANDSF policy exchange system.....	34
2.4.2 Our implementation of the ANDSF policy template	36
2.5 LTE and Heterogeneous Networks	37
Chapter 3: Intelligence and Decision Making in the LTE	40
Chapter 4: Development of the Simulation Tool.....	44
4.1 State of the Art and Beyond SotA for LTE Simulation	44
4.2 Requirements Analysis.....	48
4.3 System Design.....	48
4.3.1 Simulation Engine.....	48

4.3.2	System Modules and their relations	51
4.4	Analysis of Interfaces and Abstract Classes	53
4.4.1	Abstract Class Event	54
4.4.2	Abstract class Measurement	57
4.4.3	Abstract class Application	60
4.5	Analysis of Data Structures	61
4.5.1	The Mobility Stats Data Structure	61
4.5.2	The Topology Data structure	62
4.5.3	The LTE Information Data Bundle Structure	63
4.5.4	The Energy Module data structure	64
4.5.5	The ANDSF module data structure	64
4.6	Analysis of Graphical User Interface	66
4.6.1	The Initialization window	67
4.6.2	Analysis of the Playground graphical elements	68
4.6.3	Analysis of the Output graphical elements	71
4.6.4	Analysis of the various system dialogs	73
4.7	Analysis of the Multi-Threaded Environment	77
4.8	Flow Diagram of the Program	79
4.9	External API Analysis	79
4.9.1	The JFreeChart API	80
4.9.2	The Apache POI library	81
4.9.3	The DOM xml parser library	81
4.9.4	Other External Classes	82
4.10	Analysis of the Simulation Input and Output	83
4.10.1	Input Parameters	83
4.10.2	Output	92
4.10.3	Constants	94

Chapter 5: Simulation scenarios and Results.....	98
5.1 General Assumptions of the Simulation.....	98
5.2 Test Cases used for this study	101
5.2.1 Comparison between different sizes of networks	101
5.2.2 Comparison between different deployments of Wi-Fi Aps	103
5.2.3 Comparison between different decision making intervals.....	104
5.2.4 Comparison between Network-initiated Decision-making and User Equipment-initiated Decision-making.....	107
Chapter 6: Recommendations and Conclusions	110
6.1 Recommendations regarding the Mobile Network Operators.....	110
6.2 Recommendations regarding the User Experience	114
6.3 Recommendations regarding the LTE equipment vendors	115
References.....	118
Appendix I – Genetic Algorithm Implementation	120
Problem Model.....	120
Genetic Algorithm.....	120
Results	122
Appendix II – Simulator’s source code	124

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Table of Tables

Table 1 - Timing Diagram of Graphics.....	66
Table 2 - Table of the simulator's Network input parameters.....	84
Table 3 - Table of the Simulator's signalling input parameters	86
Table 4 - Table of the simulator's traffic/decision-making parameters	87
Table 5 - Table of the Simulator's input parameters for the simulation	88
Table 6 - Table of the Simulator's ANDSF input parameters	90
Table 7 - Table of Constants in the Simulator	94
Table 8 - Network Input general assumptions	98
Table 9 - Transmission input general assumptions.....	99
Table 10 - Signalling Input general assumptions.....	99
Table 11 - Network traffic general assumptions.....	100
Table 12 - Decision-Making general assumptions.....	100
Table 13 - ANDSF general assumptions	100
Table 14 - Objective Function general assumptions.....	101
Table 15 - Miscellaneous general assumptions	101
Table 16 - Input parameters, scenario 1	102
Table 17 - Input parameters, scenario 2	103
Table 18 - Input parameters, scenario 3.....	104
Table 19 - Input parameters , scenario 4.....	108
Table 20 - Mobility correlation with RAT's antenna coverage.....	111
Table 21 - Failure percentage of handover procedure for scenario 1	112
Table 22 - Failure Percentage of handover procedure for scenario 2	112
Table 23 - Comparison between the two scenarios	113
Table 24 - Algorithm Input.....	122

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Table of Figures

Figure 1- System requirement comparison between UMTS and LTE[12].....	23
Figure 2 - (a) The OFDMA and SC-FDMA schemes, (b) the OFDM Sub-Carriers[12]	25
Figure 3 - Diversity Gain, Array Gain, Spatial Multiplexing[12]	27
Figure 4 - The block architecture of EPS[12].....	29
Figure 5 - LTE attach procedure	32
Figure 6 - Data Call setup request.....	32
Figure 7 - Data Call Release request.....	33
Figure 8 - Intra-LTE handover procedure.....	33
Figure 9 - Example of an ANDSF template[9].....	35
Figure 10 - Extended example of the ANDSF policy template[9]	35
Figure 11 - Our implementation of the ANDSF template	37
Figure 12 - The Decision making sequence diagram.....	43
Figure 13 - (a) Before the Graphical modification and (b) after the Graphical modification	47
Figure 14 - Basic Operation of the Event-Queue.....	50
Figure 15 - Abstract Module Diagram.....	53
Figure 16 - The Application's state machine	60
Figure 17 - Analysis of the Mobility Stats Data structure	62
Figure 18 - The Data Bundle's role in the two decision-making cases	64
Figure 19 - Analysis of the initialization GUI	68
Figure 20 - Analysis of the main graphical view	71
Figure 21 - The graphical elements of simulation output	73
Figure 22 - Dialog for the selection of the imported topology file.....	74
Figure 23 - The runtime simulator settings dialog.....	75
Figure 24 - The JFreeChart additional options dialog	76
Figure 25 - The signaling procedure customization dialog	77
Figure 26 - The thread diagram	79
Figure 27 - Flow Diagram of the Software.....	79
Figure 28 - Example of a JFreeChart plot.....	80
Figure 29 - Example of the Apache POI API-generated Excel file.....	81
Figure 30 - Example of the generated xml file from the DOM library.....	82

Figure 31 - Example of the Gaussian Blur transformation83

Figure 32 UE and Network-initiated decision making latency..... 102

Figure 33 UE and Network-initiated decision making latency..... 103

Figure 34 Considered topology..... 105

Figure 35 – Average handover count for each criteria (decision-making interval in seconds)..... 105

Figure 36 – Cumulative signalling bytes per scenario (decision-making interval in seconds)..... 106

Figure 37 – Average Channel Quality Indicator per scenario (decision-making interval in seconds) 107

Figure 38 – Average Downlink throughput per scenario (decision-making interval in seconds)..... 107

Figure 39 - Average Handoff count for the UE and Network initiated Decision Making 108

Figure 40 – (a) Average UE Downlink Throughput (b) Average Cell Load for the Network and the UE initiated decision making 109

Figure 41 – (a) Average cumulative signalling Bytes (b) Average eNodeB back-end Throughput for the Network and the UE initiated decision making..... 109

Figure 42 - Genetic Algorithm Results 123

Acronym Analysis

LTE: Long Term Evolution for the 3GPP technologies

OFDMA: Orthogonal Frequency Division Multiple Access

MIMO: Multiple Input Multiple Output

UMTS: Universal Mobile Terrestrial System

HSDPA: High Speed Downlink Packet Access

RRH: Remote Radio Head

GSM: Global System for Mobile communications

RLC: Radio Link Control

PDCP: Packet Data Convergence Protocol

RRC: Radio Resource Control

3GPP: 3rd Generation Partnership Project

IEEE: Institute of Electrical and Electronics Engineers

WiFi: Wireless Fidelity

ANDSF: Access Network Discovery and Selection Function

MIH: Media Independent Handover

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Chapter Analysis

The layout of this Master Thesis is as follows:

- **Master thesis abstract in the writer's native language(Greek)**
- **Master thesis abstract in the English language**
- **Chapter one is the introduction of the document to the scientific field it will analyze.**
- **Chapter two contains general information about the LTE technology with a layout that relates to this research.**
- **Chapter three is dedicated in the analysis and formulation of the optimization for the decision making functionality of the handover procedure.**
- **Chapter four is the analysis of State of the Art simulation platforms, and a detailed overview of the simulation tool we have developed.**
- **Chapter five is dedicated in different scenarios and results generated by the created software.**
- **Chapter six is the conclusion of this master thesis as it derives from the simulation data analysis we have conducted.**
- **Chapter seven is the references section.**
- **Chapter eight is the appendix section**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Περίληψη

Οι τεχνολογικές εξελίξεις στα προηγμένα δίκτυα κινητής τηλεφωνίας αλλά και οι αυξημένες λειτουργίες που παρέχουν στους χρήστες τα smartphones, οδηγούν σε υψηλότερες απαιτήσεις στην απόδοση των αλγορίθμων και των διαδικασιών που εκτελούνται σε αυτά. Μία πολύ σημαντική παράμετρος των δικτύων είναι η τοποθέτηση της λογικής οντότητας που εκτελεί τις ευφυείς αποφάσεις ενός 4^{ης} γενιάς ετερογενούς δικτύου κινητής τηλεφωνίας. Η τοποθέτηση αυτή μπορεί να βρίσκεται στο κινητό τερματικό, ακριβώς στην πηγή των μεταβολών της τηλεπικοινωνιακής ζήτησης του χρήστη, ή στο δίκτυο εξυπηρέτησης και σε κάποια οντότητα του όπως το eNodeB. Μία πολύ σημαντική απόφαση που πρέπει να ληφθεί για αυτά τα δίκτυα είναι η απόφαση της μεταπομπής της ενεργής κλήσης ενός τερματικού καθώς αυτό διασχίζει το σύνθετο δίκτυο του παρόχου το οποίο μπορεί να απαρτίζεται από Macro-cells, pico-cells, WiFi Access Points ιδιωτικής χρήσης αλλά και άλλες τεχνολογίες πρόσβασης. Για την εκτίμηση της βέλτιστης επιλογής έχουμε αναπτύξει μία συνάρτηση βελτιστοποίησης η οποία χρησιμοποιεί δεδομένα από το κινητό τερματικό, από τους καταχωρητές του δικτύου του παρόχου αλλά και από το επιλεγμένο πρότυπο λειτουργίας του δικτύου, το πρότυπο ANDSF, και κρίνει όλους τους πιθανούς στόχους της μεταπομπής για να βρει τον βέλτιστο όπου και θα εκτελέσει την μεταφορά της εξυπηρέτησης του. Για τον έλεγχο των δύο διαφορετικών τοποθετήσεων της λήψης αποφάσεων, ένα ειδικό προγραμματιστικό εργαλείο αναπτύχθηκε με πάρα πολλές δυνατότητες παραγωγής διαφορετικών σεναρίων, προβολής της προσομοίωσης με γραφικό και εύχρηστο περιβάλλον αλλά και έναν ισχυρό κεντρικό μηχανισμό μικρο-προσομοίωσης βασισμένο σε μία ντετερμινιστική ουρά γεγονότων. Τα δεδομένα που εξήχθησαν από προσομοιώσεις σεναρίων-κλειδιών για το επιλεγμένο πρόβλημα χρησιμοποιήθηκαν και συσχετίστηκαν με τα μεγέθη επίδοσης που αφορούν διαφορετικές οντότητες της διαδικασίας (χρήστες, παρόχους ή κατασκευαστές εξοπλισμού) και ειδική επιχειρηματολογία έχει συνταχθεί για την υποστήριξη των προτάσεων. Τέλος ακολουθεί παράρτημα με το λογισμικό που αναπτύχθηκε και τις προγραμματιστικές αρχές που ακολουθήθηκαν.

Λέξεις κλειδιά: Πρότυπο κινητής τηλεφωνίας LTE, Ευφυία, Λήψη Αποφάσεων, Προσομοίωση, Βελτιστοποίηση, ANDSF πρότυπο, Ετερογενή κυψελωτά δίκτυα

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Abstract

The technological improvements of the advanced cellular radio networks and the ever-increasing functionalities provided to the end users by the smartphones, lead to more strict requirements in the performance of algorithms and procedures that are executed by them. One very important parameter of these networks is the placement of the logical entity that executes the intelligent functionalities of a 4th Generation heterogeneous cellular radio network. The placement of the intelligence can either be in the LTE User Equipment, right next to the source of the change in the traffic demands of the user, or it can be placed in the underlying network (i.e. the eNodeB) with access to contextual information about the network composition and load. A very important decision that need to be taken in these networks is the handover decision of a connected UE's active data call as it crosses the complex network of the provider. This network consist of Macro-cells, pico-cells, WiFi Access Points owned by the provider and other Radio Access Technologies. To evaluate the optimal selection of the wireless node, a fitness function has been developed that uses as input data from the mobile terminal, from the registries of the provider's network and also from the ANDSF template, an xml-based file that contains the provider's preferences for the decision-making functionalities. Based on the results of this fitness function, all the possible handover targets are compared in order to discover the optimal and perform the handover procedure to change its serving node. For the comparison of the two different decision-making paradigms, a special simulation software has been developed with the capabilities of entering a plethora of input variables to form multiple sets of simulation scenarios, projection of the simulation during runtime with detailed and potent graphical elements and a powerful central processing core that is based on the architecture of micro simulation based on deterministic Event Queue. The extracted output data from the simulation of key-scenarios for the selected research is then utilized and correlated with the respective attributes that concern different stakeholders (End Users, Network Providers, Technology vendors) in order to synthesize an adequate list of arguments to proceed to recommendations. After that a detailed appendix section follows that contains the summary of the developed source code and the development principles that were used for its implementation.

Keywords: LTE 3GPP technology, Intelligence, Decision-Making mechanisms, Heterogeneous cellular networks, Optimization, Simulation, ANDSF template

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Chapter 1: Introduction

Cellular networks [12] [13] are one of the most important wireless technologies of our time. They started as simple platforms for voice-oriented services and progressed to high end devices with fast internet connection and multimedia capabilities in almost a decade. The amount of people that these days use cellular networks is so high that a discrete market share is reserved only for the mobile technologies corporations. As technologies evolve, so do the expectations of the associated parties in this scientific field. This is why the cellular technologies have evolved from the 2nd Generation (GSM) to the 3rd Generation (UMTS) followed by HSDPA (3.5 generation) to the most recent technologies of LTE release 8 (3.75 Generation) and following 2 more releases (release 9 and 10). Each addition to the 3GPP standard comes with great improvements but also higher complexity, higher cost of implementation and broader opportunities for research. From the simple single-cell-type cellular network of GSM Base Stations we have evolved to the fully heterogeneous environment of multiple coexisting radio technologies (legacy and recent) and also for each technology, cells of different transmission parameters, different coverage zones and traffic serving purposes. Macro-cells, Micro-cells, Pico-cells, Femto - cells, remote radio relays and simultaneously other technologies like the IEEE 802.11(x) family all exist to serve the ever increasing demand for fast internet access and high quality communication with remote applications. Cellular network operators have begun to use the Wi-Fi technology as an off-loading mechanism for their cellular networks which creates the need for the advance in 3GPP/IEEE technology communication and interoperability frameworks. In this complex environment, a decision must be taken for each active user equipment terminal in order to better utilize the underlying network in conjunction to its current context. The context of each cell phone are all the parameters that influence the communication with the active wireless network and need to be gathered to the virtual entity that conducts this decision. In the 3rd Generation decision-making implementations, we see that this decision is conducted by the network in a rather centralized manner but with changes brought about with the LTE technology, the intelligence of the networks have come closer to the access network. The re-invention of the Base station node, brought by the 4G family of the 3GPP technologies is known as the eNodeB and presents high intelligence and efficiency at the edge of the mobile

network, something highly needed for the reduction of the delay of call initiation, one of the greatest improvements promised by the new technologies. But with the transference of part of the decision-making functions to lower (hierarchically) network nodes, some may wonder if it will ever be viable to transfer this functionality to the user terminals, thus reducing the decision-making functionality delay to its theoretical minimum. In this research, we are studying the effects of placing the center node of this intelligent decision functionality in either the user equipment or the LTE eNodeB by means of analytics and simulation. In order to determine the optimal placement of this functionality, all the network stakeholders with their respective criteria need to be considered. These may be the cellular technology vendors, the network operators and also the end users, each having a different and sometimes conflicted set of evaluating angles. We will also analyze certain aspects of developing a simulator for a heterogeneous cellular network and describe its strengths and weaknesses. Output data for various simulation scenarios will be evaluated to conclude to a detailed recommendation for the placement of the decision making functionality for the 4G cellular technologies of the future.

Chapter 2: Overview of the LTE technology

The latest stage of the 3GPP technologies[12][13] evolution is the LTE family, a State of the Art cellular technology system with potential of becoming the sole mean of mobile internet access of the 21st century. It is a complex but orchestrated multi-entity system that gives its users access in Broadband internet and various cellular services (voice, texting, other content) inside its seemingly endless coverage area. It is coexisting with other radio access technologies by the usage of orthogonal frequency bands and therefore it is the strongest addition to the available radio technologies of all urban, sub-urban and rural territories. LTE was created to fulfill the following set of requirements as they derive from the operation of its predecessor the UMTS mobile system:

- The reduction of call initiation delay and the transmission time needed for both signaling and data transferring.
- Increasing data throughput (Downlink and Uplink) by a factor of 10 in relation to the UMTS.
- Improving the behavior of the cellular system in the cell-edge environment for a fairer mobile service distribution.
- Reduced operational cost per Hz for the transition to wider frequency bands efficiently.
- Volatile radio transceiver with capabilities of fast frequency changes for greater spectral flexibility.
- All-packet network architecture that simplifies the cellular system End-to-End.
- Seamless mobility between LTE eNodeBs and between LTE and Legacy or other radio technologies.
- Power consumption on both the LTE eNodeB and the LTE user equipment that allows it to be a functional mobile system.

By fulfilling these requirements the LTE cellular system has become a State of the Art wireless system with very high Quality of Service capabilities that can be used to support all types of modern applications. The evolution of user equipment devices to smartphones means that voice and text messaging services are becoming more and more obsolete. Their place is taken by VoIP applications, web browsing, file

downloading and video streaming which require more advanced QoS standards than their predecessors. That applies pressure to the new network technologies in order to generate the wanted demand and growth to be deemed profitable and successful.

2.1 Performance Analysis of the LTE System

The LTE technology[12][13] is created to surpass its cellular network previous system the UMTS. Therefore the comparison between the two technologies is used to show the latitude of the overall improvement in mobile telecommunications. Performance in networks is usually measured by the scalar of maximum throughput capability, the minimum average throughput per user, the spectral efficiency value (bits /second/ Hz) and also the call initiation delay, the jitter caused by the infrastructure routing etc. The following table will compare the UMTS technology and the LTE technology system requirements in the pre-described variables.

		Absolute requirement	Release 6 (for comparison)	Comments
Downlink	Peak transmission rate	> 100 Mbps	14.4 Mbps	LTE in 20 MHz FDD, 2 × 2 spatial multiplexing. Reference: HSDPA in 5 MHz FDD, single antenna transmission
	Peak spectral efficiency	> 5 bps/Hz	3 bps/Hz	
	Average cell spectral efficiency	> 1.6–2.1 bps/Hz/cell	0.53 bps/Hz/cell	LTE: 2 × 2 spatial multiplexing, Interference Rejection Combining (IRC) receiver [3]. Reference: HSDPA, Rake receiver [4], 2 receive antennas
	Cell edge spectral efficiency	> 0.04–0.06 bps/Hz/user	0.02 bps/Hz/user	As above, 10 users assumed per cell
	Broadcast spectral efficiency	> 1 bps/Hz	N/A	Dedicated carrier for broadcast mode
Uplink	Peak transmission rate	> 50 Mbps	11 Mbps	LTE in 20 MHz FDD, single antenna transmission. Reference: HSUPA in 5 MHz FDD, single antenna transmission
	Peak spectral efficiency	> 2.5 bps/Hz	2 bps/Hz	
	Average cell spectral efficiency	> 0.66–1.0 bps/Hz/cell	0.33 bps/Hz/cell	LTE: single antenna transmission, IRC receiver [3]. Reference: HSUPA, Rake receiver [4], 2 receive antennas
	Cell edge spectral efficiency	> 0.02–0.03 bps/Hz/user	0.01 bps/Hz/user	As above, 10 users assumed per cell
System	User plane latency (two way radio delay)	< 10 ms		LTE target approximately one fifth of Reference.
	Connection set-up latency	< 100 ms		Idle state → active state
	Operating bandwidth	1.4–20 MHz	5 MHz	(initial requirement started at 1.25 MHz)
	VoIP capacity	NGMN preferred target expressed in [2] is > 60 sessions/MHz/cell		

Figure 1- System requirement comparison between UMTS and LTE[12]

As we can see the LTE technology Downlink throughput peak transmission rate is 100Mbps, higher by a factor of ~10 than the 14.4 maximum throughput opted out for

the UMTS system. Translating it to their respective band width, a 5 bps/Hz spectral efficiency is achieved in the LTE technology as opposed to 3 bps/HZ in the 3rd generation predecessor. Metrics about the quality of the service in the cell edge are also visible, including user plane latency and call initiation latency for each technology.

The described data rates are peak data rates that require the use of perfect transmission environment for MIMO 2x2 spatial multiplexing and 20 MHz of bandwidth. The 100Mbps throughput is divided symmetrically and requires a single LTE terminal in connected state in order to acquire it. These assumptions are crucial for this performance criteria but they also vary from real time deployments of such systems.

2.2 LTE Technological Features

In this chapter we will see a set of technological upgrades that the LTE standard has implemented across all the network layers of the system and their contribution in achieving the specified data rates and QoS parameters.

2.2.1 The OFDMA multicarrier scheme

Cellular networks[12][13] are serving a very large amount of user clients within huge geographic areas. This causes the mobile technologies to use Multiple Access techniques that ensure the orthogonality in the transmission of two nearby users and also the relative orthogonality between different eNodeBs. For the LTE, two multi carrier schemes were presented, the OFDMA (Orthogonal Frequency Division Multiple Access) and the Multiple WCDMA (Wide-band Code division Multiple Access) both using a very wide bandwidth and with different pros and cons. The selected mode was OFDMA for Downlink and SC-FDMA (Single Carrier, Frequency Division Multiple Access) was selected for the Uplink. The selected two techniques give the system great opportunity for utilizing the frequency domain as a resource pool of specially formed harmonic pulses called sub-carriers. Each subcarrier is orthogonal to its respective neighbors therefore the spectrum is simultaneously occupied by different logical information streams. The sub-carrier entity has also great behavior in bad radio environments with inter symbol interference and bad signal quality. Below we see an image showing the two selected schemes (Uplink and Downlink).

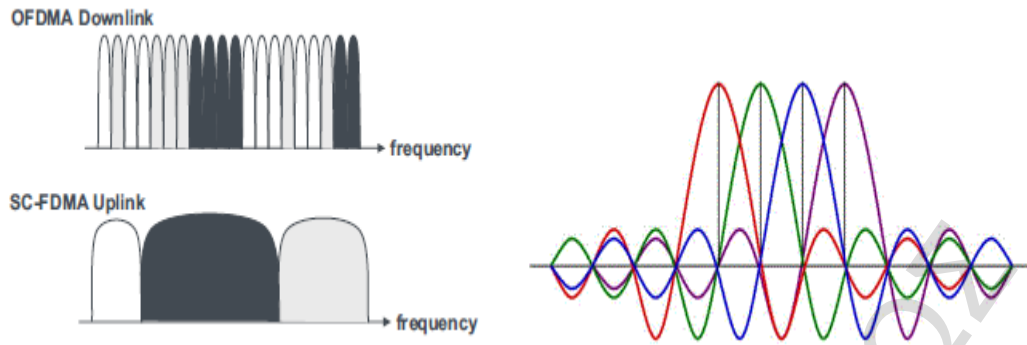


Figure 2 - (a) The OFDMA and SC-FDMA schemes, (b) the OFDM Sub-Carriers[12]

Each subcarrier may be overlapping in a small portion of its neighbors but they do not interact in a distorting way for the underlying complex symbol. If OFDMA is broken down to its key parameters they can be summarized as follows:

- The antenna, RF-Base Band filters and physical components of an LTE eNodeB need no adjustments or redesign in order to increase or decrease the spectrum bandwidth.
- The resulting transmission elements (or resources) can be distributed to different users for a highly orthogonal and un-correlated symbol transmission. Chunks of subcarriers and time slots are named resource blocks (because it has 2 dimensions) and the resource allocation schemes use this logical entity to perform the scheduling of each user.
- Fractional frequency reuse and ICIC techniques (Interference Cell interference Coordination/Cancellation) can be implemented for a global increase in spectral efficiency.

Also key elements of the OFDM modulation apply with this multiple access scheme (as they derive from various implementations in other radio technologies) increasing its credits. They are presented as follows:

- Robustness against radio environments with many reflections, resulting in delayed copies of the signal being received by the target antenna. This happens due to the spectral form of the subcarrier, especially because the greater part of the transmission power is absorbed by the central lobe of the subcarrier which consumes a small sub band of the subcarrier bandwidth.
- Low complexity receivers (system block parts) due to the usage of equalizing in the frequency domain.

- Flexibility in combining different radio symbols from different transmitter to generate broadcasts.

These elements while important for the Downlink of the LTE system, they cannot be used for the Uplink scheme as well because the User Equipment has limited power resource and therefore cannot accommodate for the high PAPR (Peak to Average Power Ratio) it demands.

2.2.2 The Usage of Multiple Antenna's (MIMO)

One of the great revolutions in radio telecommunications[12][13] is the opening of the space domain, which increased the dimension and therefore the logical capacity of the radio telecommunications. While satellite telecommunications have great difficulty in exploiting that, the cellular telecommunications and the LTE standard feature the MIMO capability as a way to greatly improve the quality of service in the suitable environments. With small drawbacks, the creation of multiple spatial beams give an almost scalar increase in the average throughput of the radio channel helping the LTE radio interface even exceed the system requirements as shown in previous chapters. The benefits of the multiple antenna scheme can be summarized as follows:

- Diversity gain, is the profit that comes with using different paths of the same radio transmission environment to overcome the opportunistic shadowing and random behavior that it may produce. By using diversity, a smart radio transmitter will change between two different spatial links in order to always transmit on a channel power peak.
- Array gain, is the profit that comes with using multiple antennas to broadcast the same, or time shifted versions of the same data, taking into consideration the additive effect that this will have on the receiver. The gain that comes from array gain is almost every time a gain in SINR (dB) and helps reach terminals with very bad radio circumstances.
- Spatial Multiplexing is the profit that comes with using all the possible spatial "routes" to transmit different information in order to use the same spectrum for more than one logical channels. Spatial multiplexing is the hardest but most efficient usage of MIMO scheme and many modes have been proposed for its operation.

Below we can see the three different methods of utilizing the MIMO schemes in their respective context:



Figure 3 - Diversity Gain, Array Gain, Spatial Multiplexing[12]

2.2.3 The implementation of Packet-Switched Radio Interface

The convergence between the computer networks world and the telecommunications world happens with the LTE technology specification[12][13] of an all packet radio network. Circuit-switching and other connection oriented protocols are being reduced to minimum to allow for the full quantization of the information traffic into discrete, self-contained entities the packets. This has been a motion started from before the LTE technology. The HSPA release for the UMTS generation of cellular telecommunications allowed for a small packetization of information with size close to the coherence time of the channel. This was a cross-layer approach (Layer 1, Layer 2) in the information fragmentation procedure that essentially opened the road for further implementation of an all packet architecture. Packets with small time duration are being introduced and combined with the new dimension of subcarriers (frequency) and space (streams) led to the Link Layer of the LTE and the LTE-A. The key features of this fusion are:

- Intelligent scheduler schemes that use the physical layer feedback for greater efficiency and usage of the air medium and its properties.
- Real time shifting between different MIMO configurations depending on the channel condition and capabilities.
- Adaptive coding and modulation scheme for different variations of the transmission conditions in accordance to the channel quality indicator index.

For these operations, many signaling protocols have been implemented between the User Terminal and the eNodeB in order to provide information about their respective context to both of the entities. The signaling section will be covered further in this document.

2.3 The LTE System Architecture

As described in previous sections, the LTE technology aims to create a virtual IP Layer link between every User Equipment Node and the IP servers of the Network operator's subnet or the Internet. The entity in the network's end that provides this connectivity is the PDN (Packet Data Network). This "tunnel" must not be disturbed by the users change in contextual profile i.e. moving to a different location with different speed and requesting for a different service. The advances that the Long Term Evolution[12][13] for the 3GPP produce appear to be mostly on the radio interface of the system but in actuality many core network adjustments have been considered. The new Physical Layer of the technology is sometimes mentioned as E-UTRAN (Evolved Universal Terrestrial Radio Access Network) whereas the evolution in the system core is called SAE (System Architecture Evolution). Together they form the EPS (Evolved Packet System) that describes the system by end to end. EPS presents a new communication description envelop technique called and EPS bearer. The entities of the core network of an LTE network each contain tables of all the activated EPS bearer used to carry information from and to an End User. Every EPS bearer dictates the QoS parameters of this information stream and therefore provides the requested user experience under normal system operation. An EPS bearer links the IP (or set of IP's) of a user with a specific throughput tolerance and a delay barrier. The EPS bearers and the other features of the EPS will be further analyzed in the following sub chapters of the document.

2.3.1 Block Architecture Analysis of the EPS

In this section we will see each individual physical entity of the LTE back bone[12][13] system with the network interfaces that interconnect them and allow for its operation. As we mentioned before, the EPS sub-nodes serve as a proxy between the LTE user equipment node and the PDN gateway node to provide IP connectivity over the internet as well as native services such as VoIP and other. This is being monitored and enforced by the EPS bearer session for each request that occur in the access network. Also security and privacy are being considered resolved due to low layer encryption. The following figure will show each entity separated by their scope in the EPS.

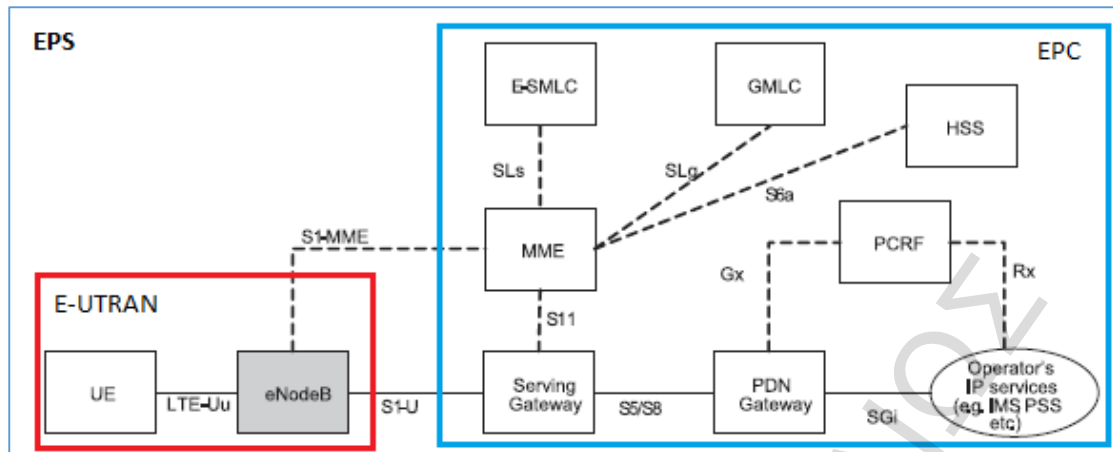


Figure 4 - The block architecture of EPS[12]

We will now analyze the procedures that take place in each of the involved backbone block.

- **The Evolved Packet Core.** This group of entities consists of the PDN Gateway node, the serving gateway, the MME (Mobility management Entity) and the E-SMLC (Evolved Serving Mobile Service Location Center)
- **The E –UTRAN.** This group of entities consists of the LTE UE and the LTE eNodeB along with their respective protocol stack.

The EPC consists of more nodes than the ones mentioned at (1) therefore the next part of this chapter is dedicated to explaining their purpose and functionalities.

- **PCRF.** The PCRF node is the entity in charge of policy enforcing, policy decision making and charging parameters for each user of the LTE network. It also authorizes the usage of certain QoS parameters in accordance to the user's subscription to the network and therefore is an important part of the EPS bearer node chain. The function that dictates the QoS of the users is called PCEF (Policy Control Enforcement Function).
- **GMLC.** The GMLC is supporting the Location Services by performing the authorization and requesting to the MME node for positioning requests. This is used for the LTE-Advanced operation of location determination.
- **HSS.** The home subscriber server is a database that stores each SAE subscriber's information such as each user's quality of service profile and also all the access permissions / restrictions (if any) that have been implemented by the operator. It also contains the high level routing information of which PDN

node should be used by each user to connect. This can either be by providing the Access Point Name (URL of the underlying network) or by the means of an IP address. Information about each user's respective MME node as well integration with the Authentication Center are some more features of this entity.

- **P-GW.** The Proxy Gateway is the entity in charge of granting the UE and IP address for its communication inside and outside the network. It also uses various ip tables to control the data flow of the ip packages in accordance to the QoS parameters provided by the PCRF node. The each traffic flow is described by a specified file called a "Traffic Template" which is being exchanged during the call initiation procedure. The most important quality of service class is the Guaranteed Bit rate bearer and the P-GW node is in charge of its maintenance. Some other functionalities of the P-GW node are interoperability functions for non 3GPP technologies such as the CDMA2000 and WiMAX.
- **S-GW.** The Serving Gateway is the central routing entity of the core network. All user packets (encapsulated in IP form) are transferred through this entity in their way inside and outside of the network. Another operation of this node is the mobility anchoring of each user equipment in the case of intra – eNodeB handoff of the traffic (the UE changes serving eNodeB). It also stores meta-data about each user equipment's previous EPS bearers during the IDLE state of the UE. It also serves as the mobility anchoring entity for legacy 3GPP technologies such as the GPRS and the UMTS networks.
- **MME.** Mobility Management Entity is the node that conducts all the signaling procedures between the User Equipment and the back bone network. This signaling is mostly referred as the *Non-Access-Stratum* protocols. The functions that these protocols conducts are :
 - **EPS bearer control.** The initiation, release and maintenance of the EPS bearers requested from the LTE terminal.
 - **Security and Connection control.** This procedure involves the security key exchange and handshake as well as connection establishment.
 - **Interoperability control.** The MME also dictates the horizontal handover procedure between LTE and other 3GPP legacy networks
- **E-SMLC.** The ESMLC node is responsible for the coordination and optimization of the resources used by the network to perform the location

identification of any user equipment served by the network. It also estimates contextual information about the UE such as direction of motion and velocity.

2.3.2 Analysis of signaling procedures

The procedures [1][2][3][4][5][6][7][8] that we analyzed in the previous chapter are crucial for the overall operation of the LTE core network. Most of these functions are being conducted between the MME and the LTE UE and they define the state machine of the system (IDLE, ESTABLISHING and CONNECTED) in a way that it will be required for the simulation tool we have developed.

In this document a thorough signaling study has been conveyed in order to determine the anticipated amount of bytes expected to be transmitted to the network due to specific procedures.

These procedures include:

- UE Attachment procedure to the network (initial)
- Data call initiation and data call release.
- Handover between two eNodeB cells (Intra-LTE)
- Additional periodical signaling messages exchanged in a small time interval during the user equipment's IDLE state.

The procedures are defined in 3GPP as mentioned in references and are illustrated in the figures at the end of this chapter. Each message has been evaluated separately according to the used protocol e.g., RRC, e GTP-C (Enhanced GPRS Tunneling Protocol), S1AP etc. According to this study, the following signaling bytes have been calculated (which are taken also into account in the evaluation of the decision making):

- UE attachment procedure: 1189 Bytes
- Call initiation procedure: 2013 Bytes
- Call release procedure: 944 Bytes
- Intra-LTE handover: 590 Bytes
- RRC Idle State Bytes/ user/ cell: 84 (with a transmission interval of 40 ms)

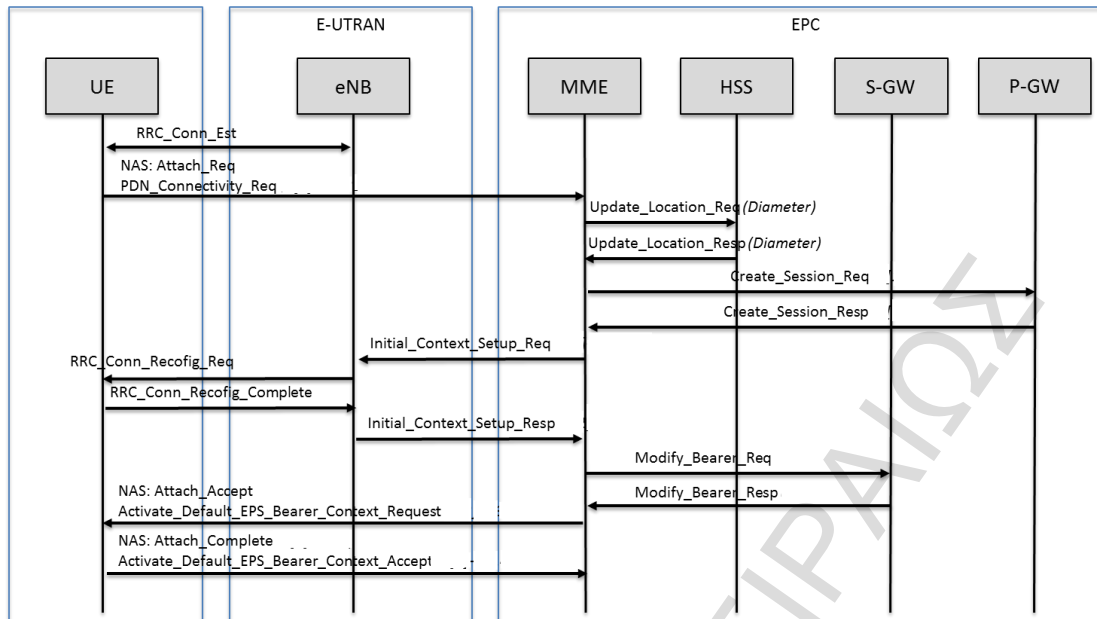


Figure 5 - LTE attach procedure

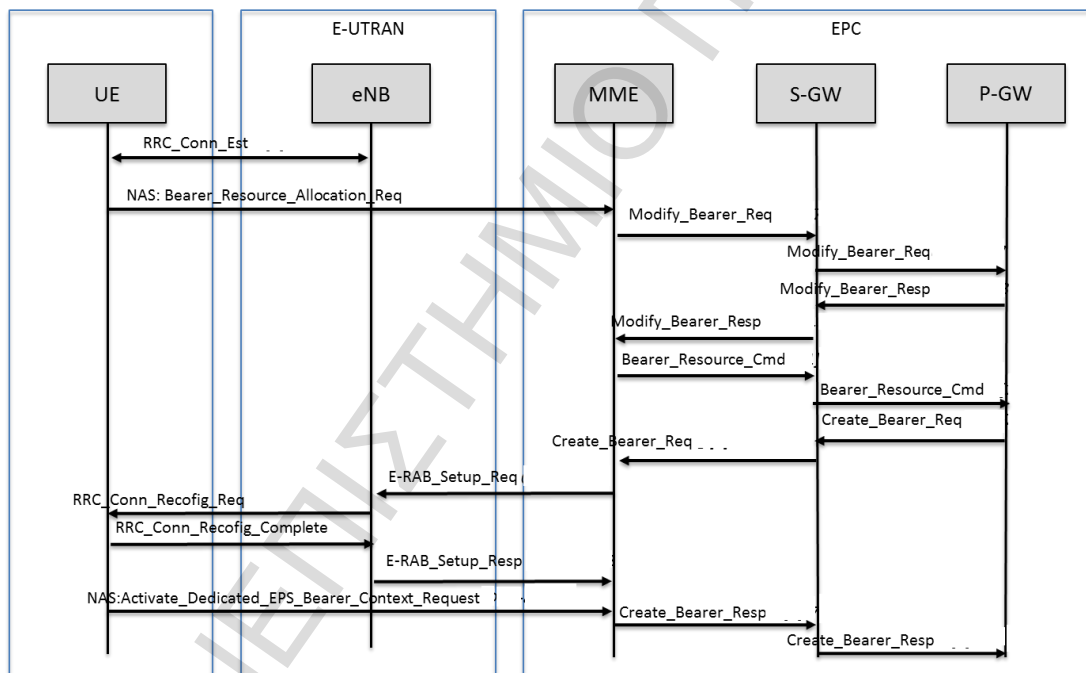


Figure 6 - Data Call setup request

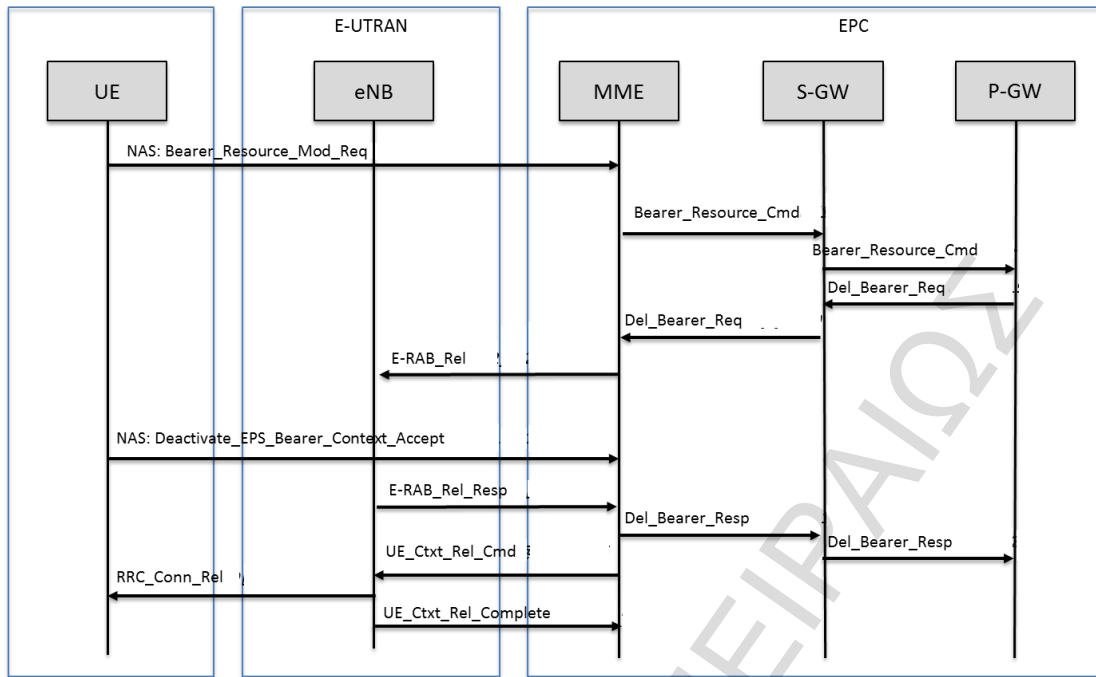


Figure 7 - Data Call Release request

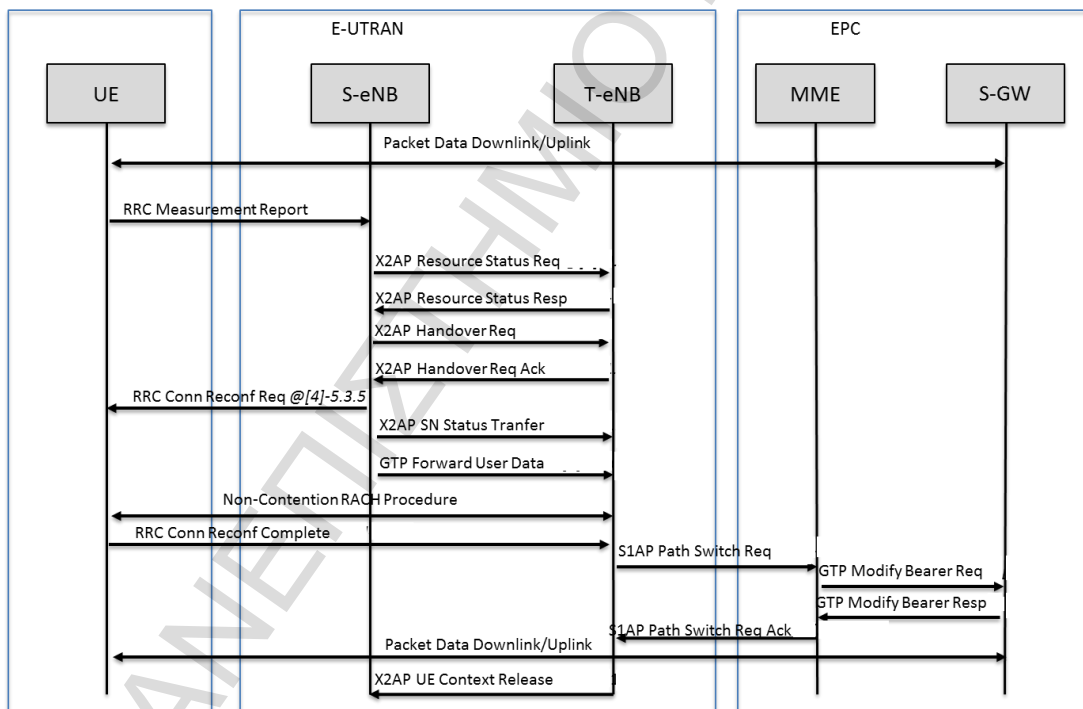


Figure 8 - Intra-LTE handover procedure

These procedures are of key importance in this study because after we analyze them we are using the produced data for our simulator in order to give an estimate about the signaling bytes that are being forwarded through the network. This metric essentially shows the amount of overhead these procedures have on the overall network resources of the system therefore we can compare this value for each of the Decision Making

cases in order to determine which one has the most negative effect on the backbone (and air) resources.

2.4 LTE Policy enforcement and the ANDSF template

Another aspect of the new cellular network brought by the evolution of the LTE standard is the creation of xml templates that oversee the behavior of the network nodes in many procedures such as cell selection and handover. This method of resolving complex problems can be characterized as asynchronous or offline simply because all the knowledge required for this function is acquired in the form of a preferences file at a different instance (on initialization of the UE). Given this decision-making instruction file, a UE is able to determine which RAT to choose to conduct the operation based on the preferences of the network operator and its underlying planning. ANDSF templates[9] contain more information in order to accommodate for different scenarios of use cases. These additional information varies from different time zones to different location areas according to the network operator's chosen level of control over the UE's actions. Another important part of the ANDSF policy exchange system is the ever-rising heterogeneous nature of the 4G++ networks. In theory and practice, the implementation of radio networks with different, co-existing access technologies such as LTE, UMTS and WiFi (also WiMAX) has shown very promising end results and therefore they need to be taken into serious consideration. Therefore, the ANDSF policy template accommodates for different radio access technologies assisting the operators to prioritize WiFi access to their private WiFi hotspots. This procedure lets the End Users benefit from the key elements of the different access technology while simultaneously does not disconnect them from the operator's network.

2.4.1 Implementation of the ANDSF policy exchange system

The ANDSF policy exchange system is essentially the policy enforcement method for the resource reservation and UE behavior that is implemented by the usage of a specially formed policy template. The exchange of this document is assisted by the usage of an application layer file transfer protocol, either FTP or HTTP, on a selected dedicated ANDSF server once a day/week. After the successful transaction of the document, the lower operational layers of the User Equipment are following the selected instruction set of the file and therefore their default functionality is overridden. Below we can see an example ANDSF template in one of its implemented forms.

```

Scenario 1:
./ANDSF/Name - TeliaSonera
./ANDSF/Policy/Set_1/RulePriority - 1
./ANDSF/Policy/Set_1/PrioritizedAccess/1/AccessTechnology - WLAN
./ANDSF/Policy/Set_1/PrioritizedAccess/1/AccessID - HomeRun
./ANDSF/Policy/Set_1/PrioritizedAccess/1/AccessNetworkPriority-10
./ANDSF/Policy/Set_1/PrioritizedAccess/2/AccessTechnology - WLAN
./ANDSF/Policy/Set_1/PrioritizedAccess/2/AccessID - Other_SSID
./ANDSF/Policy/Set_1/PrioritizedAccess/2/AccessNetworkPriority-20
./ANDSF/Policy/Set_1/PrioritizedAccess/3/AccessTechnology - 3GPP
./ANDSF/Policy/Set_1/PrioritizedAccess/3/AccessNetworkPriority - 30

```

Figure 9 - Example of an ANDSF template[9]

We can see that the basic structure of the ANDSF template is a method of comparison between different RAT's in the form of a scalar called access network priority. The additional information provided helps to identify the underlying networks and link them with their respective value of priority. These fields are the minimum required fields for the enforcement of policy-based decision making in an LTE system. An extension to this form can be created by adding more context information creating a more elaborate but accurate instruction set. Below we see an example of the extended form of ANDSF policy template.

```

Scenario 2:
./ANDSF/Name - TeliaSonera
./ANDSF/Policy/Set_1/RulePriority - 1
./ANDSF/Policy/Set_1/PrioritizedAccess/1/AccessTechnology - WLAN
./ANDSF/Policy/Set_1/PrioritizedAccess/1/AccessID - HomeRun
./ANDSF/Policy/Set_1/PrioritizedAccess/1/AccessNetworkPriority-10
./ANDSF/Policy/Set_1/PrioritizedAccess/2/AccessTechnology - WLAN
./ANDSF/Policy/Set_1/PrioritizedAccess/2/AccessID - Other_SSID
./ANDSF/Policy/Set_1/PrioritizedAccess/2/AccessNetworkPriority-20
./ANDSF/Policy/Set_1/PrioritizedAccess/3/AccessTechnology - 3GPP
./ANDSF/Policy/Set_1/PrioritizedAccess/3/AccessNetworkPriority - 30
./ANDSF/Policy/Set_1/ValidityArea/3GPP_Location/1/LAC - 15
./ANDSF/Policy/Set_1/ValidityArea/3GPP_Location/2/LAC - 30
./ANDSF/Policy/Set_1/TimeOfDay/1/TimeStart - 0800
./ANDSF/Policy/Set_1/TimeOfDay/1/TimeStop - 1000
./ANDSF/Policy/Set_1/TimeOfDay/2/TimeStart - 1600
./ANDSF/Policy/Set_1/TimeOfDay/2/TimeStop - 1900
./ANDSF/Policy/Set_2/RulePriority - 2
./ANDSF/Policy/Set_2/PrioritizedAccess/1/AccessTechnology - WLAN
./ANDSF/Policy/Set_2/PrioritizedAccess/1/AccessID - HomeRun
./ANDSF/Policy/Set_2/PrioritizedAccess/1/AccessNetworkPriority-10
./ANDSF/Policy/Set_2/PrioritizedAccess/2/AccessTechnology - 3GPP
./ANDSF/Policy/Set_2/PrioritizedAccess/2/AccessNetworkPriority - 30
./ANDSF/Policy/Set_2/ValidityArea/3GPP_Location/1/PLMN - 24433

```

Figure 10 - Extended example of the ANDSF policy template[9]

The illustration of the two examples assumes that the file form of the ANDSF template is of the WBXML encoding. In the extended version of the ANDSF policy template that additional contextual information is included to accommodate for various use cases of the operation. Different rule sets with different priorities and also the information of the Location area of the user and the time of day that the rule will apply are one of the key differences between the simple and extended form of the ANDSF template.

To summarize the ANDSF policy template we see that an ANDSF template may include:

- The SSID of a network access node
- The RAT of a network access node (LTE, WiFi , other)
- The priority as a scalar value of a network access node to implement the decision making – selection procedure
- A specific time of day that the rule will be implemented
- A specific location in the form of Location Area Code (LAC) that the rule will apply
- Other implementation specific information such as PLMN identity of a node

One key aspect of the ANDSF policy technology is that it can be extended and altered to fit each network operator's specific architectural innovations/additions making a very flexible policy enforcement mechanism of the future networks.

2.4.2 Our implementation of the ANDSF policy template

For the sake of this document, a different version of the ANDSF policy template has been created with the usage of the knowledge modeling language XML in order to help us integrate it more easily with the developed software and also to create an estimate about the size of the file to calculate the overhead that the ANDSF policy exchange procedure has on the normal operation of the system. Below u can see our manifestation of an xml ANDSF template as it will further be used for our simulations and input data.

```

2  <andsf>
3  <andsftimezone id="1" start="8:00" end="20:00">
4  <andsfentry>
5  <technology>LTE</technology>
6  <ssid>Lte_Net</ssid>
7  <priority>30</priority>
8  </andsfentry>
9  <andsfentry>
10 <technology>WiFi</technology>
11 <ssid>prive_net</ssid>
12 <priority>20</priority>
13 </andsfentry>
14 <andsfentry>
15 <technology>WiFi</technology>
16 <ssid>*other*</ssid>
17 <priority>10</priority>
18 </andsfentry>
19 </andsftimezone>
20 <andsftimezone id="2" start="20:00" end="8:00">
21 <andsfentry>
22 <technology>LTE</technology>
23 <ssid>Lte_Net</ssid>
24 <priority>10</priority>
25 </andsfentry>
26 <andsfentry>
27 <technology>WiFi</technology>
28 <ssid>prive_net</ssid>
29 <priority>30</priority>
30 </andsfentry>
31 <andsfentry>
32 <technology>WiFi</technology>
33 <ssid>*other*</ssid>
34 <priority>20</priority>
35 </andsfentry>
36 </andsftimezone>

```

Figure 11 - Our implementation of the ANDSF template

It is clear that the same information is included in the document but the presentation form has been changed in order to group-up better the different rule sets in conjunction with their respective time zones and id's. This implementation is easily parse-able meaning that the program during runtime can read the document and extract the different priority scalars to execute the decision making mechanism that we have implemented.

2.5 LTE and Heterogeneous Networks

The LTE cellular technology [12][13][11] is not a reinvention of the cellular network systems. It coexists with all the previous legacy cellular mobile technologies such as the UMTS family and the GSM family. It also has the ability to operate in triple mode meaning that every User Equipment served by the network may perform a change in the mobile technology according to the network's decision, the channel quality or the high concentrated load on a curtain node. Apart from the radio access technologies with the same standardization consortium, there are technologies of the 802.11x (WiFi) family and also the 802.16x (WiMAX) that are being considered part of the heterogeneous network. This happens because the notion of the cellular network has evolved to a network of many possible radio access methods that collaborate in order to give to the users the mobile internet experience.

Another aspect of the heterogeneity inside cellular networks is the existence of nodes with the same radio access technology but with different coverage area, complexity and

back bone connection. This applies mostly to the 3rd and 4th generation LTE base stations and it generates the system set of {macro-cell, micro-cell, pico-cell, Femto-cell}. In this set the components are laid in order of coverage area, but there are other criteria that can diversify them. For example the Femto-cell access point is connected to the core network of the provider over the Internet by the means of an encapsulated S1 protocol. This means that the back-bone throughput is subject to the quality of service restrictions of the provider of the ADSL / other Broadband wired technology which can degrade the overall system performance.

Micro-cells and Pico-cells only differentiate in the coverage area they serve. They are placed in specific geographic locations inside the coverage area of a larger macro cell and they function as traffic off-loaders with increased performance and throughput.

The **WiFi** access points that we mentioned earlier, can be deployed by the operator to serve as emulated Pico-cells with monitoring mechanisms that will use the unlicensed band of the 2.4 GHz in order to serve the great demand for Internet traffic.

All the nodes that we have described so far are active network entities meaning that they include all the functionalities of the macro-cell but in a smaller scale. In the heterogeneous networks that will be part of the future Internet exists another piece of heterogeneity which has only passive functionalities. This is the **RRH** (Remote Radio Head) or the network edge regenerative repeaters. These cells use the radio channel as the back bone for communicating with the macro cell that controls them. They reinforce the signal that is meant to be for a target UE and therefore they greatly improve the performance of the network. Their intelligence is limited to the lower layers of the network therefore they can be described as passive nodes.

The Macro-cells, the micro and pico cells, the Femto cells, the RRHs and the WiFi hotspots are the elements that compose the heterogeneous networks of the next generation's cellular technology.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Chapter 3: Intelligence and Decision Making in the LTE

An LTE user equipment device has the freedom of mobility in an LTE network. The user can roam in the serving area by walking, using a bicycle or a car with various speeds without having any discontinuousness in internet connection or the voice calls he conducts. This happens because in a heterogeneous cellular system, there are many cell nodes of different radio access technologies that communicate with the LTE decision making network providing all the users with sufficient radio resources to maintain their services. The procedure of handing over a data call session (with the underlying EPS bearer) from one node to another is the focus of this study therefore this chapter is dedicated to its analysis and formulation.

In this document we are approaching the handover procedure of a connected User Equipment node by the means of optimization. This means that every possible candidate will be evaluated by the use of a fitness (or cost) function and the best will be chosen as the target for this procedure. In the case of the active radio access technology being different than the result of the optimization function, a handover procedure is initiated transferring the service to the new node.

For the evaluation of the decision mechanism for cell selection, a weighted objective function is developed in order to allow the different decision-making contextual factor to gain a controlled effect on the decision making functionality. The objective function takes into account attributes that reside both on the network side and on the LTE UE side and transforms them into a scalar with value range of [0,1] in order to be controllable by either threshold or margin techniques.

By considering the aforementioned aspects the fitness function (F) can be defined as follows:

Maximize:

$$F_{ij} = w_1 \cdot \frac{SNR_{ij} \cdot S_{v_i, T_j}}{\max_{(i,j) \in U \times C} (SNR_{ij} \cdot S_{v_i, T_j})} + w_2 \cdot \frac{L_j}{\max_{j \in C} L_j} + w_3 \cdot \frac{w_{ANDSF,j}}{\max_{j \in C} (w_{ANDSF,j})}$$

(1)

Subject to:

- UE $i \in U$ where U is the set of all the UEs of the area.
- RAT cell $j \in C$ where is the set of all the available RAT's of the area.
- Weights: $w_1 + w_2 + w_3 = 1$

The parameters w_x represent the weight of the corresponding attribute.

This complex RAT evaluation function uses different contextual information taken from many contextual aspects of the System. Below you can find explanation for each attribute as well as description on the means of their retrieval from the decision-making entity.

- The **Signal-to-Noise Ratio (SNR)**, so as to be able to simulate from each UE the signal strength received from each cell. Obviously, the higher values of this attribute, the better there are for the decision of cell selection. This metric is obtained by the UE's Physical layer measurements of signal strength for each RAT's broadcast beacon and uses static Noise Power (relative to the channel width) to determine the SNR;
- The **cell type** factor T (e.g., macro base stations, small cells) from the saved configuration with the LTE eNodeB that describes the surrounding network;
- The velocity v of each UE and the type T of the cell j . To this respect, the cell type to user **velocity factor S** is defined. This attribute links the cell type with the UE velocity, since UEs with higher moving speeds, are expected to experience more frequent handovers, so the transitions to small cells or Wi-Fi APs would not be preferable. An LTE UE is able to discover it's movement speed with various ways such as the GPS service, embedded accelerometer or the usage of the neighboring cell's reference signals (LTE-Advanced rel. 10);
- The **cell load** which will be denoted with L . When a cell is highly loaded, it would not be beneficial to handover more UEs to it. The cell load of each cell is transmitted to the decision making macro cell via either the X2 interface or an external connection through the LTE network back bone. The cell load is the amount of resources used by each of the candidate RAT's and it's value's range is $[0,1]$;

- The **ANDSF weight** (w_{ANDSF}), which is calculated for each available cell (macro, pico, Wi-Fi) derives from the added ANDSF policies to the ANDSF management entity. The ANDSF policies include technology (e.g., LTE, Wi-Fi), SSID and priority (e.g., Wi-Fi will be preferred from LTE if available or vice versa etc.). Information related to policies according to the current time of day are implemented as well (e.g., offloading could be triggered during busy hours only). The ANDSF weight information is stored with the ANDSF template file that either resides inside the LTE-eNodeB's database or it is downloaded to the LTE UE's cache for use;

Figure 12 provides a message sequence chart of the UE-initiated decision making procedure. Specifically, as soon as the procedure begins, the UE start requesting information (periodically) related to the type of serving cell and neighboring cells as well as their current number of users served (Request_Info_Bundle). The serving cell, requests the related status information from the neighboring cells (Request_Status_Data). When the serving cell collects the requested information from the neighboring cells (Response_Status_Data), the acquired information is sent back to the UE (Response_Info_Bundle). Then, the decision is made based on the objective function, and if it's necessary, handover is executed.

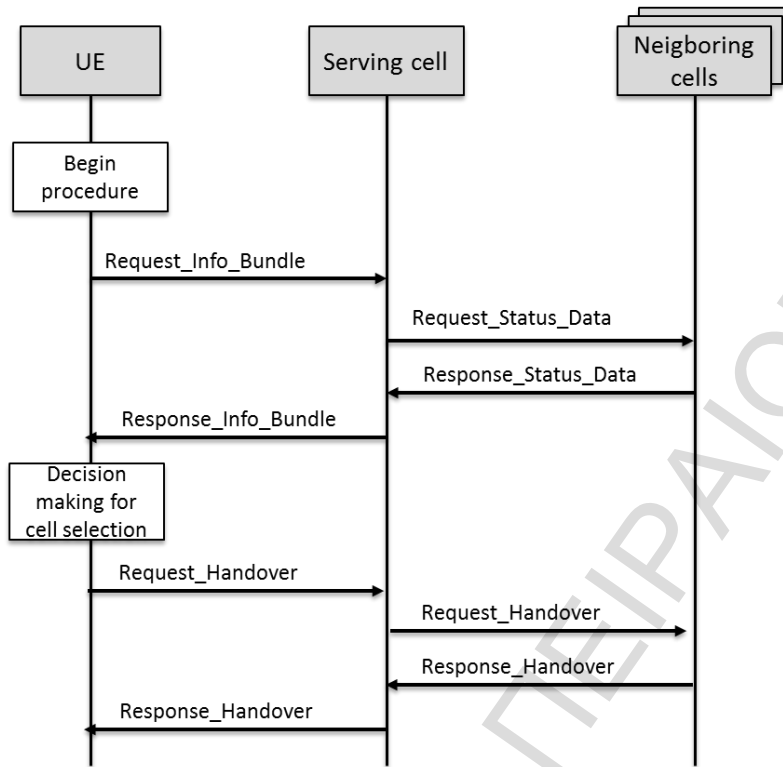


Figure 12 - The Decision making sequence diagram

Chapter 4: Development of the Simulation Tool

For the purpose of this study we have designed, developed and configured a custom made Java based Simulation / Calculation tool that will be used to test the different Decision Making scenarios (User Equipment , Infrastructure) in simulated real time deployments and use the results that will be produced to reinforce the outcome of this research. The creation of such an elaborate software is a very extensive task as many parameters have to be taken into consideration before its creation. One really important factor that a simulator must seriously consider is the detail-to-efficiency ratio. To be put in simple words, a simulator cannot be usable if given the average computational capabilities of the reference PC system, it cannot produce fast enough results for an extensive simulated time period. On the other hand, if a simulator is fast but takes shortcuts in many key aspects of the underlying study, then it will trade the accuracy of the long simulation with the inaccurate results of an on-time outcome. Therefore before and during the development of tis tool we have paid constant attention to this tradeoff and also to the State of the Art in similar simulation Systems such as the NS3 Simulator and its sub-module for LTE simulations LENA.

4.1 State of the Art and Beyond SotA for LTE Simulation

In the scientific field of mobile network simulation there are many contributors each providing the scientific community with the means to study, understand and develop various network system components and/or algorithms. For this research we have searched the existing state of the art simulation systems in an effort to avoid “reinventing the wheel” in the area of simulator software. Although there are many promising LTE simulators in the open source community, each of them presents with different drawbacks for the satisfaction of this particular research’s requirements. In the following paragraphs, we analyze them and comment on the level of compatibility they have with our system requirements.

- **ONE Simulator.** The ONE simulator is a software platform developed by the SINDTN and CATDTN project supported by Nokia Research Center (Finland), in the TEKES ICT-SHOK Future Internet project, Academy of Finland projects

RESMAN and Picking Digital Pockets (PDP), and supported by EIT ICT Labs. The simulator's capabilities are the generation of node movement using different movement models, the routing of messages between nodes with various routing algorithms and the visualization of the nodes by the means of a graphical user interface. The one simulator is not an LTE dedicated simulator and many systemic components and functionalities are missing or need to be implemented by the developers.

- **OPNET.** The OPNET Modeler, developed by Riverbed Corporation is a licensed, close source LTE simulator compatible with all of today's protocols and technologies. It provides visualization tools for all network metrics and has implemented the majority of the LTE system protocol stack. The closed source nature in conjunction with the cost for the license acquisition led us to the conclusion that OPNET modeler is not a suitable software for our study.
- **NS3.** The Network Simulator 3 is a vast simulator family supported by the open source community with a plethora of network modules and protocols that have been implemented and distributed to perform tests and simulation scenarios. It is created by the NSNAM organization and uses C++ and python programming languages that are most suitable to operate in Linux Systems. The NS3 suite contains an LTE module (codename LENA) that has a very accurate, although under development, implementation of the various protocol stacks of the LTE network. That led us to the conclusion of starting our simulation efforts with the NS3 simulator.

By the means of the NS3 simulator, this research has started to develop small topology scenarios to get an actual look on the input, output and configuration capabilities of this software. The NS3 simulator uses C++ or python scripts that describe the system parameters of the simulation and lacks other graphical user input elements. Also the LTE system variables such as transmission, scheduler, and resource allocation preferences are hard to adjust due to the complexity of the C++ modules. The runtime of the Simulation in simple operation mode is exclusively command line print functions and the output consists of various packet trace and txt files containing measurements for the Physical Layer parameters. In the case of executing the simulation using the

visualization tool PyViz (optional package of the NS3 installation), we are able to view the LTE UE nodes and the LTE eNodeB in the system and also the capacity of the radio links that are active. However the lack of visual information about the context has led us to further develop the graphics engine of the PyViz suite in order to better illustrate the context of our system. Therefore we have proceeded in altering the NS3 source code to create a better visual environment for our simulation.

Graphical Improvements:

Pre-Changes: The EPC, Remote Host, LTE eNodeB, LTE UE and WiFi hotspot nodes where initially drawn as plain circles without any additional information apart from different coloring (grey – red) which indicated that the nodes where using a special mobility model or they were static.

Post-Changes: The python visualizer now properly displays every entity with an appropriate image icon of the image type .svg (scalable vector) in order to give the viewer an accurate impression about the simulation topology. In detail: For the EPC and Remote host entities we have selected generic backbone network device icons, for the Lte UE node we have selected a mobile phone icon , for the LTE enodeb we have selected a telecommunications antenna and for the wifi we have selected a wifi hotspot image. Also we have added the option to display a label next to every Node which, in the case of enodeb displays the cell id or in the case of a LTE ue it displays the IMEI number of the device.

Modified File: core.py (src/visualizer/visualizer/core.py)

Changes: Line 764 -> Class identification has been chosen as the method to switch the different graphical modes. With the class type as a key, different icon and labeling has been implemented according to the previous paragraph. Also the different node color in relation to the mobility model chosen has now been removed in order to focus the diversity to the different icons.

Modification in the LTE Model:

Pre-Changes: Some objects inside the namespace of the entities “LteHelper” and “LteEnbPhy” where not visible by a method of the template GetObject<?> or by any public method.

Post-Changes: Public Method was created to trace the pointer of the PathlossModel used by the Lte Modules in order to calculate, from the point of view of the simulation, the free space losses and make the decision of doing a handoff request.

Modified File: lte-helper.cc (src/lte/helper/lte-helper.cc) and lte-helper.h (*lte-helper.h)

Changes: lte-helper.cc -> Line 222: Public function named GetPropagation() was created in order to return to a user the value of the Path loss Model used in downlink transmission. Simple return request inside function body. lte-helper.h -> Line 64 :GetPropagation() function defined

Below u see the comparison between the two versions of the modified NS3 installation:

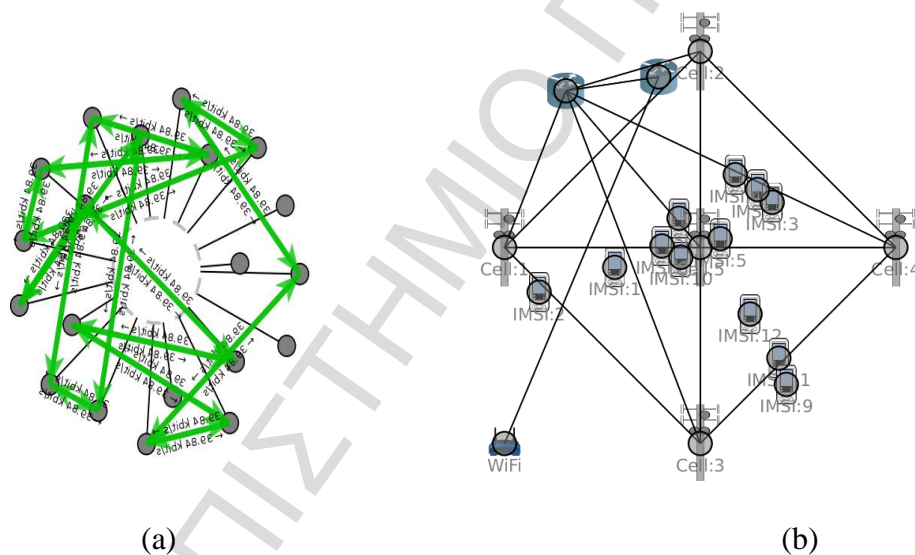


Figure 13 - (a) Before the Graphical modification and (b) after the Graphical modification

Although the improvements were made successfully, it was clear that the NS3 Simulator was not a suitable tool for the course of this resource. This was due to the fact that apart from the graphical poorness, the NS3 has not yet implemented (December 2013) the majority of the protocol stack of the LTE system, and especially the layers that conduct the decision making operations and begin the signaling procedures as analyzed in previous chapters.

Therefore we gathered all the knowledge acquired from our experiences with this System and begun the development of a new Simulation tool built into the needs of the study for the optimal placement of the Decision Making function for the LTE.

4.2 Requirements Analysis

After our experience with the state of the art in LTE simulator software, we proceed in the creation of the requirement list for the implementation of our custom made micro simulator that will help us further conduct this research. The programming language chosen must be a language with all the advanced programming capabilities such as multi-threading, graphical user interface friendly to a novice or advanced user, a way to serialize the input and output of the simulation in a readable but also parse-able manner and the simulation time must be fairly fast in relation to the run time. So to summarize the system requirements of this software we have:

- 1 A micro simulator based on discrete event simulation implementation.
- 2 A flexible object based multi-threaded programming language.
- 3 Simulation input capabilities created with a graphical user interface to account for all the possible scenario variations and network deployments.
- 4 Runtime projection of the simulation's context: This includes the position and mobility of the UE's, the position of the various RAT's and their respective coverage radius, the illustration of network-oriented events such as handover initiation and data call start / release procedures.
- 5 Runtime capabilities of altering the input parameters creating a dynamic context for the underlying simulation.
- 6 Fast simulation time in comparison to the run time.
- 7 Dynamic API for measuring different output aspects of the simulation.
- 8 Graphical display of the output results.
- 9 Creation of Excel files for meta-processing of the output data.
- 10 Serialization of the input parameters in the form of execution scenarios and stable random generation in order to recreate the same input/output pairs.

For this implementation we have chosen the programming language JAVA combined with the usage of several apache and gnu licensed API's to help us avoid "reinventing the wheel" in certain aspects of the implementation.

4.3 System Design

4.3.1 Simulation Engine

This research circles around the two different decision-making paradigms, the UE initiated and the Network initiated, in relation to the rich and vibrant context of a

heterogeneous mobile network. We have chosen to implement a discrete event micro-simulator and not a macro simulator because we wanted to analyze the decision making procedure step by step as it happens during the real time operation of this kind of system. Therefore the usage of complex statistical assumptions of the macro simulator where not considered optimal for our implementation.

Every discrete-event simulator program has at its heart a main object that handles the initiation and execution of events. This object is often called the Event Queue and in our case it is called the Simulator class. The model behind an event based simulator is pretty simple:

1. The simulation starts with an initiation event, sometimes called “The Spark Event”. This Event gets invoked instantly and begins with the re-initiation of the data structures of the System.
2. An Event is an Object with a very specific structure. It has a discrete serial number differentiating it from other events, a specific time stamp for its execution and a functional implementation in its payload.
3. The simulator iterates all the events that it has stored in its event queue. If he finds an event that its execution time matches the time of the Simulator, then the event gets executed and removed from the event queue. If such event is not found then the Simulator’s clock is increased by one unit and the procedure restarts.
4. There are three types of events in general. A static event is an event that happens once and has no relation with other events or itself. A periodic event is an event that, at the end of its execution creates a new instance of itself and gets queued again in the simulator creating the continuum of an ever functioning machine. Finally there is the chain event. The chain event is an event that gets initiated by another event and at the end of its life it generates an event of a different type. This sort of events are used for the simulation of events that macroscopically form a large procedure such as the Decision-Making functionality of the LTE.
5. A key aspect for the realistic usage of the simulator is the timestamp of an Event. Every event is created at specific time instance of the simulator and **must be timed to execute at the exact time difference (Dt) that it would execute given the specific state of the underlying system.** For example if an event is the

delivery of packet in a network, then the execution of the “delivery” method must be set in $\text{Time_of_Initiation} + \text{Time_for_delivery}$ according to the underlying network’s capacity and delay model.

- The machine of the Simulator class is not specific to an LTE system. It can work with all kind of event based simulations with the creation of the custom made events that influence specific data structures.

A discrete event simulator must have a way to scale time effectively. Every simulated second must be equal to a real second so all the functions that use time as a variable must be re-scaled in order to operate with the basis of the fraction of time that has occurred. If everything is done correctly, then the simulator engine can recreate any kind of real life events and give results with any chosen accuracy wanted. A simulator can also have information about the duration of the simulation, and the specific date that the simulation will commence. I.E. a simulator may be instructed to generate a simulation of 1 real hour for a system at the 8th of December 2013. The date of the Simulator engine is completely different of the System time. All events that generate functional invocations use the time of the simulator as a basis for their time based operations therefore the continuity and coordination is resolved.

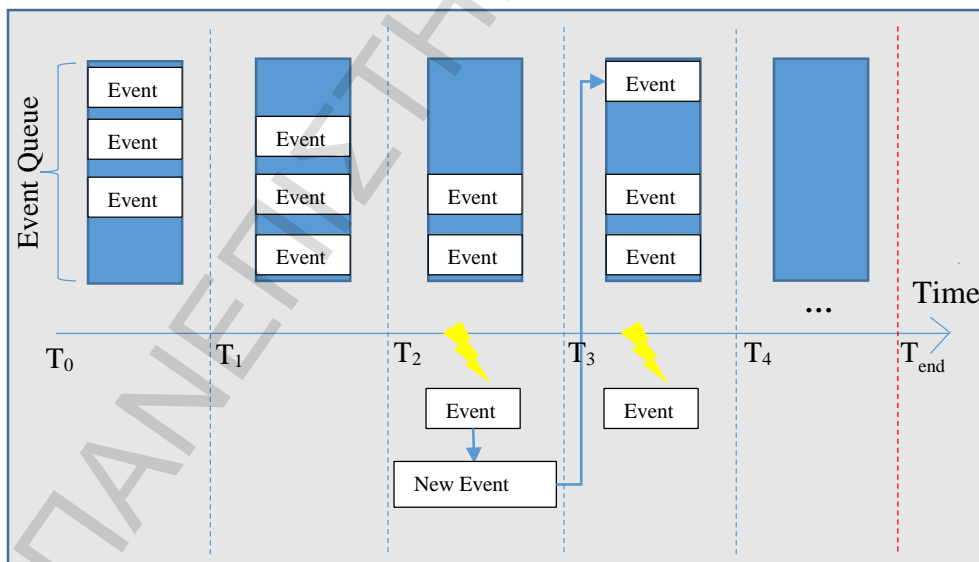


Figure 14 - Basic Operation of the Event-Queue

As we see in the figure, the basic principle of the Simulation is the iteration of the stored events, the execution of events timed in the T_{Present} and their interaction with the event

queue in the form of creating new Events and repeating the cycle until the simulation has stopped.

4.3.2 System Modules and their relations

Given the selected operation engine (Discrete Event Simulator) we have the freedom of designing the operational modules in the best way that they will fulfill the purposes of this research. Therefore we need a System module that will act as the LTE User Equipment module and its operation must simulate the operation of a real LTE UE and also a System module that will act as the LTE eNodeB and any other Radio Access Technology node that has the ability to serve the packet based traffic requested by the LTE UE. These are the basic operational modules of the simulated network. The exchange of signaling and traffic bytes between these two nodes is being simulated and then measured to generate data that will have the exact same behavior in the case of a real time system. These modules can be analyzed further to a few smaller parts that can be described as different layers. Each layer, as with network stack layers, operates like an isolated block with very specific end-points where it communicates with its neighboring layers. Below we analyze each of these two entities into their respective parts with a small summary of their functionalities:

LTE User Equipment sub-Modules:

- The LTE UE Application Layer. This Layer is represented by a container that includes instances of simulated programs (or Applications). These applications behave differently with the passage of time and therefore they create their own kind of events and graphical representations.
- The LTE UE Network stack Layer. This Layer summarizes all the functions that an LTE UE device will perform in order to communicate with the application layer and serve the requested traffic by transmitting it to the Network stack layer of the serving LTE eNodeB. This is the layer that performs the decision making functionality that surrounds this research and also the signaling procedures of cell attachment, call initiation, call release and handover procedures.
- The LTE UE Energy stack Layer. This experimental layer communicates with the LTE UE Network Layer in order to create an estimate about the power consumption of the device's operation. This has its own effects on the graphical display of the UE terminal.

- The LTE UE Mobility Layer. This layer is linked with all the previous layers in a way that each layer may require information that lies inside its context. Here the physical node that contains all this virtual layer is described in the parameters of physical type, position in the playground, speed, mobility model and other information.

Together all these data modules form the structure that this simulator understands as the LTE UE Module. The way these modules communicate may be described as a *stack* but can also be grouped like a graph or an asterisk diagram revolving the most important class the LTE UE Network stack Layer. The second main module of this program is the module that acts as the serving node of the network. Although this is an LTE network, this study revolves around the heterogeneous form of the next generation's networks. Therefore we have summarized many of the functionalities of a different RAT into this module creating the same entity with different functional operations, the Serving Station Module.

LTE eNodeB / WiFi sub-Modules:

- The LTE eNodeB Network stack layer. This layer communicates directly with the traffic of all the LTE UE network stacks that it's serving. This means that the requesting telecommunication traffic must be multiplexed and resolved through the radio channels and to be scheduled through time in order to succeed transmission. In the case of this module emulating the WiFi technology, different policy is implemented with regards to the multiple access technique of the specific underlying system. As with LTE networks, scheduling between users follows the round robin technique and other more complex systems, the WiFi access point splits all the available capacity to the users. This is an emulated form of the opportunistic CSMA –CA MAC layer.
- The LTE eNodeB Energy stack layer. This layer serves as the energy consumption mechanism that communicates with the upper layer (network) to determine the power consumption of each consecutive minute and to calculate the total energy that the RAT requires. This implementation is only functional for the LTE macro eNodeB entity.
- The LTE eNodeB mobility layer. This layer contains all the information that describe the physical entity of this node. The location, the speed (which is 0

since this is a static node) and also the coverage area of this object is stored within this data module.

These two compound entities are the main protagonists of the simulation procedure. The methods they implement are actual implementation of the emulated real systems and their invocation results in the generation of events that form chains and makes the output data realistic and accurate. For every different simulation paradigm, a basic class stores the summary of the modules by placing them in different containers to help every Object get access to each other. This class is the **Main** class of this program and is the communication point of the different graphical elements, the different data structures, the internal and external API's and the Simulator module. Below we can see the previous modules, the abstract view of the compound module they form and their relation to the Main module's context.

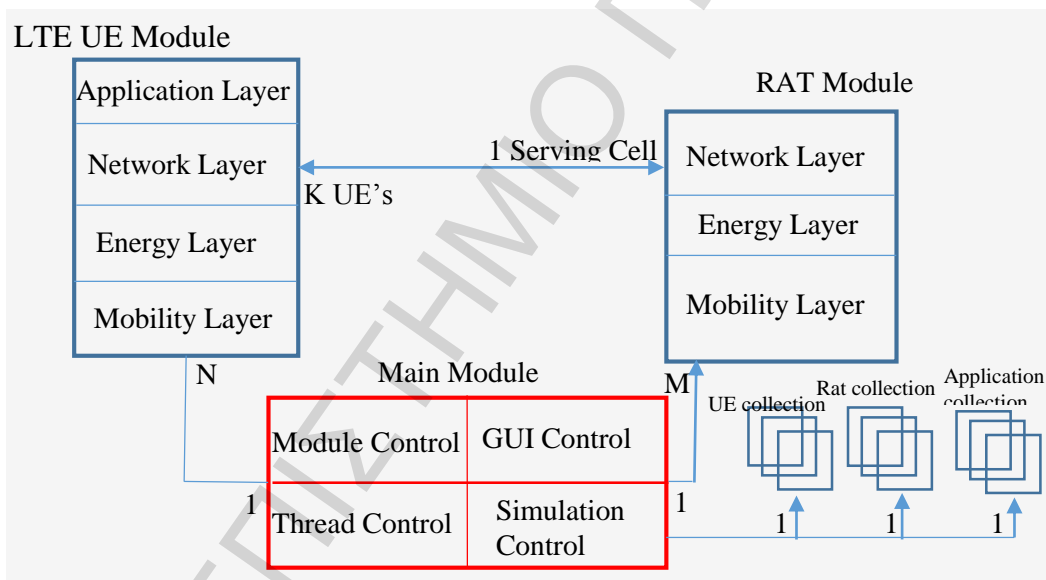


Figure 15 - Abstract Module Diagram

As it is clear in the diagram, the Main module controls many aspects of the system and is the second most important module, after the Simulator module, of this application.

4.4 Analysis of Interfaces and Abstract Classes

When someone develops a program, he needs to take into consideration the possibility of extensions, additions and many features that weren't in the initial system requirements. To make an application scalable and potent for many extensions, one needs to develop several programming interfaces that specify an abstract skeleton of

how a module works and interacts with its environment and then to generate implementations of this skeleton to account for the different extensions. In this program this logic is applied in a moderate scale and in different operation points and in future work we will consider applying it to more code parts in order to create a more sophisticated simulation machine.

In Java programming language there are two ways to create a functional skeleton:

- A programming *interface*
- An *abstract class*

The difference between the two data structures is that where the programming interface is a description of a set of Java function archetypes, the abstract class extends this to include also some basic local variables. To be put plainly, the abstract class is an empty implementation of an interface that needs to be extended in order to obtain functional qualities.

In this System architecture, we have chosen to use the following abstract classes because of the vast majority of implementations we foresaw that will be implemented.

- Abstract class Event. This class is the basic data structure that the Simulator iterates in order to determine the sequence of event execution for this program.
- Abstract class Application. This class is the basic object skeleton that communicates with the LTE network layer in order to generate network traffic.
- Abstract class Measurement. This class is revolving around scalar variables that need to be gathered from the system modules in a specific way, plot in its respective graphical representation and present as an output Microsoft Excel style sheet in order to meta-calculate various statistics.

These programming interfaces will be analyzed in more detail in the next chapters.

4.4.1 Abstract Class Event

This Abstract class is extended in order to generate a plethora of Events created to simulate the operational procedures of our simulated heterogeneous cellular network. The constructor of this class basically stores on the new object instance the information of the time that it will be triggered. Extensions of the constructor vary but they all summarize the initialization of the Main class instance that is used to communicate with all the other modules of the System. The fields of this data structure are being accessed

by simple get/set methods as seen in traditional java and the most important method it requires to implement is the “execute” method. Regardless of the Event implementation, the simulation class does not need to have any specific information about the execute method. It simply invokes it when the timestamp on the Event Object equals to the Simulation’s internal clock. Below you will find a list of the Event abstract class implementations that are being used for our simulation results along with a sort description of the underlying procedure they undertake.

- ANDSF Event. The ANDSF Event implementation is a periodic event for the simulator with a configurable interval of a *DAY*. Its execution invokes the “resolve ANDSF” method from inside the body of the LTE UE network module that initiates the download procedure of the ANDSF xml document required for the decision-making procedure.
- Application Enabled Event. This Event implementation is a triggered Event that is generated following each Application’s state machine life cycle. The activation or deactivation of a specific Application instance, hosted at a UE is controlled by the execution of this Event and also a relative notification appears in the simulation’s monitor.
- Application Event. This Event implementation is a periodic event that occurs every *Second* of the Simulation. It invokes for each member of the Main module’s Application instance the method “next state” which resembles the Markovian Chain Model next state procedure. Every second, each application with different probability may transit from the IDLE state to the ACTIVE state or the HIGH_ACTIVE depending of its respective “next state” implementation.
- Attach Request Event. This Event implementation is a part of the sequence of Events that are required for a UE to complete the Attachment procedure (as shown in previous chapters of this document). It represents the initiation or the *Request* of this procedure and also it invokes the periodic Event of the Decision making functionality.
- Handoff Check Event. This Event implementation is the first sub-Event that synthesizes the Decision-Making functionality. It is a periodic event with the interval of the selected decision-making interval value from the GUI. It has two different operation cases, one for network-initiated and one for UE-initiated decision-making operation.

- Handoff Info Exchange Type 1 Event. This event implementation simulates the exchange of missing information from the UE device to the network in the case of the network-initiated decision making mechanism. It also triggers the next part of the operation which is the Handoff Process event.
- Handoff Info Exchange Type 2 Event. This event implementation simulates the exchange of missing information for the Network to the UE in the case of UE-initiated decision making mechanism. It also triggers the next part of the operation which is the handoff process event.
- Handoff Process data bundle Type 1 Event. This event implementation simulates the processing of the decision making mechanism in the case that it is resolved at the UE (UE-initiated decision making).
- Handoff Process Data Bundle Type 2 Event. This event implementation simulates the processing of the decision making mechanism in the case that it is resolved at the Network (eNodeB).
- Handoff Request Acknowledgement Event. This event is sent as a confirmation mechanism for the completion of a handoff event. It involves the execution of the “Handoff Complete” method at the UE module.
- Handoff Request Event. This event simulates the operation of “Intra-Lte Handover” and therefore uses the delay from the 3GPP standards and also the signaling study conducted in previous chapter.
- LTE Add Acknowledgement Event. This event simulates the response from the Attachment Event that ensures the UE it is successfully connected to the eNodeB and move to the “IDLE” state.
- LTE Function Event. This event is the implementation of the mechanical function of each simulated LTE eNodeB and LTE UE node of the system. It has a preset frequency of occurrence (100ms) and resolves all the functions that would had been completed during that time interval in each device.
- Main Event. This event is referred also as the “Spark Event”. This event initializes the simulator with the first events according to the input parameters of the GUI and begins the simulation.
- Measurement Event. This event is the implementation of the Event class that simulates the measurement of each output parameters chosen to be monitored

and displayed. It communicates with the next programming API that we will analyze the Measurement API.

- **Mobility Event.** This Event implementation simulates the spatial displacement, the movement of the UE nodes through the playground in accordance to the selected movement parameters (speed, movement model). This event is a periodic event that occurs with an interval that is pre-calculated in order to gain accuracy in relation to the monitor resolution.

These events synthesize the operation of the Simulation along with the state machine of the different modules, the graphical elements and the Simulator Core Engine.

4.4.2 Abstract class Measurement

The measurement class skeleton is a programming structure that allows for the monitoring of certain variables of the Simulation, their display in custom made graphical plots, the formatting of special excel style-sheets and the communication with the user preferences from the Graphical User Interface. It provides a Boolean field that communicates with a corresponding graphical element (CheckBox) that dictates whether the user wishes this particular quantity to be measured and shown as an output of the simulation. Also has methods that require custom implementation (abstract methods) for each different measured quantity. These methods are:

- The “Make Plot” method. This method’s implementation generates an Object of the JPanel class that is the product of the custom plotting of this element’s dataset. This is used after the simulation runtime to be viewed by the user.
- The “Measure” method. This method’s implementation is responsible for sampling the sum of the Simulator’s context for the parameters that are required for this data set. It is invoked at the Measurement Event, covered at the previous section of this chapter.
- The “Make Excel” method. This method’s implementation uses the Apache POI API in a specific way to generate custom made excel sheets with the simulation data, formatted for easy plotting and meta-processing.
- The “Clean-up” method. This method is invoked after every simulation to ensure that all the data structures inside this class will be reset to default state and ready for the next set of measurements.

We can see that the measurement API is very important for the system because it generates all the information we want to extract and analyze for the purpose of this study. Below we see a list of measured Elements that are accessible to a user

1. The eNodeB Downlink Bytes Measurement. This extension measures the amount of bytes (payload) that are being transferred from a certain eNodeB serving the Downlink of all the UE's. It can also be referred to as the outbound backbone traffic.
2. The eNodeB Downlink Throughput Measurement. This extension measures the rate in which the amount of bytes (payload) in each eNodeB change. It can also be described as the outbound backbone traffic throughput.
3. The eNodeB Uplink Bytes Measurement. This extension measures the amount of bytes (payload) that are being transferred from a certain UE to its target eNodeB. It can be described as the inbound backbone traffic.
4. The eNodeB Uplink Throughput Measurement. This extension measures the rate in which the amount of bytes (payload) in each eNodeB change. It can also be described as the inbound backbone throughput.
5. The channel quality indicator Measurement. This extension measures for each of the system's User Equipment device the channel quality indicator index. It is used to determine the average quality of service given the selected decision making mechanism.
6. The Handoff Measurement. This extension measures for each of the system's User Equipment device the amount of total handover procedures they have conducted in order to determine the effectiveness of each decision-making paradigm.
7. The Cell Load Measurement. This extension measures for each of the system's available Radio Access Technology node the amount of load it has at the specific moment of measurement. The value of the cell load can vary from 0 to 1 according to the way its respective technology handles the traffic.
8. The Signaling Byte Measurement. This extension counts for every eNodeB device of the system, the cumulative byte count of the signaling procedures as they derive from the analysis of our study.

9. The Signaling Throughput Measurement. This extension measures the rate in which the signaling bytes in each eNodeB change to determine the throughput on those specific channels.
10. The SNR Measurement. This extension is used to measure for each of the UE devices connected to the system, the channel state of each RAT within the range of their antenna. It provides plots about the channel quality of each cell.
11. The PDF of SNR Measurement. This extension is used to create a statistical distribution (Probability density function) showing the collection of the average Signal to noise ratio of each LTE UE in the playground.
12. The CDF of SNR Measurement. This extension is used to create the statistical cumulative density function for the PDF of the SNR in order to show the probability of a UE enter an area with a given SNR value.
13. The Downlink Bytes of UE Measurement. This extension is used to measure the total cumulative bytes that have been served by either radio access technology for the Downlink channel of each LTE UE device.
14. The Downlink Throughput of UE Measurement. This extension is used to measure the rate in which the downlink bytes in each LTE UE change creating the metric of the Downlink Throughput.
15. The Uplink Bytes of UE Measurement. This extension is used to measure the total cumulative that have been served by either radio access technology for the Uplink channel of each LTE UE device.
16. The Uplink Throughput of UE Measurement. This extension is used to measure the rate in which the uplink bytes in each LTE UE change creating the metric of Uplink Throughput.
17. The PDF of the average Downlink Throughput Measurement. This extension is used to collect the average Downlink Throughput of each LTE device and create a statistical distribution in order to determine the variation and mid value of average throughput for all the system.
18. The CDF of the average Downlink Throughput Measurement. This extension is used to collect the average Downlink Throughput of each LTE device and create the function that gives the probability that an LTE device has to acquire the specific average downlink throughput.

These were the implementation of the Measurement interface that use the measured time stamp every Measurement event in order to generate data for the output of the Simulation.

4.4.3 Abstract class Application

The abstract class application is the programming interface that allows the Application Layer to contain a diversity of applications with different requirements from the network. The application interface forces the applications to follow a certain markovian model of state machine in order to have a realistic stochastic behavior for the simulation.

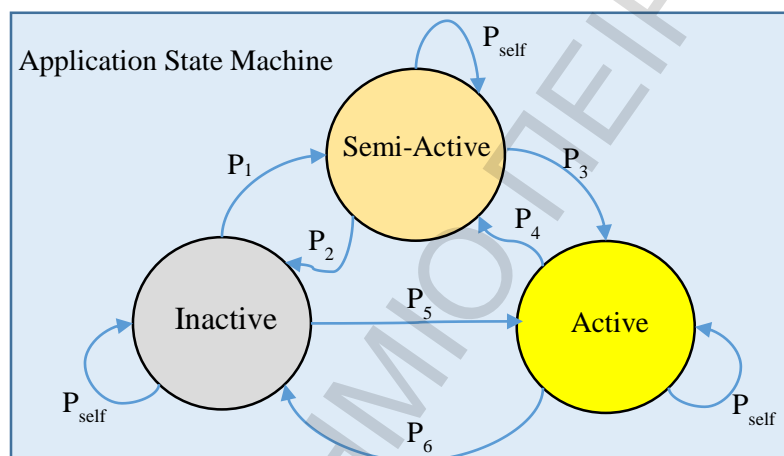


Figure 16 - The Application's state machine

The transition from each state to the other is dictated by a probability that derives from the nature of the application. This probability is calculated using statistical analysis from mobile usage data that derive from various mobile network operators and it gives us a good picture of the behavior pattern of each application type. In the inactive state, an application is not using the network for transmission, therefore it stands in a waiting mode. In the Semi-active state the application is active but not using the full potential of throughput it may request. This means that probably this application was activated previously and now it simply uses the data channels for small update functionalities. The active state is usually the state that comes after the inactive state. Requires the maximum amount of throughput to be served from the lower layers and does not stay active for many epochs. For the time being, the following applications have been developed in order to install them to the simulated UE devices and generate their respective throughput on the Network layer.

- **ANDSF Application.** This application extension simulates the application that instigates the transferring of the ANDSF template from the Network database to the UE in order to be later used by the decision-making mechanism. This application has a simple transition mechanism that activates it once every “Simulator.DAY”.
- **Voice Application.** This application extension simulates the transmission of a voice over LTE service (VoIP) without compression or encoding. It creates data rates that use the Gaussian normal distribution with mean value 32kbps and variation 16kbps symmetrical for uplink and downlink traffic.
- **HTTP application (Web Browsing).** This application extension simulates the user operation of browsing into web pages or other means of communicating over the worldwide web. It uses a deterministic function that changes the amount of bytes requested for download by using a table of values, which come from statistical analysis of traffic in conjunction to the time of day. It does not produce uplink bytes (the HTTP request is considered of very small payload) so it helps us calculate the Downlink overhead that this procedure has to the various radio access technologies.

These are the implementations of the Application abstract class that help us simulate effectively the behavior of the higher level of the LTE UE system module.

4.5 Analysis of Data Structures

This chapter involves commentary on the basic data structures and modules that are being used to simulate the real entities that take place in the simulation procedures. These entities can be entire systems, like the LTE UE module, simple data structures storing information, like the LTE Data Info bundle, or physical entities like real life Objects and the Mobility Stats structure.

4.5.1 The Mobility Stats Data Structure

The mobility stats data type is a very important object in our implementation of the Simulator. It is used to store the information of the physical entity that hosts many virtual modules for the system. By physical information we mean the location, the movement speed the movement model or other information like antenna radius and RAT type. They are stored in a collection within the body of the Main class of the simulation and they are being used by the graphical system engine in order to be viewed

in the form of a “Node” or a visual effect at the simulation playground. This type of separation between the graphics engine and the data structure gives us the freedom to interact with the Mobility stats separately from the graphical rendering procedure and then await for the “Repaint” procedure to give us the information about the change we have conducted. Another important aspect of the mobility stats data structure is that it includes the “Move” function which gets invoked by the Mobility Event in order to simulate the actual movement of an LTE UE node during the Simulation. Below we can see the way the Mobility Stats data structure influences the graphical environment of the simulation.

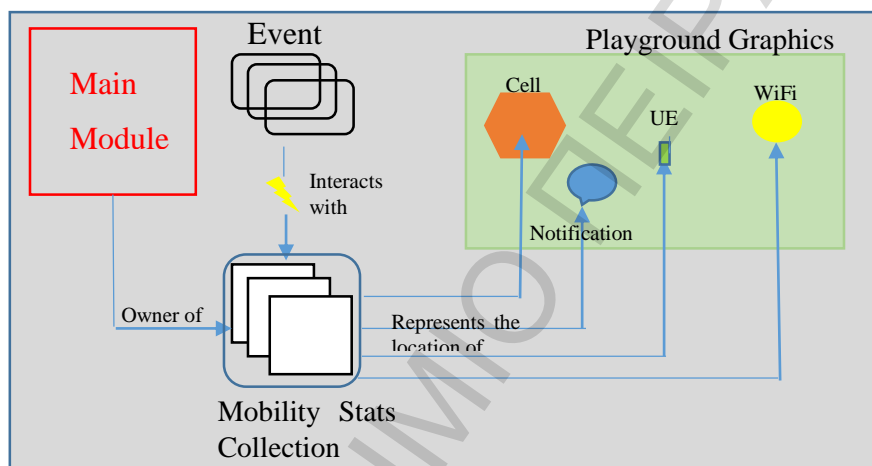


Figure 17 - Analysis of the Mobility Stats Data structure

4.5.2 The Topology Data structure

The Topology Data structure is an Object that describes a set of preferences for a number of Mobility Stats Objects. It is created by the Main class graphical user interface and saved in a created folder by the means of Object serialization. The reason that we have created this data structure is because we wanted to create a mechanism that saves a topology after the changes we have made for it and then gives us the means to import it to the system to be used for other different simulation parameters. If someone wants to compare the impact of a specific parameter in a simulation then the majority of the system needs to perform in an identical way except from the specific parameter.

Below we can see the steps that the Topology data structure mechanism works.

1. Initialize the simulator with specific input parameters.

2. Generate topology using the default generation function (Genetic)
3. Use the GUI options to alter the final result of the topology to the desired topology
4. Press the save topology button given in the GUI. Save the file with a specific name.
5. Restart the simulator and select the “Import topology from file option”.
6. A window prompt will be showing all the saved topology files and will compare the compatibility they have with the selected preferences.
7. Select a topology file and it will be imported to the Program. The results will be shown in the playground.

It is a simple mechanism that helps the usability and repeatability of the Simulator while using simple Serial Objects. Other ways to serialize the topology are the creation of a special .xml file that describes the topology of certain Objects but for this implementation the simple serial object is selected.

The methods that are included in the Topology body are the comparison method “is Match” which checks the selected input parameters and the saved input parameters of the file and generates a Yes or No response (Boolean) and the “Digest Topology” method that fuses the file’s preferences with the active Object collections inside the Main class’s body.

4.5.3 The LTE Information Data Bundle Structure

This Data structure is the envelope or carrier of the information that is being exchanged between the two entities in order to decide for the handover decision. In reality it contains a set of tables that contain information in a primitive form of Integers or Double precision variables that indicate either the SNR received for each RAT in the playground along with the movement speed of a user or the cell load of each surrounding RAT and the Cell type identification. The way we have formulated this procedure is that the same exact data structure is exchanged in either of the decision making cases but different payload is included. This data structure has an internal method to determine the size of the file and the delay that will be caused for its transmission through the selected signaling channel. It contains respective methods for this calculation and is included in the event chain of the decision making function.

Below you can see the importance of this data structure for the decision making functionality.

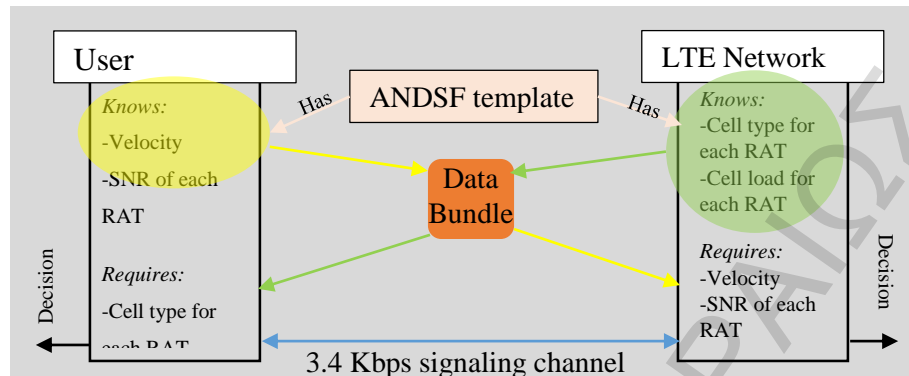


Figure 18 - The Data Bundle's role in the two decision-making cases

4.5.4 The Energy Module data structure

The energy module of the LTE User Equipment module and the Radio Access Technology module is represented by the data structure "Energy Model". This java class contains the information about the capacity of the battery of the LTE phone and the voltage and usage of the power line used by the LTE eNodeB. The class body contains methods to set and get the current of the battery (Ampere of electrical current), method to set and get the upper limit of the capacity in the battery and also methods that resets the data structure to its original state. The most important function in this data structure is the "Resolve" function which is invoked during the life cycle of the LTE UE device (after the triggering of an LTE function Event) and generates a measurement for the time passed altering the context of this data structure. This way we can illustrate the graphical element of the "battery" of the mobile phone according to the usage of the LTE network.

Although a lot of thought is put in this module it remains **under test** for consistency of the output data.

4.5.5 The ANDSF module data structure

As we described in the decision making mechanism in previous chapters, apart from the contextual information of both the underlying network and the specific LTE UE there is need of information about the network operator's selected ANDSF policy

template. The way we have designed this data structure to interact with our simulator is an importing mechanism to translate (or parse) the xml file containing the policy information, a data structure that encloses the information that derived from the ANDSF template and a functional API that lets the LTE UE modules query the ANDSF weight of each device for the decision making procedure. To do this we have created 4 classes, 1 of them being a static class designed as a helper to implement the set of operations required in the runtime and 3 classes that form the structure of the ANDSF template.

- The ANDSF basic entry class. This class is the basic data structure that populates the ANDSF template. It is an entry with four fields containing information about the SSID, the Radio access technology, the priority of the specific entry and also the time zone that this rule applies.
- The ANDSF time entry class. This class encloses a series of ANDSF basic Entry objects in a way that all the basic entries have as their time field that particular time Entry. A time entry has a time entry serial id, a beginning and an end time following the 24hour time cycle.
- The ANDSF template class. The ANDSF template class is one of the most important data structures of the program. It contains an array of ANDSF time entry and basic entry in order to represent the runtime form of the xml document (ANDSF xml). It also works as the API for the acquisition of their information with the “get Max priority” and “get priority” functions. It also gives the ability to export the data in a two-dimensional Object Array in order to use it for the constructor of the JTable graphical element that shows the selected ANDSF template.
- The ANDSF Helper static class. This class is the endpoint of the ANDSF data structure. We use this class in the Main class body in order to parse the xml file and generate ANDSF template instances to store its information. It has the ability to store different ANDSF templates and swap their activation with a single list index and this gives us the opportunity to dynamically change the effective ANDSF template and alter the behavior of the simulator.

This set of classes describes the API of the ANDSF mechanism that we have implemented and is used by the Simulator in the case the ANDSF policy system is activated. The ANDSF templates need to be stored in the ANDSF folder of the programs root directory and to contain valid .xml files with ANDSF information.

4.6 Analysis of Graphical User Interface

This Program is a graphical program which means that it generates graphical elements of input and output in order to interact with its user. It varies from command line programs in the sense that it reserves more resources for its execution but provides with greater visual results, better flexibility and usability. Graphical elements can be divided into a few categories in order to better explain their importance in this program. The categories are:

- Windows used for the input of the desired preferences for the simulation procedure
- Small window dialogs that are supporting the operation of the program
- Custom Graphical canvas that illustrates the context of the heterogeneous network and the changes it undergoes during the simulation
- Functional plots of the output metrics to provide a better visualization and illustration of the results

A different way to separate the categories of each graphical component is to put them in different groups according to the moment a user interacts with them. With this grouping procedure we can see:

Table 1 - Timing Diagram of Graphics

Initialization	Pre-Simulation	During-Simulation	Post-Simulation
<ul style="list-style-type: none"> • Input parameters • Defaults • Export Settings 	<ul style="list-style-type: none"> • Modify topology • Input parameters • Save topology • Zoom in/out • Begin the simulation 	<ul style="list-style-type: none"> • Modify topology(not recommended) • Change Input parameters (not recommended) • Pause/Stop/Speed simulation • Enable graphical effects for the simulation 	<ul style="list-style-type: none"> • Run new simulation • View/Save results • Change input parameter • Change network topology

The following chapters will analyze each graphical user interface of the program in order to understand its functionalities and limitations as well as the impact it has on the simulation program.

4.6.1 The Initialization window

After someone executes the Simulator, the first graphical component he will face is the initialization window. This window contains multiple tabs that contain all the possible input parameters that can create the numerous different simulation scenarios. This window consists of different Java graphical components of the Swing library that influence the local variables of the Main class which afterwards initializes the operational modules in the preferred manner.

Another important element of the input graphical interface is the Constants class. The Constants class of the Helper package is a very large collection of captions and values that are referenced from the graphical components in order to create the Labels, the Buttons, the JComboBoxs, the JLists and all the other Components. The correlation between those factors is:

- Local Variable <-> Graphical Component <-> Data List from Constants
- If the local variable is of a numeric type (Integer, Float, Double) then the representation of the graphical input component is a JTextArea in the case the user has complete freedom of input or a JComboBox (or drop list) in the case the user has a specific set of parameters he needs to choose from. If that's the case then the Constants class is used to create the predefined input parameters.
- If the local variable is of a type String then the representation of the graphical input component is a JTextArea or a JComboBox linked to a predefined dataset inside the Constants class.
- If the local variable is of the type Boolean then the representation of the graphical input is a JCheckBox.
- If the local variable is of the type Date then a custom made JComponent is used for the input parameters.
- Other types of graphical input elements are JSliders (drag-able) and JSpinners

The different groups of options is separated with the JTabbedPane logic meaning that different Java tabs are used for different families of input parameters.

This window has 3 buttons that lets you interact with the sum of the options in all its tabs.

1. The apply button. This button transits the simulator to the next operational state
2. The defaults button. This button puts the default parameters in each of the simulator's field according to pre-set hard-coded data.
3. The export button (**under test**). This button uses a custom made serialization mechanism that uses the knowledge representation language XML in correlation with the Java Reflection API in order to take a snapshot of the simulator's input parameters to create scenario inputs for multiple, scripted executions.

Below you can see a thorough analysis of the initialization window and its components:

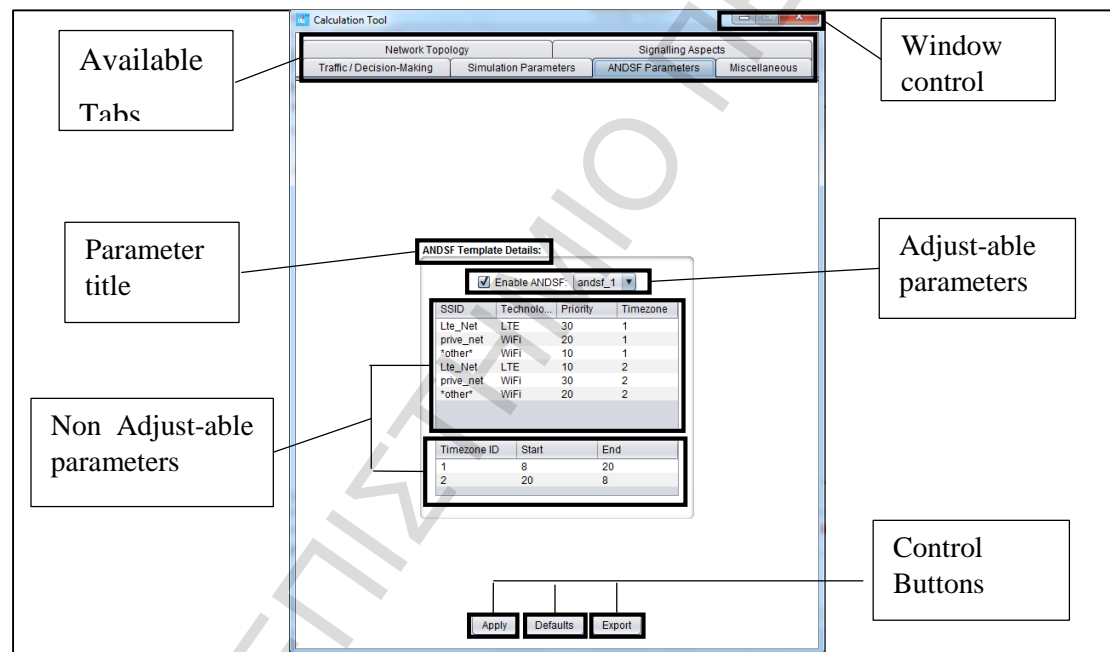


Figure 19 - Analysis of the initialization GUI

The apply button leads the program to its next stage which is the pre-simulation step. Analysis of the graphical elements of this step will be found in the next sub-chapter.

4.6.2 Analysis of the Playground graphical elements

After the initialization stage, where the user inputs his preferred parameters for the simulation, the graphical monitor of the simulator along with the command buttons and capabilities is created. The topology of the network is created according to the user's preferences (pre-constructed or generated using a genetic algorithm). Afterwards there

are some of the most important user interface additions that allow for micro-adjustments of the underlying topology in order to cover all the possible variations of the imported technology.

The capabilities of the main screen are:

- The zooming in /out in the playground's point of view
- The selection of one, multiples (by using Ctrl) or area with multiples in order to move them in a specified location.
- Their movement is implemented by either using the directional buttons of the keyboard or by pressing the right click on the mouse which generates a movement thread and an arrow showing the direction of movement.
- Escape button clears the selected objects, F1 swaps the theme of the simulator's view, F2 deactivates the refresh of the graphics (increases performance), F3 deactivates the update of the simulator's control panel (increases performance), F2 deactivates the refresh of the graphics (increases performance), F3 deactivates the update of the simulator's control panel (increases performance), F4 shows or hides the labels of each object in the playground and F5 increases/decreases the quality of the Java graphics. Finally the space button, during the simulation, pauses the screen giving it an effective blur filter and the "Paused" indication.
- The control panel has commands that influences the simulator's operation. The play button switches to stop after pressed and switches back to play after stop or the end of the simulation's duration. There is also the reset button which reset's the topology of the network, the save topology button that creates the .top extension file containing the serialized Topology Object (previous chapters). Afterwards there is the "STOP ACTION" button which only is clickable if a movement action is initiated through the user interface , the progress bar of the simulation that shows an estimate about the duration of the simulation (time left) and the Slider of the simulators speed. The option to adjust the simulator's speed is because fast simulations may not give you a precise picture about the contextual changes that undergo. Below these buttons there are some checkboxes that alter the graphical layout such as the energy checkbox, which shows the energy layer's battery state on each UE and also the grid checkbox that displays a 10x10 grid overlay on the simulator's playground.

Finally there are some indications concerning the location of the cursor inside the screen and the amount of selected nodes in the playground and also a drop list object that helps u select a specific object. One final feature is the “Lamp” button which enables or disables the pop-up notification messages of the simulation.

All these graphical elements are contained within the specially formed java class “Topology Panel” which is an extended version of a Java JPanel Object. Below we will analyze the programming principles that have been used in order to achieve all these complex tasks.

Topology Panel class Analysis

Topology Panel is a java class that extended the JPanel draw-able object to a specially made graphical context object for our simulator. The “paint component” method has been overridden in order to use the collection of Mobility Stats found in the Main class and paint them in their precise location using a series of buffered images and real time painted graphics. In this method we also draw the additional (side) components which are the control panel, the Label with the title and the zoom slider. In order to use the keyboard input and all the mouse functionalities described earlier, this class also implements the mouse listener, the mouse motion listener, the key listener, the action listener, the document listener, the change and the mouse wheel listener interfaces. This maps the events thrown by the Event Queue with actions that change the behavior of objects inside this compound class. Another key feature of the Topology Panel class is that it has its own thread that repaints the image (different than the window it is being contained). This is implemented with the usage of a Swing Timer which has the ability to repeat a certain action (the repaint) for this class. The rest of the function of this class is the iteration of the Mobility Stats object and invoke a different paint method for different types of objects. Methods for drawing exist for Cloud Messages, eNodeB, pico Cell, WiFi AP, and LTE UE and also for the displaying of motion events (created manually) and ongoing EPS bearers (radio traffic). The selected mobility stats by the selection option use the same system in order to paint the green frame around them. Also the area –based selection, although it is a method implemented in the Main class body, it uses mobility stats and it is being painted by the Topology Panel. Finally the

zoom in/out and the pause procedures are executed in this class by either scaling the graphics or by producing the blurred image and overlaying it.

This implementation is a pretty straight-forward graphics implementation that uses the CPU-accelerated java graphics in order to display the simulation's context. Other methods including OpenGL or OpenCL frameworks could also work but the plan of this system's design is to be formed by a single programming framework.

Below we can see the graphical view of the main screen with analysis of its components:

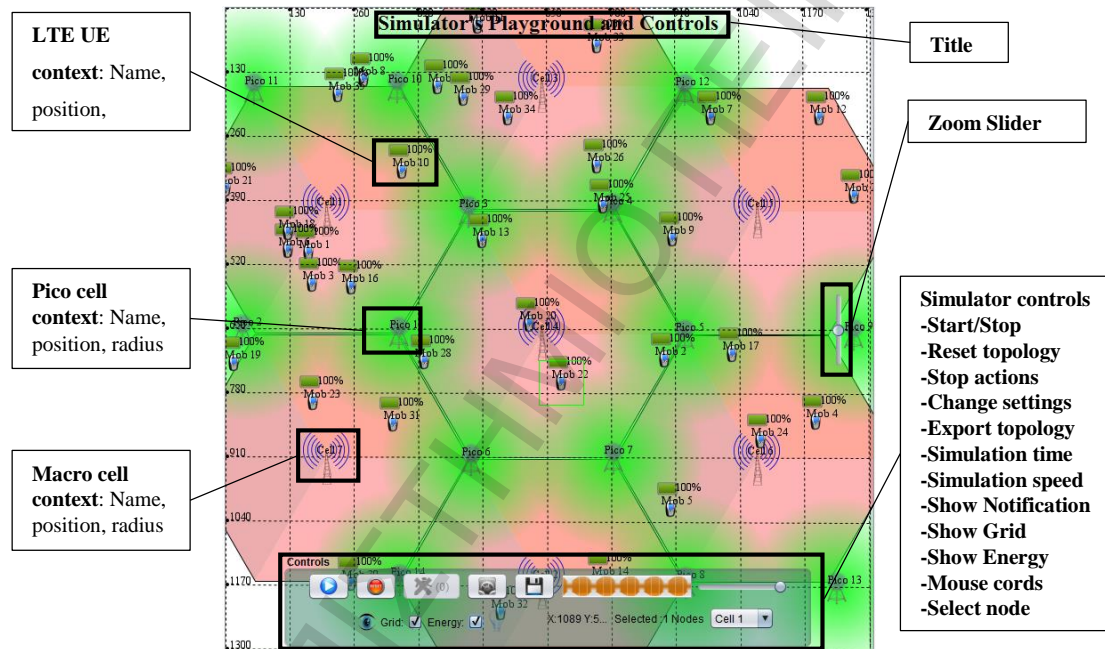


Figure 20 - Analysis of the main graphical view

After a simulation is finished, according to the user's selected output, graphical plots are being created and put on the right side of the screen. The output graphics will be covered on the next chapter.

4.6.3 Analysis of the Output graphical elements

Every simulation generates a series of output data that need to be displayed properly to the user of the Simulation tool in order to fulfill the research purposes of its execution. Most of the output data are displayed in the right panel of the simulator's main window and are separated by different tabs for different simulation instances. There are some special UI abilities that are built in the plot system but the engine that creates them is

the JFreeChart API which we used in order to avoid “Reinventing the wheel” in Java plots. JFreechart gives an easy to use API to use as input any kind of data forms and generate potent illustrations with many automatic features like zoom in / out and data selection. Even so, some additions to the JFreeChart functionalities were made in order to better utilize the graphical output. Every instance of the Measurement class that has been implemented and activated in the simulator has an implementation of the abstract method “Generate Plot” which returns an object of the JPanel type containing the plot. These objects are then placed inside the tabbed pane using a special java layout the Wrap Layout. The plots are big in resolution so vertical scrolling policy is implemented with the usage of the Scroll Pane Container. Apart from visual the output section of the simulator has a few interface capabilities:

- The delete button, which clears the memory for all the graphical data of the simulation.
- The export images button, which generates a folder containing an image of each graphical result for the simulation.
- The meta-adjustments plot dialog which gives the user additional capabilities to alter the JFreeChart plots.

All these features together, along with the generated Microsoft Excel sheets with all the data form a great representation of the simulator’s output that looks well defined and efficient in the terms of memory usage. Below we see the analysis of the output panel:

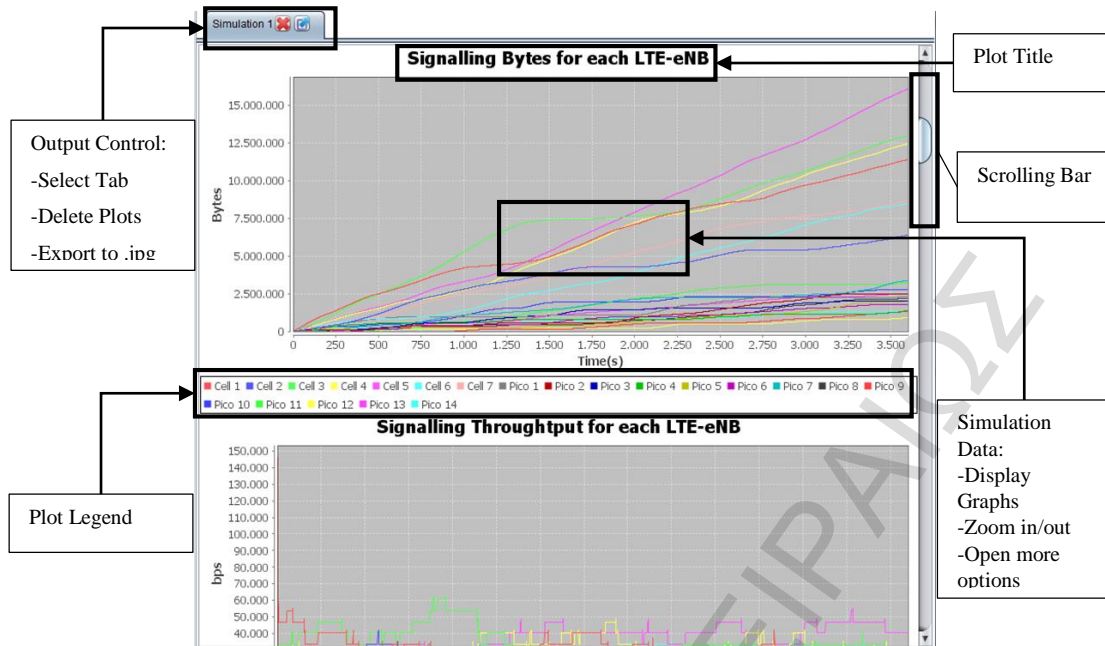


Figure 21 - The graphical elements of simulation output

4.6.4 Analysis of the various system dialogs

Apart from the main window which shows the initialization options, the simulation playground and also the output data of the simulation, this program has a set of specially made dialogs in order to interact with the user in a custom way. These dialogs appear mostly after the pressing of a button on the UI and we will describe them below.

Select Topology Dialog

This dialog is enabled at the creation of the system module after pressing the apply button in the initialization procedure. The dialog follows the user's preference to load the topology of the network from a predefined file and searches the "topology" folder of the working directory for serialized objects of the Topology type. Then it creates a JTable element for the user to select the wanted topology file. The JTable element has selectable rows so the selection process is done by the mouse or the directional keys. The table shows many information about the topology files such as the file name, the playground size, the number of macro cells, pico cells and wifi AP's and the compatibility of the file with the preselected parameters. Below we see the dialog window:

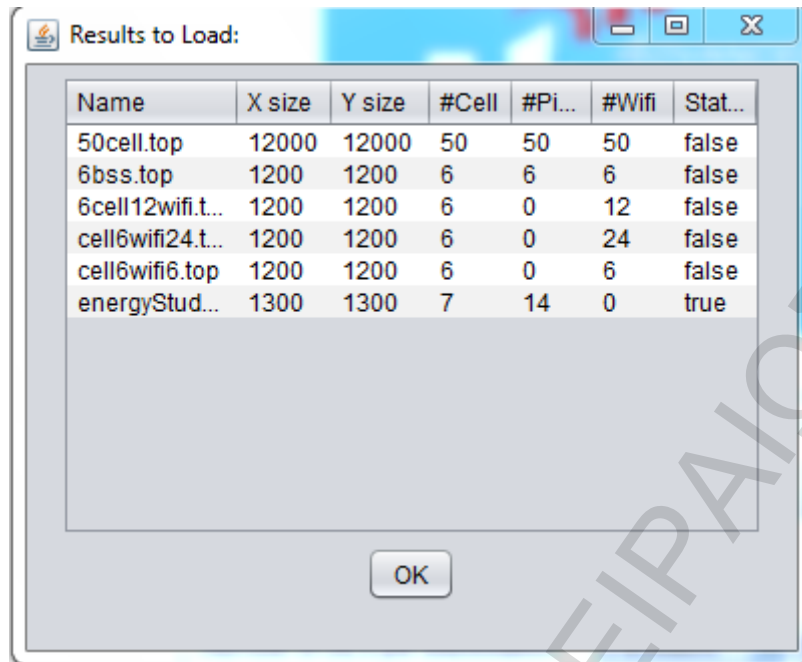


Figure 22 - Dialog for the selection of the imported topology file

The runtime simulator settings dialog

In the case that the user of the simulator wants to change an operation parameter of this simulation during the runtime, he can pause the simulator and press the “Settings” button in order to open a new window with the simulator’s parameters. Note that while all the settings from the initialization window appear, only those that do not occur on the initialization of the modules can affect the running simulation. The selection of the category of parameters that a user wishes to alter is done with a JList graphical object. Below we see the runtime simulator settings dialog.

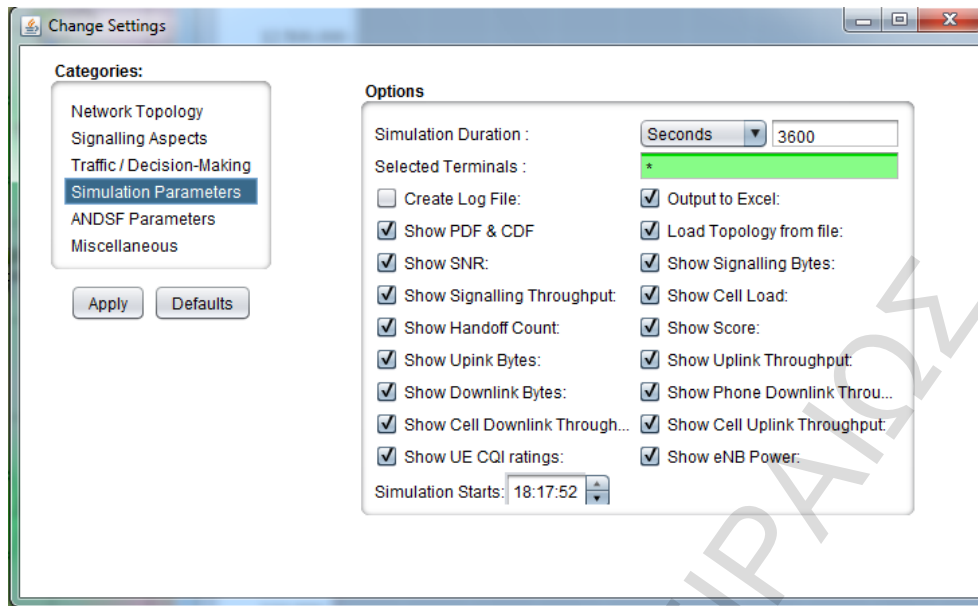


Figure 23 - The runtime simulator settings dialog

The JFreeChart customization dialog

JFreeChart is the external API we have used in this program in order to plot the output data of the simulation in multifunctional potent Java panels. Although JFreeChart gives us a plethora of embedded customization options (using the right-click option menu), we needed to create a set of our own custom preferences in order to add some additional features. These features are the selection of a specific XY series to display, the grouping of the XY series with a same property such as same RAT and the disabling of the Plot's Legend. Below we see the custom made dialog for this operation:

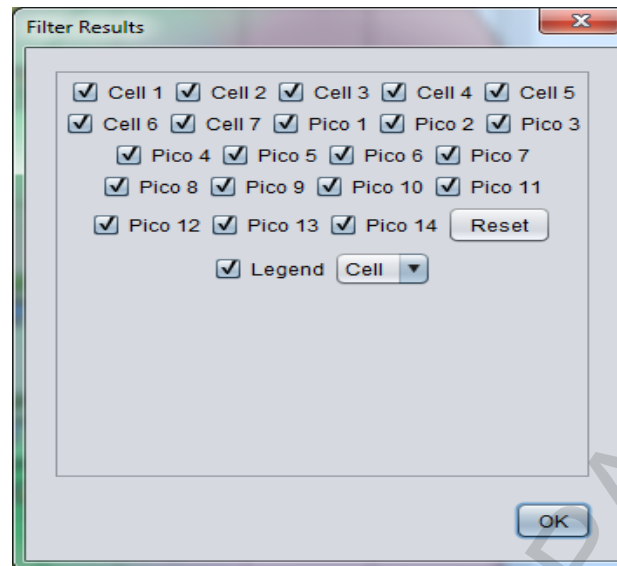


Figure 24 - The JFreeChart additional options dialog

The Signaling procedure customization dialog

In the initialization procedure of the simulator, there is an input section about the signaling procedures we have analyzed in the LTE overview chapter. There we have hardcoded the estimated parameters of the payload size and delay for each of the following procedures:

- LTE attachment procedure
- LTE call initiation procedure
- LTE call release procedure
- Intra-LTE handover procedure
- RRC idle state periodic exchange

In the signaling section of the initialization window, next to each of these parameters is an options button which prompts the user to alter the estimation of the payload size of one or all of the parameters that form the general procedure. The dialog shows a list of sub-operations and the signaling bytes they need to be created. Below we see an example of the signaling procedure customization dialog:

Step	Count
1. RRC Connexion Request	24
2. RRC Connection Setup	15
3. RRC Conn Setup Complete	27
4. Attach Request	8
5. PDN Conenctivity Request	4
6. Create Session Request	358
7. Create Session Response	171
8. Initial Context Setup Request	68
9. RRC connection Reconfiguration	8
10. RRC Reconfiguration Complete	7
11. Initial Context Setup Response	19
12. Modify Bearer Request	204
13. Modify Bearer Response	134
14. Attach Accept	61
15. Activate Default EPS Bearer Req	71
16. Attach Complete	7
17. Activate Defaults EPS Bearer accept	3

Figure 25 - The signaling procedure customization dialog

4.7 Analysis of the Multi-Threaded Environment

The simulator software we have developed has many dynamic elements that run in the background or the foreground. For such a task to be implemented, a multi-thread architecture has been chosen by the Thread API of Java programming language. In general, a program with a graphical user interface has more than one threads. This happens because the functions that are required by the graphical classes are needed to be repeatedly invoked in order to keep the user constantly informed about the current status of the graphical elements. Apart from this thread, the paint thread, for this implementation we have developed a set of threads that perform other, parallel tasks in order to create a more potent software. Below we enumerate the running threads of the software:

1. **The Main thread.** Every program has a main thread and it begins at the static main function of the Main class. This thread begins the initialization of the program and generates the rest of the threads in order to create the multi-thread environment.

2. **The Initialization window graphics thread.** Upon the generation of the window of the simulation, the main thread and the graphics thread get separated in order to cover the basis explained in the prologue of this chapter.
3. **The repaint thread of the special graphical canvas.** This thread is an implementation of the “Swing.Timer” class that allows the setting of a repetition interval between consecutive tasks. The task it produces is the repaint function of the graphical element JPanel we have extended to the Topology Panel class.
4. **The update UI Thread.** This special thread is created for the sole purpose of updating the progress bar, the buttons and all the graphics elements that change during the simulation.
5. **The simulation-dedicated Thread.** This thread is the Thread that hosts the Simulator start method. By separating this operation from the main thread, we can set the JVM thread priority to high and focus all the processing on it in order to increase performance.
6. **The Notifications Thread.** This thread is responsible for handling the fade duration of the messages that appear as notifications at the simulator’s playground. This procedure is painted by the repaint thread, but the alpha value of the colors and their removal is handled by the notification thread.
7. **The User-Initiated movement Thread.** As we have mentioned in the graphical user interface capabilities of the program, in the pre-simulation stage the user has the ability to order some nodes to perform a motion in order to set the substrate network to the wanted topology. This operation is performed by the movement thread, which changes the location of the mobility stats elements gradually and allows for the display of an arrow pointing to the target location.

The threads along with the rest of the non-visible threads that occur during runtime consist the parallel operation that gives the simulator its effective graphical performance. Below we can see a diagram that displays each thread and its corresponding functionality.

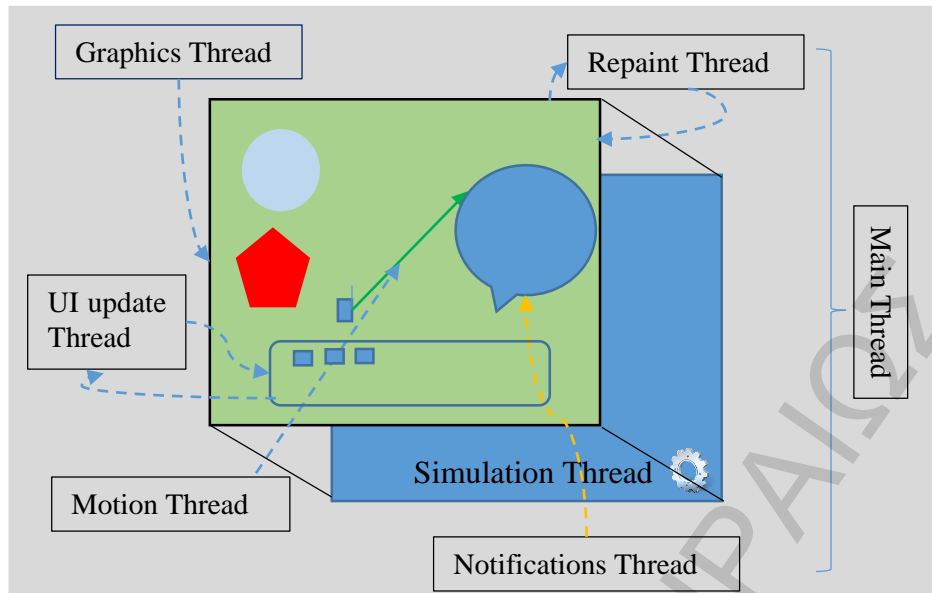


Figure 26 - The thread diagram

4.8 Flow Diagram of the Program

After having analyzed the simulation engine, the graphical components, the data structures and the multi-threaded environment of the software, this chapter is dedicated in the analysis of the high level flow chapter of the simulator in order to summarize the functionalities and present them in a linear way. This diagram provides information about the use scenarios of the software and the logical links between the operational blocks of the system.

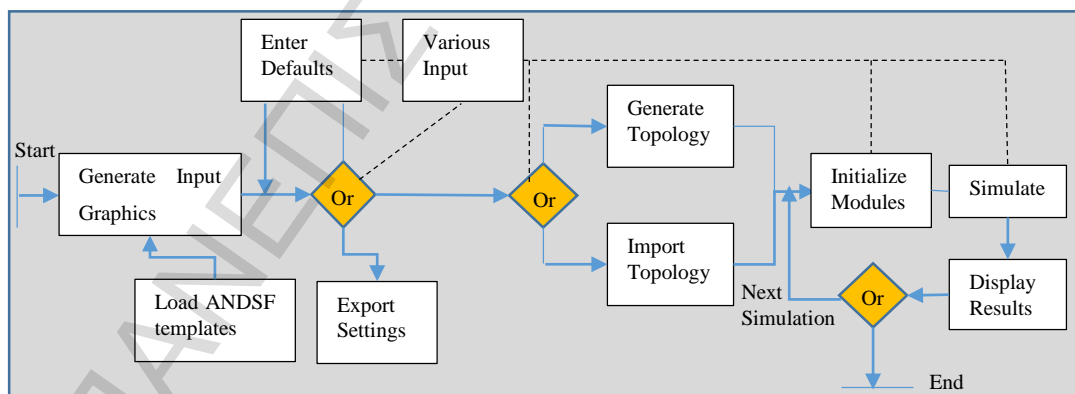


Figure 27 - Flow Diagram of the Software

4.9 External API Analysis

The program we have developed is a collection of software modules we have developed specifically for the sake of this research. Having said that in order to increase the

productivity of the development procedure, a set of external API's and class implementations have been used with the permission / license of their respective owners. These code resources are used in different sections of the operation of the program and in this chapter we will enumerate and analyze them, explaining their importance in the development of the program.

4.9.1 The JFreeChart API

JFreeChart is a very popular java library that helps visualize any type of mathematical and logical data in a series of colorful and potent graphical Plots that integrates with most of Java programming language's graphical user interface API's. In this implementation we have used the JFreeChart library in order to generate custom-set JPanel objects, the Chart Panels containing the measured datasets and the meta-calculated results of the simulation.

To use this library we have created the Plot class, which takes as input the labels, the implementation of the Measureable class (see abstract class section of the chapter) and other preferences which then generates a Chart Panel Object that illustrates the data. The Chart Panel Object is then integrated to the program's window.

The JFreechart project's URL: <http://www.jfree.org/jfreechart/>

Example of the JFreeChart API's usage can be found in the screenshot below:

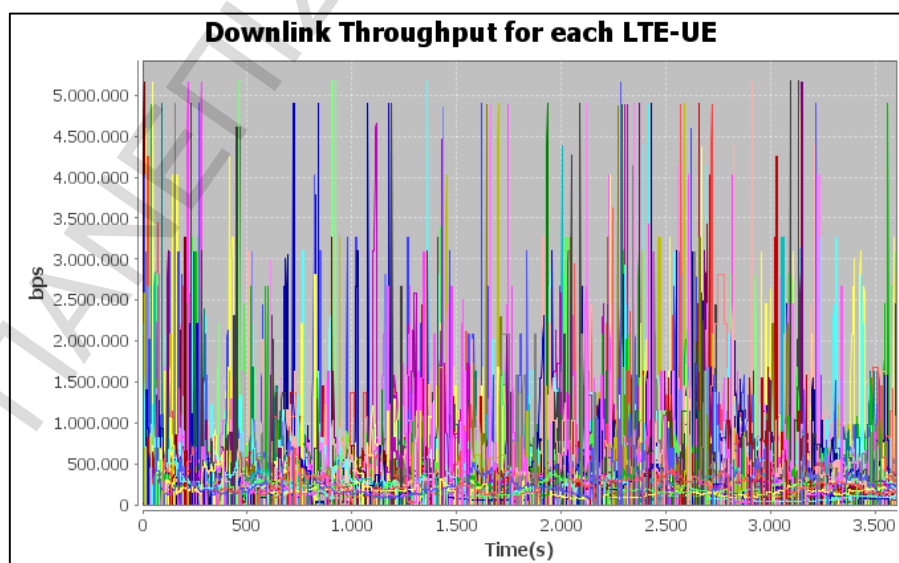


Figure 28 - Example of a JFreeChart plot

4.9.2 The Apache POI library

The apache POI library of the apache foundation is a very potent API that allows, among other, the translation of Java data structures into Microsoft Excel documents with customized tables and properties. For this precise reason we have used this library because we wanted to give the ability, having a user selected it, to generate excel files with all the output data and then perform meta-calculations like average, variation and probability density analysis. The apache POI license is apache-type therefore it is used in this non-commercial program for academic purposes.

The apache POI Project's URL: <http://poi.apache.org/>

Below you can see an example of the output documents in the excel extension generated in a simulation scenario:

	A	B	C	D	E	F	G	H	I	J	K
1	Time	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Pico 1	Pico 2	Pico 3	Pico 4
2	0	10,37189	9,521913	13,19581	6,398631	8,110757	8,48866	-19,2479	-23,5346	-22,5269	-25,0001
3	1	10,37189	9,521913	13,19581	6,398631	8,110757	8,48866	-19,2479	-23,5346	-22,5269	-25,0001
4	2	10,38542	9,532367	13,20662	6,406638	8,109183	8,483434	-19,2465	-23,5296	-22,5373	-25,0043
5	3	10,38542	9,532367	13,20662	6,406638	8,109183	8,483434	-19,2465	-23,5296	-22,5373	-25,0043
6	4	10,39836	9,542646	13,21761	6,414593	8,106859	8,478159	-19,2453	-23,5247	-22,5477	-25,0086
7	5	10,39836	9,542646	13,21761	6,414593	8,106859	8,478159	-19,2453	-23,5247	-22,5477	-25,0086
8	6	10,41059	9,552568	13,2283	6,421917	8,103826	8,472512	-19,2455	-23,5199	-22,5584	-25,0132
9	7	10,41059	9,552568	13,2283	6,421917	8,103826	8,472512	-19,2455	-23,5199	-22,5584	-25,0132
10	8	10,43366	9,562578	13,2395	6,429276	8,100357	8,467769	-19,2454	-23,5158	-22,5684	-25,0165
11	9	10,43366	9,562578	13,2395	6,429276	8,100357	8,467769	-19,2454	-23,5158	-22,5684	-25,0165
12	10	10,4574	9,572432	13,2509	6,43601	8,096429	8,462998	-19,2455	-23,5117	-22,5785	-25,0199
13	11	10,4574	9,572432	13,2509	6,43601	8,096429	8,462998	-19,2455	-23,5117	-22,5785	-25,0199
14	12	10,48242	9,582578	13,26324	6,442322	8,092534	8,459681	-19,245	-23,5072	-22,5876	-25,0231
15	13	10,48242	9,582578	13,26324	6,442322	8,092534	8,459681	-19,245	-23,5072	-22,5876	-25,0231
16	14	10,49761	9,593234	13,27678	6,448504	8,089156	8,455759	-19,2429	-23,5015	-22,5836	-25,0266
17	15	10,49761	9,593234	13,27678	6,448504	8,089156	8,455759	-19,2429	-23,5015	-22,5836	-25,0266
18	16	10,50147	9,603419	13,29051	6,453427	8,085437	8,452241	-19,2409	-23,4964	-22,5794	-25,0296
19	17	10,50147	9,603419	13,29051	6,453427	8,085437	8,452241	-19,2409	-23,4964	-22,5794	-25,0296
20	18	10,5051	9,615945	13,30506	6,45715	8,081868	8,449166	-19,2387	-23,4907	-22,5748	-25,0323
21	19	10,5051	9,615945	13,30506	6,45715	8,081868	8,449166	-19,2387	-23,4907	-22,5748	-25,0323
22	20	10,50742	9,626474	13,31953	6,459194	8,078983	8,445838	-19,2368	-23,4858	-22,5702	-25,0354
23	21	10,50742	9,626474	13,31953	6,459194	8,078983	8,445838	-19,2368	-23,4858	-22,5702	-25,0354

Figure 29 - Example of the Apache POI API-generated Excel file

4.9.3 The DOM xml parser library

In previous chapters we have shown that, both in the implementation of the ANDSF Policy template and also for the custom-made serialization of the input data of the simulator, there is need for a fast and accurate Java library for the input and output of files of the xml extension. Therefore we have used this the Java DOM library which provides very easy to use xml reading functionalities in order to create function for the custom made special documents of the program. The Java Runtime environment

(release 7) has included this library in the default class set so the reference to the library jar is deprecated (December 2013).

The DOM project's URL:

<http://www.java2s.com/Code/Jar/w/Downloadw3cdomjar.htm>

Example of the generated data structure: filename – defaults_1.xml

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <settings>
3 <variable>
4 <class>int</class>
5 <name>$_numberOfCells</name>
6 <value>0</value>
7 </variable>
8 <variable>
9 <class>int</class>
10 <name>$_numberOfMobile</name>
11 <value>30</value>
12 </variable>
13 <variable>
14 <class>int</class>
15 <name>$_radius</name>
16 <value>100</value>
17 </variable>
18 <variable>
19 <class>int</class>
20 <name>$_picoradius</name>
21 <value>0</value>
22 </variable>

```

Figure 30 - Example of the generated xml file from the DOM library

4.9.4 Other External Classes

Apart from the imported external libraries we have analyzed beforehand, in the program we have used source code from online tutorials with respect to their authors in order to complete requirements of the program in a more efficient way. In this chapter we will analyze these Classes and give credit to their respective creators.

The Wrap Layout class

Although the Swing library provided by the Java Runtime Environment is a complete set of resources to create a huge plethora of graphical user interfaces, for the purposes of this software we required a special implementation of the Layout Manager implementation that has the special treat of specifying only the width dimension as a limitation for a “Flow Layout”-like Layout manager. This requirement could not be fulfilled by any of the settings provided by the Flow Layout’s methods so we used the Wrap Layout custom java layout manager.

Source code URL: <http://tips4java.wordpress.com/2008/11/06/wrap-layout/>

The Gaussian Filter class

During the simulation, the program has the ability to respond to the stroke of “SPACE” button in a way that pauses the simulation and displays a blurring effect combined with the indication “Paused”. After search in the internet we concluded that we needed an application of a Graphical filter called a gaussian blur filter. This filter takes as input a Buffered Image we wish to process and exports a new instance of the Buffered Image class with blurred pixels of the same color /shape.

Source code URL: <http://www.jhllabs.com/ip/GaussianFilter.java>

Example of the effect produced by the Gaussian filter Procedure:

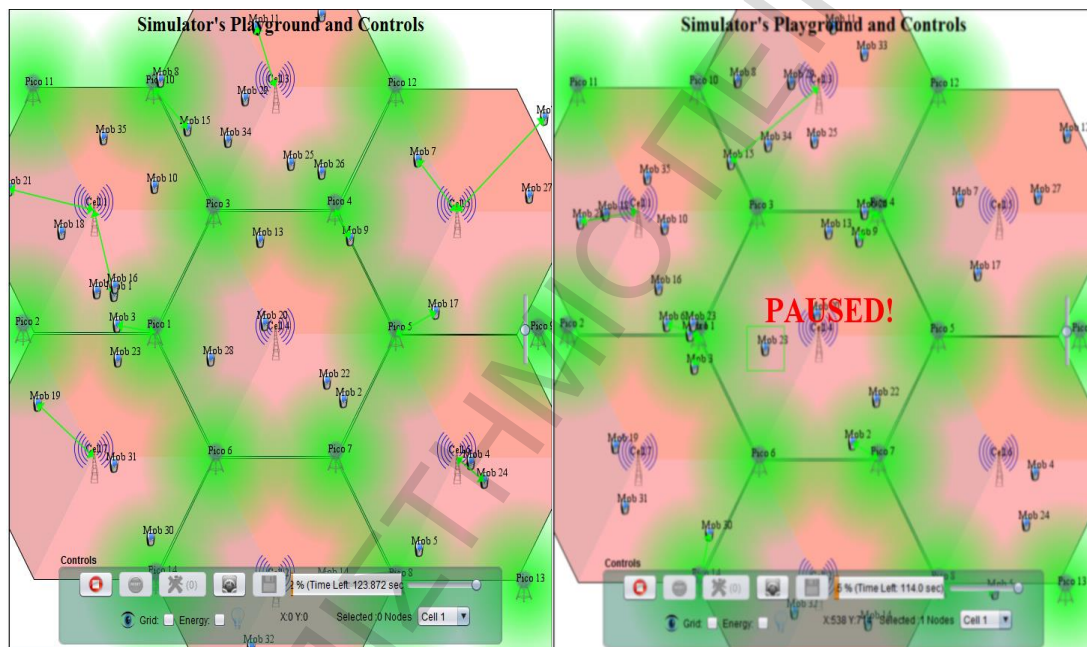


Figure 31 - Example of the Gaussian Blur transformation

4.10 Analysis of the Simulation Input and Output

4.10.1 Input Parameters

The simulator software has a large collections of adjustable parameters in order to generate as many different scenarios as possible and also to study the impact of these changes in various network deployments. These options are selectable from either the initialization input window, where a user sets all his preferences for the simulation or during the simulation with the runtime options provided from the GUI control panel.

Although the options provided during runtime are many, it must be noted that some of them only apply in the pre-initialization state and therefore require system restart.

The input parameters are divided into different tabs / categories in order to browse them more easily and to display them in a grouped way. Below we will enumerate the categories and analyze its corresponding graphical element.

Network Topology

This category is the first options group of the simulator. It displays information about the geographical context, the number of nodes of each RAT and the mobility parameters of the User Equipment. It also displays the Path-Loss model that we use in order to determine the signal strength of the simulation. In detail the parameters are:

Table 2 - Table of the simulator's Network input parameters

Playground side (meters)	The side of the square area that will be created to serve as the simulation's playground.
Area Type	This determines the Inter-Site distance between the LTE macro eNodeBs and therefore it is used to determine the number of macro cells for the simulation. The enumeration is {Dense Urban ,Urban and Rural}
Number of Cells	This field is auto-generated according to the selection on the previous two parameters.
Number of User Equipment	This field has two option fields. The first lets you select the number of User Equipment wanted for the simulation and the second gives you the option to apply this number for each of the Macro eNodeBs.
Number of Pico Cell per Macro cell	This field has the values of {0, 1, 2, and 4} for the selection of the density in the pico cell deployment for the playground.

	There is also the option of adjusting the pico cell's coverage radius.
Number of WiFi AP's per Macro cell	This field has the values of {0, 1, 2 and 4} for the selection of the density in the private (operator-deployed) WiFi AP's that will be used to off-load the 3GPP network. There is also the option of adjusting the WiFi AP's coverage radius
User Equipment's Movement Speed	This parameter helps create scenarios of User Equipment nodes with different movement speed. The values of this parameter are {0(static),3, 30,60 kmh}
User Equipment's Mobility Model	This parameter in conjunction with the previous parameter dictates the motion behavior of the UE nodes. There are four options of mobility {static, linear, random indoor and random waypoint} each providing with an entirely different behavior from the LTE network.
Information about the Path Loss Models	The table at the bottom of the input screen has no input capabilities. It simply displays the path loss model used for each of the different Radio Access Technology used.

Signaling Aspects

This input parameter section is dedicated to the signaling study we conducted at the beginning of this research. It displays every LTE procedure we have analyzed with its corresponding total signaling payload size and delay of the procedure. These fields can be manually adjusted in order to change the output data of signaling for the eNodeB's.

Table 3 - Table of the Simulator's signalling input parameters

User Equipment Attachment procedure	Contains the payload size and the delay in ms of the Attachment procedure that occurs in the initialization of the UE and also at the cell reselection during the idle state of the mobile phone. There is also the option to adjust every subfield of the procedure by the procedure customization dialog.
User Equipment Data call initiation	Contains the payload size and the delay in ms of the Data Call initiation procedure (the reservation of an EPS bearer) by the UE. There is also the option to adjust every subfield of the procedure by the procedure customization dialog.
User Equipment Data call release	Contains the payload size and the delay in ms of the Data Call Release procedure (the release of the reserved resource by the EPS bearer) by the UE. There is also the option to adjust every subfield of the procedure by the procedure customization dialog.
Intra-LTE handover procedure	Contains the payload size and the delay in ms of the Intra-LTE handover procedure (the change of a serving RAT during the connected state) of the UE. Also there is the option to adjust every subfield of the procedure by the procedure customization dialog.
RRC Idle-state data exchange	This Fields contains the payload that repeatedly gets exchanged between the UE and the eNodeB during the IDLE and

	CONNECTED states and also the interval of the procedure.
--	--

Traffic / Decision-Making parameters

In this input options section, parameters concerning the behavior of the application layer of the Simulator and the decision-making procedure are displayed. Also this section has the option to generate a simulation with either the UE or the Network-initiated decision making mechanism. Below we see each parameter separately.

Table 4 - Table of the simulator's traffic/decision-making parameters

Voice Call initiations / day / User Equipment	This parameter is the frequency parameter in which voice calls are generated by each LTE UE of the simulator.
Voice Call duration	This parameter is the average duration of a voice call conducted in the simulator (seconds).
Data Session Call initiations /day / km ²	This parameter dictates the frequency of data calls made by Application in the UE's of the Simulator.
Data Session Average Payload Size /call	This parameter shows the size (in Mbit's) of the Layer 2 traffic generated by the data sessions of the Simulator.
The delay for the processing of the Decision procedure in the User Equipment	This parameter changes the behavior of the system in the case of the UE-initiated decision-making paradigm. The input value is in ms and generates additional delay for the decision-making event chain
The transmission delay for the User Equipment initiated decision making procedure	Additional delay caused by the transmission of information required by the User Equipment to perform the decision-making algorithm.

The delay for the processing of the Decision procedure in the Network	This parameter changes the behavior of the system in the case of the Network-initiated decision-making paradigm. The input value is in ms and generates additional delay for the decision-making event chain.
The transmission delay for the Network initiated decision making procedure	Additional delay caused by the transmission of information required by the Network to perform the decision-making algorithm.
Intelligence Mode	This selection decides whether the decision-making procedure will be executed in the UE or the Network of the Simulator (KEY PARAMETER)
Decision-Making Interval	This parameter is one of the latest additions of the Simulator. Since the decision making procedure is not a triggered but rather a periodic event, this parameter adjusts this event's period.

Simulation Parameters

In this section we see many options that revolve around the actual operation of this software. This means that we can adjust the output data that it will present after the simulation, the id of the User Equipment nodes that we wish to take part in the measurement procedure and other various options such as the Simulation actual time, the simulation duration and the creation of an Excel file with the results.

Table 5 - Table of the Simulator's input parameters for the simulation

Simulation Duration	One of the most important parameters of this software is the simulation duration. This duration is in simulated seconds and there are additional options to adjust the
---------------------	--

	units in which the simulation duration will be input. (KEY PARAMETER)
Selected Terminals	This field is a very important input section of the simulator because it uses expressions in order to determine the set of id's that we wish to monitor their measureable properties for the output data. The "*" wildcard dictates that ALL the User Equipment will be measured therefore some parameters will be generated by using the average function. Other expressions are the comma "," and the high fen "-" that generate subsets of the asterisk wildcard.
Create Log File	This option is responsible for binding the Java Runtime System out data stream to a file output stream meaning that all the debug messages generated will be redirected to a log file for debug purposes.
Output to Excel	This option is executed after the simulation procedure and generates all the excel sheets from the selected output data into an excel file (see measureable abstract class).
Multiple Output data selection	For each measurable parameter of the system, a corresponding selector exists in order to select which of the output parameters you wish to be measured. For more information see the measureable abstract class.
Simulation Actual time selection	The simulation Actual time option puts the parameter of the Starting simulated

	<p>time of the simulator. Because some events of the simulation such as the change in the traffic volume and also the change in the ANDSF policies is a function that has different values for different time of day, therefore we have implemented this function to create different scenarios based on that.</p>
--	--

ANDSF parameters

In this section of the Simulator’s Input parameters, we have implemented a flexible UI to select, display, and import ANDSF templates of the .xml format that we created for the runtime operations of the Simulation. The graphical elements are flexible and simple in an understandable and potent way.

Table 6 - Table of the Simulator's ANDSF input parameters

<p>Enable ANDSF</p>	<p>This is the key parameter of the ANDSF function. Having this deactivated means that the system will not exchange the ANDSF template between users and also it will not be used in the decision-making Objective function. (KEY PARAMETER)</p>
<p>Selected ANDSF template</p>	<p>This parameter comes from the list of andsf files that are found in the andsf folder of the working directory of the software. By changing the selected template, you alter the behavior of the policy system implemented by the operator and also you change the displayed information below.</p>

ANDSF information about different RAT's display	This section displays the selected ANDSF template's data that is relative to each different Radio Access Technology.
ANDSF information about different Time Zones	This section displays the selected ANDSF template's data that is relative to each different Time zone it involves.

Miscellaneous Input Options

This section of input contains the collection of the leftover parameters of the system. This parameters influence the physical and the data link layer of the LTE network and also some other key parameters of the simulation such as the measurement sampling rate and the Density parameter of the UE's in relation to their macro base stations.

Random Mode	This parameter lets you choose between using a static random function that gives identical results or a randomly generated random function.
MIMO Mode	In this parameter a user choses the maximum permitted performance boost that can come from the usage of a MIMO scheme. The effects of the MIMO scheme are shown in an increase in the Resource Blocks at the Scheduler Layer.
Scheduler Mode	In this input parameter you can select the scheduler type of the resource allocation layer for the LTE network. The two implemented schedulers is the normal round robin scheduler which favors the users with better channel conditions and the fair scheduler which gives more resources to users with worst channel in an effort to equalize the throughput for all users.

Fractional Frequency Reuse	This parameter is not yet implemented. It is designed to alter the amount of resources provided by different zones of coverage in the eNodeB in order to increase throughput and spectral efficiency.
Frequency Reuse factor	This parameter is not yet implemented. It is designed to influence the distribution of resource blocks throughout different sections of the playground in a way that the interference caused between them is minimum and also same frequency blocks are being reused.
Operator's bandwidth	This options is adjustable and dictates the total width of the available band and also the number of subcarriers provided for the generation of the LTE resource blocks.
Measurement sample rate	By adjusting this value, the graphical representation of the output data and the samples of the excel-generated files change in order to accommodate for different sampling frequencies.
Macro – Micro cell pivot	This slider lets the user select the percentage of the UE's that will be spawned in the playground near the macro base stations or near the pico cells.

4.10.2 Output

The simulator's output can be divided in three categories based on the form the output data takes. These categories are:

- Graphical function plots - .JPG images

- Excel Data sheets
- Console prints – Runtime Notifications

For the creation of a specific set of results we also need to accompany them with the settings xml file that describes the simulated scenario and also has the ability to be used in the simulation to recreate the same exact results. Below we analyze the measureable variables and their meaning in the simulation.

- 1 The SNR measurement. In the SNR measurement, the graphical plot displays for each of the selected UE's the measurement of the Signal-to-Noise-Ratio every second of the simulation.
- 2 The signaling bytes measurement. In this measurement, the graphical plot displays for each Serving RAT node the signaling bytes that we have measured as they generate for procedures that occur by the change in the network's context.
- 3 The signaling throughput measurement. This measurement analyzes measurement [2] by its first derivative in order to determine the rate that [2] changes.
- 4 The cell load measurement. This measurement shows the utilization of resources in every RAT. The values are from [0 – 1] and they are also used in the measurement of the power consumption in the eNodeB.
- 5 The Objective function score measurement. This measurement shows the value of the Objective function for each possible handover target at every second of the simulation for an example UE (default is 0).
- 6 The backbone bytes measurement. This measurement shows the amount of bytes that have been loaded to the RATs in order to serve the demand of the underlying UEs.
- 7 The backbone throughput measurement. This measurement shows the rate that [6] changes in order to determine its derivative size.
- 8 The UE uplink bytes measurement. This measurement measures the amount of bytes that have been successfully transmitted from the Up Link channel of the UE as they result from the traffic demand of the higher layers.
- 9 The UE uplink throughput measurement. This measurement measures the rate in which [8] changes in order to generate its derivative size.
- 10 The UE downlink bytes measurement. This measurement measures the amount of bytes that have been successfully transmitted from the Down Link channel of the UE as they result from the traffic demand of the higher layers.

- 11 The UE downlink throughput measurement. This measurement measures the rate in which [10] changes in order to generate its derivate size.
- 12 The SNR PDF measurement. This plot generates the probability density function of the average SNR rating for each selected UE of the System.
- 13 The SNR CDF measurement. This plot uses the [12] PDF in order to generate a CDF of the average SNR rating for each selected UE of the System.
- 14 The average downlink throughput PDF. This plot generates the probability density function of the average Downlink throughput for each selected UE of the System.
- 15 The average downlink throughput CDF. This plot uses [14] in order to generate a CDF of the average downlink throughput for each selected UE of the System.
- 16 The handover count measurement. This plot shows the value of the handover counter through all the operation of the simulation for each selected UE.
- 17 The average CQI measurement. This plot shows the value of the channel quality indicator for each selected UE of the simulation.

By using the abstract class measurement, we can easily generate more output results and to project them by using the JFreeChart API.

4.10.3 Constants

This software is using a very large amount of hard coded data in order to operate. All this hard coded data is placed inside a storage class that has public fields that can be accessed by the Main class in order to extract them in a form that is usable for the Simulation.

Table 7 - Table of Constants in the Simulator

MIMO Class	Constants that oversee the impact of the selected MIMO scheme in the scheduler and resource allocation layer of the simulator.
Notifications Class	Enumeration of different Notification messages of the simulator.
Scheduler Class	Enumeration of different Schedulers implemented
Intelligence Class	Enumeration of different Decision-Making Entities implemented.

Code Class	Constants that oversee the impact of the different coding schemes in the scheduler and the resource allocation layer of the simulator.
WiFi Class	Constants that oversee the impact of different backbone connections in a WiFi hotspot.
Modulation class	Constants that oversee the impact of the usage of different modulation schemes in the scheduler and resource allocation layer
FRFactor class	Constants that oversee the impact of different frequency reuse factors
FFRFactor class	Constants that oversee the impact of dividing the transmission zone into different fractions of the same band.
CQIMappings class	Constants that are used to link a channel quality index with a specific coding and modulation scheme.
LTEEnb class	Constants that are important for the function of the LTE ENodeB module of the simulator.
Adjacency class	Constants that are used by the objective function of the genetic algorithm (see Appendix 2) in order to determine the neighboring cells.
Genetic class	Constants that are used by the genetic algorithm toolkit (see appendix 2).
UE class	Constants that are used by the LTE UE module in order to describe its state machine.

Bandwidth class	Constants that describe different available bandwidth sizes for different provider requirements.
Mobility class	Constants that are used by the Mobility Stats class to implement the user preferences.
Application class	Enumeration of different applications and their respective QoS parameters.
Data class	Constants that contain Strings for the labels of the graphical user interface in all the stages of the execution in this program.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Chapter 5: Simulation scenarios and Results

In this chapter we will analyze the structure of the various simulated scenarios conducted for this research in the form of general assumptions, individual scenario – related information for input and simulation output. The tool used is the custom made Java simulator for the LTE networks created for this research (covered in the previous chapter) and the results will be used in the next chapter in order to form the recommendations as they derive from this study.

5.1 General Assumptions of the Simulation

The Simulator has a large set of parameters that can be either predefined or adjusted from the graphical user interface console it provides. In this section we will cover all the simulation fields that remained static for the next scenarios in order to stay clear about the simulated data that is being considered for this study.

The following assumptions have been considered for the results. The mentioned parameters are configurable, in order to be able to run simulations with different characteristics but in the case of our simulated scenarios we have kept them stable for better comparison.

Table 8 - Network Input general assumptions

Network Topology Parameters	Parameter Value
Area of the Simulation	1300 x 1300 meters
Areas and Inter-Site Distance (configurable)	Dense Urban Area (ISD 500m)
Number of macro-cells (configurable)	6
UEs per macro-cell (configurable)	20
UE movement speeds considered (km/h, configurable)	0, 3, 30, 60
UE mobility model	Random Waypoint model

Table 9 - Transmission input general assumptions

Transmission and Path Loss parameters	Parameter Value
Total transmission power of Macro eNodeB	46 dBm
Total transmission power of Pico eNodeB	30 dBm
Path loss model (Macro- UE)	$L= 128.1+37.6\log_{10}(R)$, R in km
Path loss model (Pico- UE)	$L= 140.7+36.7\log_{10}(R)$, R in km
RF (Carrier Frequency)	2.0 GHz

Table 10 - Signalling Input general assumptions

Signaling Parameters (calculated according to analytical study based on 3GPP reports)	Bytes	Delay for the transmission / interval value
UE attachment procedure	1189	94 ms (delay)
Data call initiation procedure	2013	188 ms (delay)
Data call release procedure	944	94 ms (delay)
Intra-LTE handover procedure	590	90 ms (delay for preparation and execution according to 3GPP)
RRC during idle state	84	80 ms (interval)

Table 11 - Network traffic general assumptions

Network traffic demand Parameters	Parameter value/description
Supported applications	Voice service, HTTP (data sessions)
Average time of each call (in sec)	90
Average number of calls per day per user	12
Data session payload size	16Mbit
Voice rate	32Kbps (symmetric)

Table 12 - Decision-Making general assumptions

Parameters about the Decision Making procedure	Parameter value
Processing delay for the operation in the UE	50ms
Processing delay for the operation in the Infrastructure (eNodeB)	Minimal (1ms)

Table 13 - ANDSF general assumptions

ANDSF template parameters	Parameter value
ANDSF enabled	True
ANDSF priority paradigm	LTE > WiFi private > WiFi public

Table 14 - Objective Function general assumptions

Optimization Parameters	Function	Value
SNR Parameter weight		0.6
Cell Load Parameter weight		0.2
ANDSF factor weight		0.2
Sum		1 (normalized function)
Algorithm margin		0.1

Table 15 - Miscellaneous general assumptions

Miscellaneous parameters	Value
Random generator seed	Static
MIMO	2x2
Scheduler	Round Robin
Bandwidth	20 MHz
Resource Block allocation	Orthogonally allocated

5.2 Test Cases used for this study

This section includes a sample of the scenarios we have executed with the LTE Simulator software. These scenarios are especially selected to analyze in the form of input and output and they will be referenced in the following section of conclusion in order to strengthen the recommendations of the output.

5.2.1 Comparison between different sizes of networks

The following section provides a set of results which are investigating the network-initiated and UE-initiated decision making. Decision making is related to cell selection and handover according to the fitness function that is analyzed in Section 3. Results have been obtained for two different network configurations in an area of around 1km².

Input:

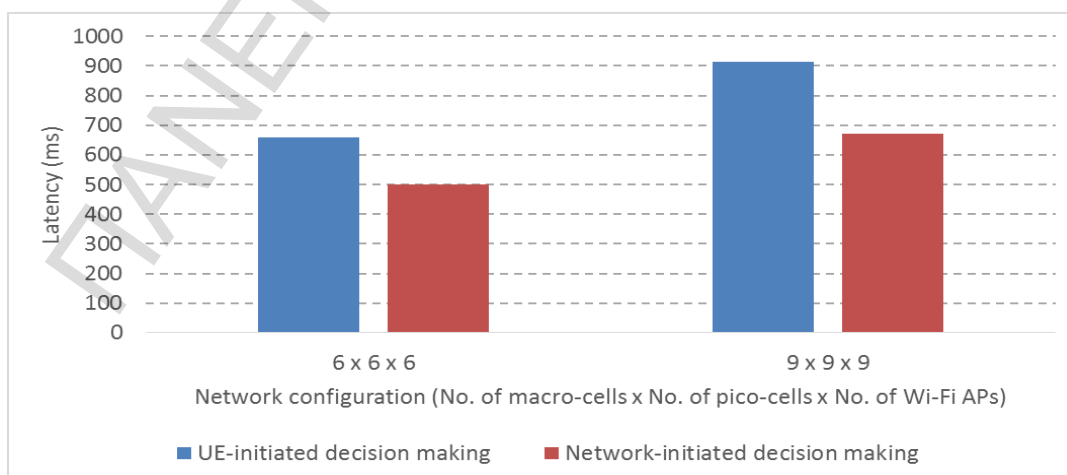
- Simulation assumptions of Section 5.1 are taken into account
- Scenarios differentiate according to these parameters:

Table 16 - Input parameters, scenario 1

Parameter	Value
No. of LTE macro cells	6, 9
No. of pico cells	6, 9
No. of Wi-Fi APs	6, 9
Speed of UEs	3 km/h
Simulation time	1 hour

Output:

In general, it is being observed that UE-initiated decision making has a higher decision-making latency (not counting the extra time needed for processing delay –which is a parameter affected by the device’s processor unit and the amount of data which need processing). Moreover, decision making is affected by the amount of cells and Wi-Fi APs, which are deployed in the area. Specifically, an increase in both UE-initiated and Network-initiated decision making is observed, at the order of around 35%, when the number of cells and Wi-Fi APs rises from 6 macro cells, 6 pico cells and 6 Wi-Fi APs to 9 macro cells, 9 pico cells and 9 Wi-Fi APs.

*Figure 32 UE and Network-initiated decision making latency*

7.2.2 Comparison between different deployments of Wi-Fi Aps

This subsection investigates the impact of having a constant number of macro cells, without pico cells and a variable number of Wi-Fi APs. Results have been obtained for three different network configurations in an area of around 1km².

Input:

- Simulation assumptions of Section 5.1 are taken into account
- Scenarios differentiate according to these parameters:

Table 17 - Input parameters, scenario 2

Parameter	Value
No. of LTE macro cells	6
No. of pico cells	0
No. of Wi-Fi APs	6, 12, 24
Speed of UEs	3 km/h
Simulation time	1 hour

Output:

Decision making is also affected by the amount of Wi-Fi APs which are available in the area. Specifically, an increase in both UE-initiated and Network-initiated decision making latency is observed, at the order of around 35%, when the number of Wi-Fi APs rises from 6 to 12 and around 50% when the number of Wi-Fi APs rises from 12 to 24.

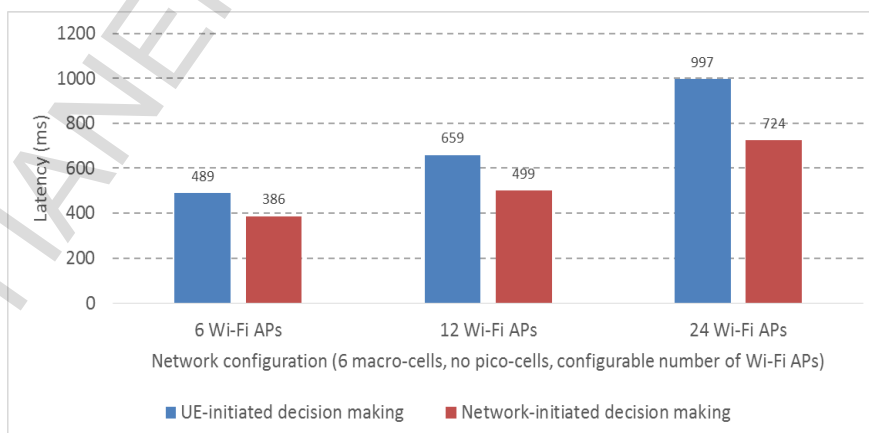


Figure 33 UE and Network-initiated decision making latency

5.2.3 Comparison between different decision making intervals

Input:

- Simulation assumptions of Section 5.1 are taken into account
- Scenarios differentiate according to these parameters:

Table 18 - Input parameters, scenario 3

Parameter	Value
No. of LTE macro cells	7
No. of pico cells	7
No. of Wi-Fi APs	0
Speed of UEs	3, 30, 60 km/h
Decision-making interval	1, 2, 5, 10 seconds

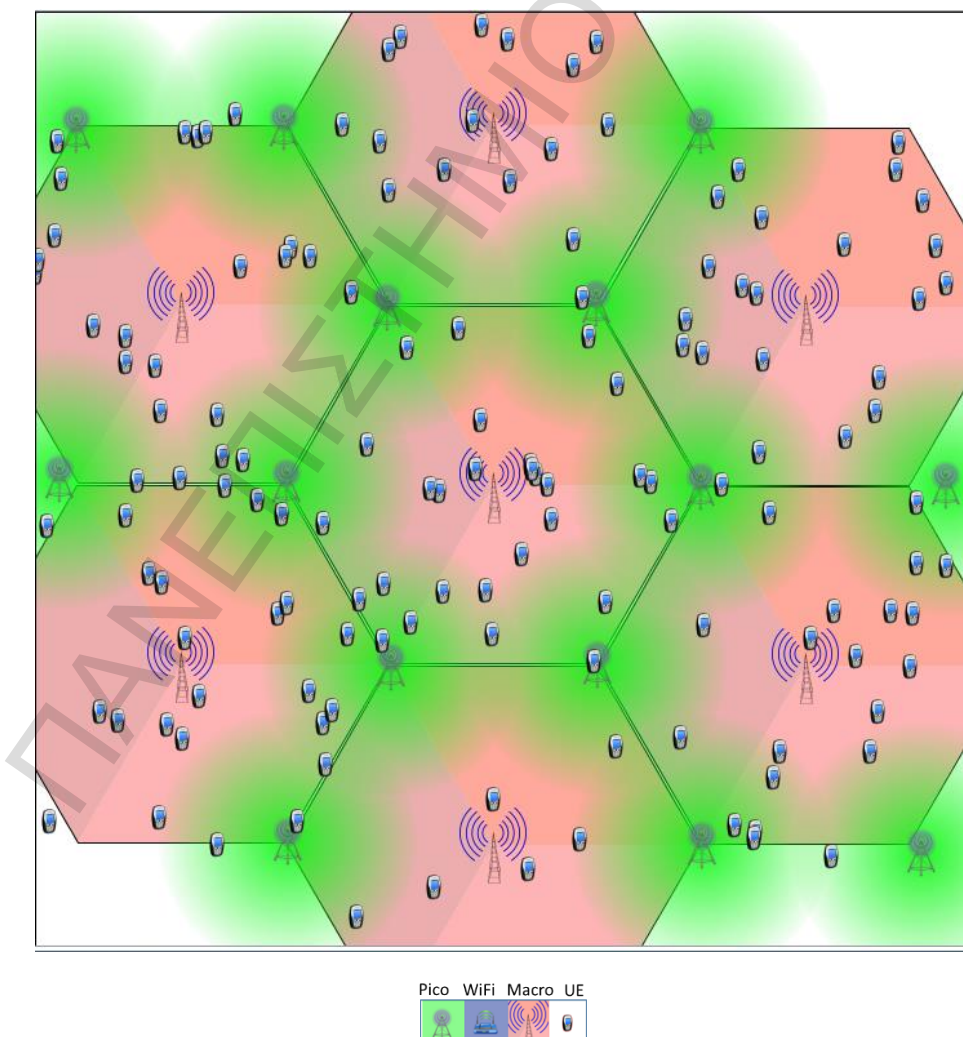


Figure 34 Considered topology

Output:

In this result set, the influence of the four different decision making intervals on the handover count for each individual UE speed is measured. An important result is the increase in the average number of handovers (handover count) with the increase of the UE movement speed. Another observation is the decline of the average handover count as the decision making interval increases. This happens because by checking for the best cell with smaller frequency, the probability that a UE will discover a better cell drops. Also greater decline is observed in the higher mobility scenarios where the UE context changes faster.

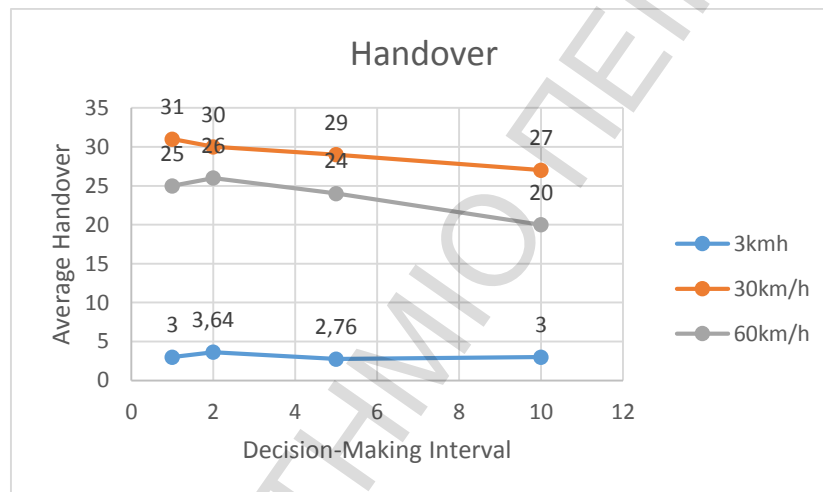


Figure 35 – Average handover count for each criteria (decision-making interval in seconds)

In Figure 36, the average signaling bytes for the system during each of the selected scenarios is measured. In the case of the 60km/h scenario, the highest amount of signaling bytes is observed due to the fact that the handover procedures are happening more frequently. UEs are served by the macro base stations because of their high mobility. The scenarios of 3-30 km/h have significantly lower value in this metric and also appear to be unaffected by the change in the decision making interval. In the case of the 60km/h scenario, we can clearly see that the 5 and 10 seconds interval stabilizes the signaling byte value at a certain threshold (~61MB), while in 1 and 2 seconds the signaling tends to increase. According to the observations, the macro base station's cumulative bytes are influenced by the decision making interval. In fact it is the slow sampling rate of the context domain in correlation to its fast change rate that triggers many handovers.

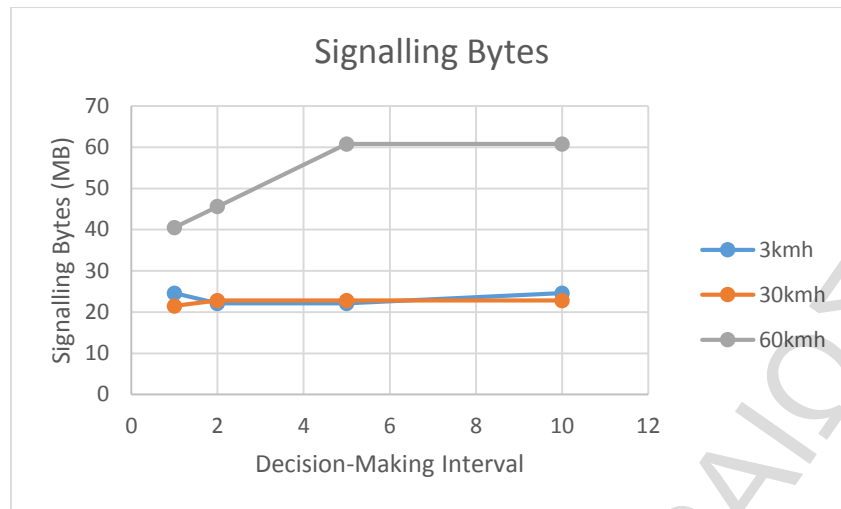


Figure 36 – Cumulative signalling bytes per scenario (decision-making interval in seconds)

Figure 37 shows the comparison of the average CQI index on a set of UEs in scenarios with different mobility levels and different decision making interval. An overall observation can be that as a user equipment's average speed increases, the average channel quality indicator declines. This is expected because with higher mobility the probability of a UE being located in a cell edge area is increased. What also can be generally observed is the effect of different decision making intervals have on the same index. In the cases of the highest mobility, an increase on the decision making interval leads to a greater decrease in the channel quality indicator index. This is predictable as the slower the decision-making interval is, greater the duration of the non-optimized operation of the LTE UE will be. Therefore the user equipment is served for a larger amount of time by the non-best cell which causes this decline in the average CQI. On the other hand, this is not the case in the 3km/h scenario where the effects of the decision making interval are not following a certain trend. Due to the fact that this behavior is not frequently observed, this fluctuation is dismissed as a statistical error.

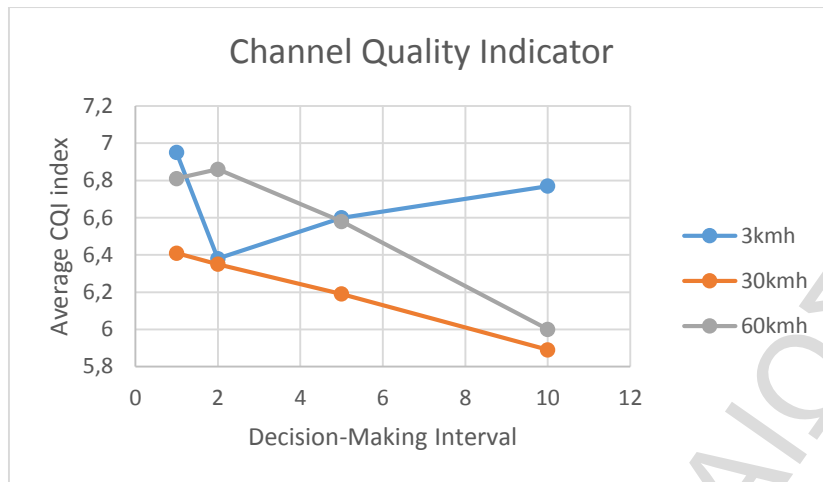


Figure 37 – Average Channel Quality Indicator per scenario (decision-making interval in seconds)

In Figure 38, the average throughput metric for the different simulation scenarios is measured. The average throughput value of the LTE wireless system is more sensitive to users with higher mobility. Regardless of the decision-making interval, a decline is observed as the movement speed increases. In general, the greater decision making interval shows an additional decline to the data rates following the diagram of the Channel Quality Indicator (CQI) shown previously.

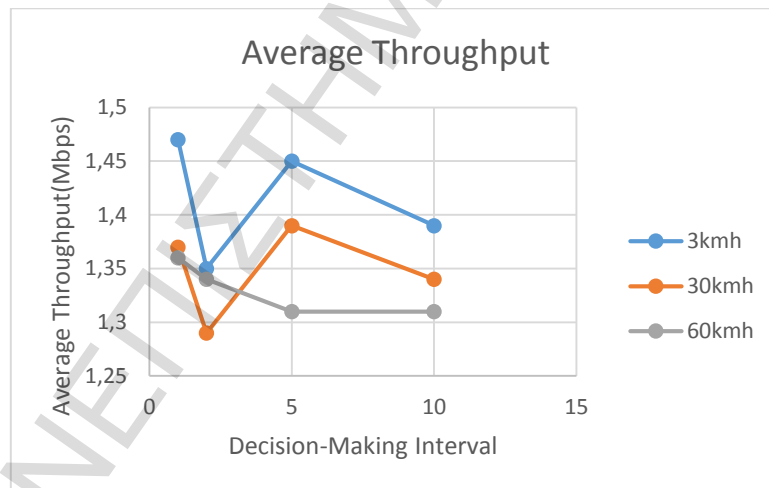


Figure 38 – Average Downlink throughput per scenario (decision-making interval in seconds)

5.2.4 Comparison between Network-initiated Decision-making and User Equipment-initiated Decision-making

Input

- For this scenario, the original assumptions (mentioned in section 5.1) are used to run two simulations with their only difference the decision making entity of the system.

Table 19 - Input parameters , scenario 4

Parameter	Value
No. of LTE macro cells	7
No. of pico cells	7
No. of Wi-Fi APs	0
Speed of UEs	3 Km/h
Decision-making entity	UE/Network

- The considered topology of the network is identical to the one used in the scenario mentioned previously.

Output:

For the output of this scenario, various parameters are being measured and the average value is calculated in order to evaluate the impact of each of the different decision-making entity to the operation of the system. These parameters vary from quality of service parameters (throughput, channel quality indicator) to systemic parameters that concern more the network operator such as (average handoff count, signaling bytes).

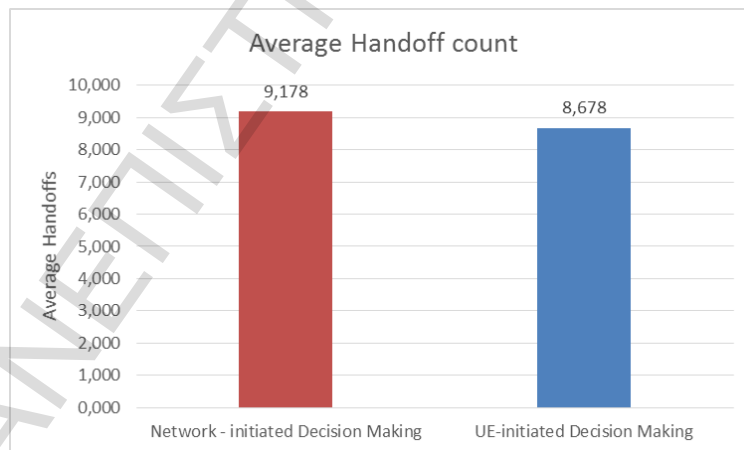
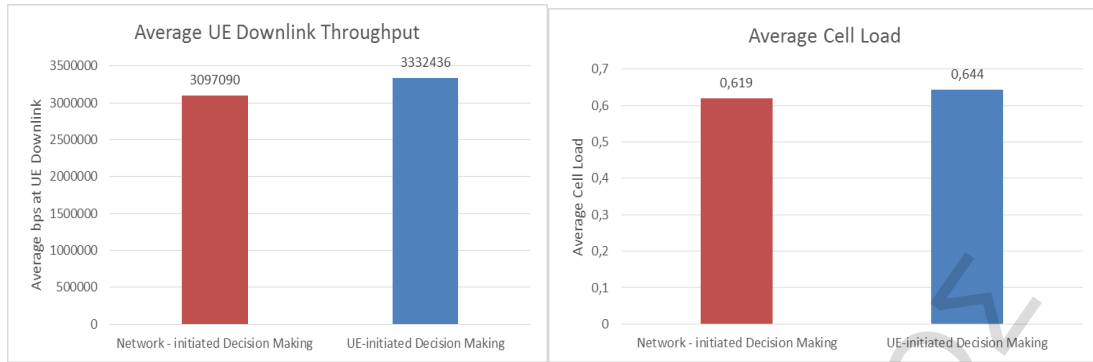


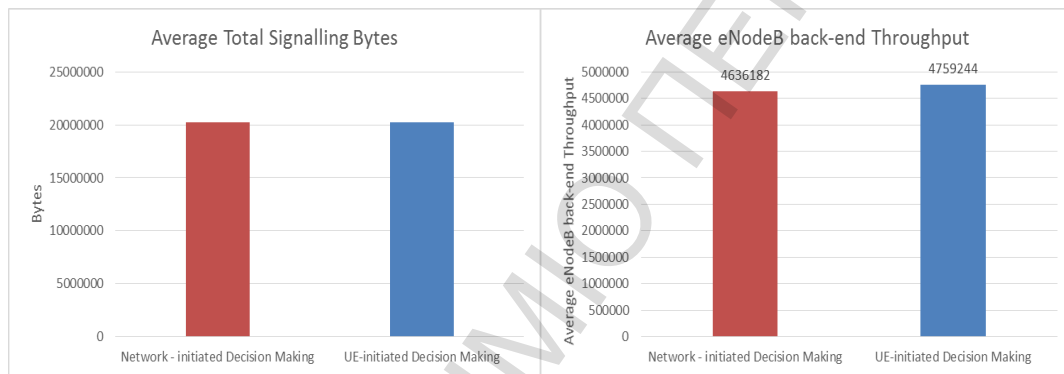
Figure 39 - Average Handoff count for the UE and Network initiated Decision Making



(a)

(b)

Figure 40 – (a) Average UE Downlink Throughput (b) Average Cell Load for the Network and the UE initiated decision making



(a)

(b)

Figure 41 – (a) Average cumulative signalling Bytes (b) Average eNodeB back-end Throughput for the Network and the UE initiated decision making

As we can see in these diagrams, the UE initiated decision making results in higher average CQI ratings which influence the increase in the downlink throughput and simultaneously the eNodeB back end throughput while seeing very small differentiation in the total signalling bytes for each cell. On the other hand we see that with Network initiated decision making the average cell load metric is lowered compared to the UE initiated.

Chapter 6: Recommendations and Conclusions

In this section, the simulated assumption mentioned in Section 5, combined with the output and the statistical analysis of the simulated scenarios, create a series of recommendations regarding different (and sometimes conflicted) stakeholders. In some cases the results do not differentiate greatly, in a way that the outcome of this research is clear, but if meta-considerations take place it will be clearer that the entity that seems more suitable for the decision making operation is the **LTE eNodeB**.

6.1 Recommendations regarding the Mobile Network Operators

The mobile network operator entity in the LTE system faces a great amount of risks that result from the complexity of its context. As the radio technologies are pushed to their limits, the simulation tools become more complex in order to fully account for all parameters. In this respect, the following set of considerations can be made based on the outcome of the simulated scenarios that we have investigated in this study:

- In the case of **Network-initiated Decision Making**, the cost of acquiring the ANDSF policy template is extremely low due to the fact that the interconnectivity between the LTE eNodeB and the remote server hosting those templates is served by *wired* network. Therefore no resource reservation when it comes to the radio interface needs to be considered. This gives great flexibility in terms of using many different templates in order to alter the network performance and behavior for various occasions.
- As we can see from Test Case 1 and Test Case 2 (sections 5.2.1, 5.2.2 respectively), it is clear that the Decision-Making delay is smaller in the Network-Initiated decision making than the User Equipment initiated case. As the network size raises due to the positioning of WiFi offloading hotspots (section 5.2.2) or the positioning of more pico-cells and macro-cells (section 5.2.1) we see that the gap between these two values raises. Meta-estimations of the final output result from test case 5.2.2 show that the difference in delay between Network and UE initiated decision making can reach around 300ms. If we correlate this result with a moving user moving at maximum average speed of 60 km/h we can provide some estimations to determine the impact of this procedure:

- $60 \text{ km/h} = 16.6 \text{ m/s} \rightarrow 16.6 \text{ m/s} * 0.3\text{s} = 5 \text{ meters}$

- This means that the validity of the handover procedure will be burdened by a +/- 5 meter factor in the case of UE-initiated decision making operation compared to the Network-initiated case. These meters would not be taken into account into a GSM network with a single type of cells but with the heterogeneity of the LTE networks and the multiple RAT's used with different transmission powers such as the pico cell and the WiFi hotspot a fluctuation of a few meters in near cell edge context can result in call drop and signaling failure.
- Further analysis of this result compared to different movement speed and radio access technologies show that:

Table 20 - Mobility correlation with RAT's antenna coverage

	Velocity 3kmh = 0.83 m/s	Velocity 30kmh = 8.3m/s	Velocity 60kmh = 16.6 m/s
Network-Initiated Decision making delay = 724ms	Distance covered during the function= 0.59m	Distance covered during the function= 5.9 m	Distance covered during the function= 12.01 m
User Equipment-Initiated Decision making delay = 997 ms	Distance covered during the function= 0.83m	Distance covered during the function = 8.3m	Distance covered during the function = 16.2 m

Now if this distance is compared to each different radio access technology's respective coverage radius, we can create the percentage of worst case scenario coverage misplacement for each of the above case: If this percentage exceeds **the 5% of the RAT radius**, then there is high probability that the decision making operation will occur falsely. For percentages near **20% of the RAT** radius the operation fails with very high probability (~95%).

For the Network-Initiated Decision making function the following data derives:

Table 21 - Failure percentage of handover procedure for scenario 1

Network Initiated Decision Making	Velocity =3kmh	Velocity = 30kmh	Velocity = 60kmh
Macro cell radius = 250 m	$0.59/250 = \ll 1\%$	$5.9/250 = 2\%$	$12/250 = \sim 5\%$
Pico cell radius = 100m	$0.59/100 = < 1\%$	$5.9/100 = \sim 6\%$	$12/100 = \sim 12\%$
WiFi AP radius = 70m	$0.59/70 = \sim 1\%$	$0.59/70 = \sim 10\%$	$12/70 = \sim 17\%$

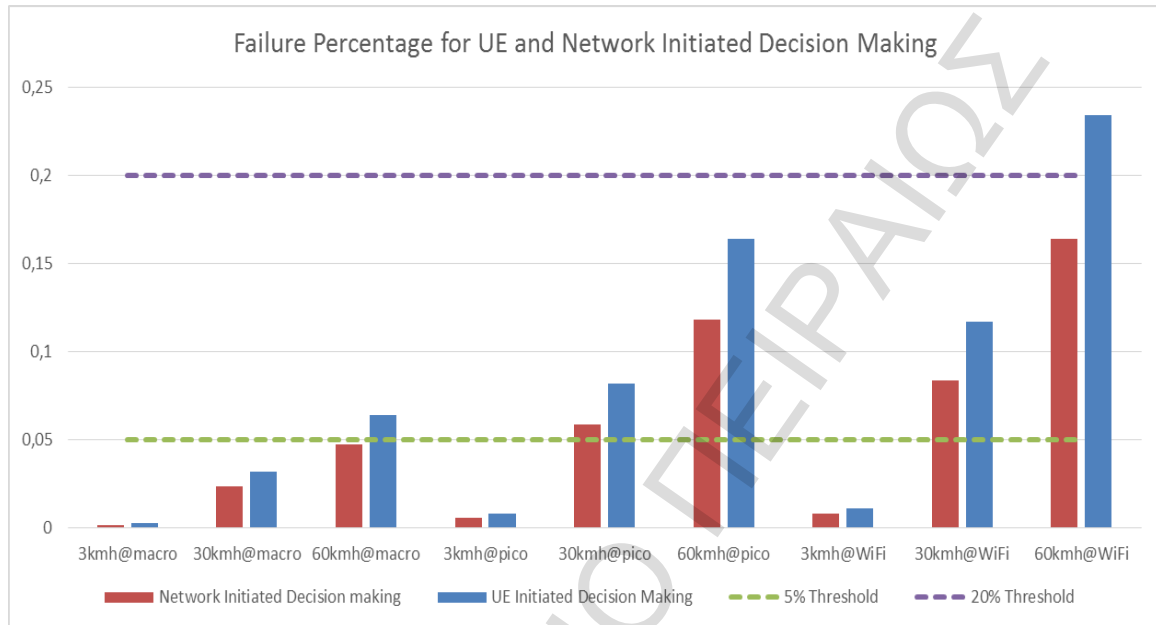
And for the User Equipment Initiated decision making function the following data derives:

Table 22 - Failure Percentage of handover procedure for scenario 2

User Equipment Initiated Decision Making	Velocity =3kmh	Velocity = 30kmh	Velocity = 60kmh
Macro cell radius = 250 m	$0.82/250 = \ll 1\%$	$8.2/250 = \sim 3.2\%$	$16.4/250 = \sim 6.5\%$
Pico cell radius = 100m	$0.82/100 = \sim 1\%$	$8.2/100 = 8.2\%$	$16.4/100 = \sim 16.4\%$
WiFi AP radius = 70m	$0.82/70 = \sim 1\%$	$8.2/70 = \sim 12\%$	$16.4/70 = 23\%$

These two charts put together give us a clearer picture about how the reduced delay from the **Network Initiated Decision Making** function affect the heterogeneous network of our study:

Table 23 - Comparison between the two scenarios



As we see in figure, the Network Initiated decision making operation has more end results closer to the soft threshold of the operation therefore it results as more effective.

- In the horizontal comparison of the two decision making cases (section 5.2.4) we see that both of the implementations have their positive and negative effects to the network operator. It is also observed that with network-initiated decision making, it is possible to keep the cell load values lower than the UE-initiated case. This can be considered crucial for the network operators because especially in dense urban environments, the management of the ever increasing load is one of the key aspects of operating a successful cellular network. The amount of average signaling bytes in each case seems to be at even values in both the UE and Network while the handoff count is increased in the case of Network-initiated decision making due to the fact that the procedure is completed more quickly. In the case of the throughput related variables, we see that the lower amount of handovers result in higher average channel quality indicator (CQI) index for the user equipment that results in higher average throughput and symmetrically higher LTE eNodeB back end throughput to serve that traffic. While the increase in the throughput is a welcome aspect, we

find that the better load balancing factor outweighs it in a way that favors the Network-Initiated decision making procedure.

- The Long Term Evolution of the 3GPP is designed with revolutionary relocation of the decision making entity, from higher up in the network hierarchy as seen in the UMTS family, to the highly capable LTE eNodeB access network node. The interconnectivity between these nodes with the X2 interface creates a very flexible intercommunicating environment with very low delay and a hive-like way of coordinating its operation. That combined with the many, lower tier, access entities (pico cells, Femto cell, wifi AP's) gives us the impression that the LTE eNodeB is better suited as the key entity for the decision making operations of the network.

Taking into consideration the previous key points of this study, we find that the lower delay, the better behavior of the procedure with higher mobility users, the better load balancing outcome and the technological advances that happen in the LTE eNodeB entity that the Network-initiated Decision Making function will be better suited for the Network Operators.

For this conclusion we have used results created with the considerations of our custom made simulation tool, the input parameters shown in section 5.1 of this document, the simulated scenarios of section 5.2.1 to 5.2.4 and our experience with the 4th generation of the 3GPP cellular networks.

6.2 Recommendations regarding the User Experience

The End user and owner of the LTE user equipment device is a different stakeholder of this system paradigm. User's requirements focus on availability and higher wireless resources (which translates into higher throughput) but also considers the energy consumption of its device. Below we present the key points of our recommendation regarding the End User stakeholder:

- Using the output results of Test Case 4 where we see the horizontal comparison of the User equipment and Network initiated Decision making function, we see that the operation that better suits this stakeholders system requirements is the **User Equipment**. That being said the increase in the Throughput on average is not extremely high therefore extensive simulation and real life paradigms could indicate otherwise.
- In the case of the UE-initiated decision making entity, the cost of acquisition a single ANDSF template in order to be used locally is considerable. On the other hand, due to the fact that the frequency of change in this template is daily or in some cases weekly, then the impact can be considered minimal.

- The delay of conducting the decision making functionality in the user equipment is higher (as it results from section 5.2.1 and 5.2.2 output) and this results in situational out of context decision making behavior (as seen in section 6.1). Also this study has no reference hardware device for the user equipment to build a processing load mechanism so we cannot predict the additional delay that the repetition of the procedure may induce.

Taking the previous points into consideration, we believe that the User Equipment entity can be used as the decision making entity in order to achieve higher final throughput values for the End User.

As we have mentioned before, for this recommendation many parameters have been either considered fixed or been thoughtfully ignored mainly because of no reference hardware device. Putting this aside we believe that it is expected to see this conflict between the Network Operator and the End user result, as their success depends on different key aspects of the system.

6.3 Recommendations regarding the LTE equipment vendors

This section is dedicated to recommendations concerning equipment manufacturers. The manufacturers of this technology can be considered important in a sense that by producing LTE eNodeB systems they most probably prefer that all the intelligence functionalities are being conducted in their product, making it more important for the implementation of the LTE functionality. A disadvantage is that due to low system functionality information for LTE devices, is hard to proceed to accurate estimation of the power consumption related to the decision making operation, or the effects of multiple requests being processed (due to the fact that LTE eNodeB entity will be conducting this operation for *each* user equipment device). Therefore the following point can be used to help us conclude to a decision:

- The output results of Test Case 4 (horizontal comparison of Network and User Equipment initiated decision making) give us a picture of different average load values for each case. The average system load metric can be considered as a key aspect for this stakeholder because meta-considerations can link it to the average power consumption of the LTE eNodeBs and as a result to the OPEX of the system. The LTE device vendors need to advertise the power consumption capabilities of their devices therefore we believe that the results of the Network-initiated decision making seem better suited for this stakeholder.

- The analysis of the decision-making interval (as seen in section 5.2.3 of the document) indicates that the system's performance is greatly affected by slower decision making (i.e., with higher decision making interval) functionality. If we consider that the Network-initiated decision making functionality can easily be conducted faster than the, energy-intensive UE initiated, then we can see that all the benefits of the smaller decision making interval favor the Network entities.

It is hard to determine the impact of the decision making operation entity with regards to this stakeholder, mainly because we lack architectural reference to test the operation. However, we can see many preliminary indications that the Network-initiated decision making will be more preferable for this stakeholder.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

References

- [1] 3GPP TR 32.835 v 1.2.0 , “Telecommunication management: Study of Heterogeneous Networks Management (Release 12)” , Jun. 2013
- [2] 3GPP TS 24.007 v12.0.0 , “Mobile radio interface signaling layer 3 : General Aspects (release 12)” , Jun. 2013
- [3] 3GPP TS 24.008 v 12.2.0 , “Mobile radio interface signaling layer 3 : Core network protocols , Stage 3 (Release 12)” , Jun. 2013
- [4] 3GPP TS 24.301 v 12.1.0 , “Non-access-stratum (NAS) protocol for Evolved Packet System , Stage 3 (Release 12)” , Jun. 2013
- [5] 3GPP TS 29.274 v 12.1.0 , “Evolved General Packet Radio Service (GPRS) Tunneling Protocol for Control plane (eGTPv2-C) , Stage 3 (Release 12) , Jun. 2013
- [6] 3GPP TS 36.331 v 11.4.0 , “Evolved Universal Terrestrial Radio Access (E-UTRA) , Radio Resource Control , Protocol specification” , Jun. 2013
- [7] 3GPP TS 36.413 v 11.4.0 , “Evolved Universal Terrestrial Radio Access Network (E-UTRAN) , S1 Application Protocol (S1AP) (Release 11), Jun. 2013
- [8] 3GPP TS 36.423 v11.5.0 , “Evolved Universal Terrestrial Radio Access Network (E-UTRAN) , X2 Application Protocol (X2AP) (Release 11), Jun. 2013
- [9] Nokia Siemens Networks , White Paper , “Realization of Policy-based resource management concept” , Version 1.0 , Feb. 2010
- [10] Motorola, Technical White Paper , “Long Term Evolution (LTE) : Overview of LTE Air-interface”, 2007
- [11] Ericsson , White Paper , “LTE Release 12 :Taking another step toward the networked society” , Jan. 2013
- [12] S. Sesia , I. Toufik , M. Baker , “LTE – The UMTS Long Term Evolution : From theory to practice” , Second Edition , Wiley, Jul. 2013
- [13] P. Lescuyer , T. Lucidarme , “Evolved Packet System : The LTE and SAE evolution of 3G UMTS” , Wiley , Jan. 2008
- [14] Rohde & Schwarz , Application Note 1MA111, “UMTS Long Term Evolution : Technology Introduction” , Sep. 2008
- [15] G. Ku , Drexel , Presentation , Adaptive Signal Processing and Information Theory Group , “Resource Alloaction in LTE” , Nov. 2011

- [16] S. Hussein , “Dynamic Radio Resource Management in 3GPP LTE” , Master Thesis , Blekinge Institute of Technology, Jan. 2009
- [17] F. Sandu, S. Cserey, E. Mile-ciobanu, "Simulation of LTE Signaling", *Advances in Electrical and Computer Engineering*, vol. 10, no. 2, pp. 108-114, 2010, doi:10.4316/AECE.2010.02019
- [18] A. Kaloxylou , F. Georgiadis , I. Modeas , N. Passas , “Design and Implementation of Radio Access Selection Algorithm for Multi-mode Mobile Terminals” , 1st International Conference Mobile Networks and Management (MONAMI) , Athens , Greece , Oct. 2009
- [19] T. Lian , K. Sinkar , L. Kant , K. Kerpez , “Resource Allocation and Performance Study for LTE Networks Integrated with Femtocells” , *Global Telecommunications Conference (GLOBECOM 2010)*, 2010 IEEE , vol., no., pp.1,6, 6-10 Dec. 2010
- [20] A. K. Afridi , “Macro and Femto network aspects for Realistic LTE usage scenarios” , Master Thesis , KTH Electrical Engineering , 2011

Appendix I – Genetic Algorithm Implementation

Problem Model

For this software we have used an implementation of a genetic algorithm to solve the continuous problem of placing N macro cell's in a playground with a specific width and height in a way that their inter site distance is equal to the input parameter selected. We understand that this problem can be solved used geometrical identities, but due to the importance of the scientific field of genetic problem-solving we chose to approach it with this implementation.

For the representation of the position of the N cells within the space of the playground, an array of integer has been selected. Every two consecutive integer values indicate the X and Y position of each macro cell. So the array consists of $2N$ integer values.

The fitness function of the genetic algorithm routine is the following:

Input:

- Genom of genetic algorithm (Integer array)
- Wanted Inter-Site distance (Integer in meters from the GUI)=IST
- Number of cells = nOc
- Adjucency matrix (1 if two cells aren't neighbors 2 if they are)

Minimize:

$$F_{genom} = \sum_{i=0}^{nOc} \sum_{j=0, j \neq i}^{nOc} |((IST - distance(cell(i), cell(j)) * adj(i, j))|$$

Subject to:

- Genom_count = Max_Genom_count
- Generation <= Max_Generation
- The i, j exist in $\{0, 1, \dots, nOc\}$
- The $i \neq j$ applies

Genetic Algorithm

The main skeleton of the methods that form a genetic algorithm consists of:

1. Initialization of the population
2. Selection of the population
3. Breeding of the population
4. Mutation of the population
5. Go to next generation and repeat step 2 until some threshold is reached*

*The threshold may be a real time threshold or a discrete threshold such as generation count

This is a description of the algorithmic procedure that uses the laws of natural selection in order to alter the information that lies inside the population's chromosomes.

Initialization of the Population

This method begins the genetic algorithm procedure by creating an initial population of N genoms that are generated randomly inside the playground. This is done by invoking the createNGenom method which then invokes N times the generateTopologyGenom function. The resulting population is stored in a 2-dimensional integer array and passed on to the next step of the algorithm.

Selection of the Population

The selection procedure of this genetic algorithm implementation is using the roulette routine. An implementation of the roulette routine can be found in the source code of this document (Appendix II). The roulette random is a data structure that needs as input a vector of values and generates a discrete probability density function in order to select with higher frequency the dimensions of the vector with higher value. This vector is generated by each genom's objective function's value and through the roulette random procedure, the most fit chromosomes are selected to take part into the next procedures.

Breeding of the Population

The breeding procedure is a very important part of the genetic algorithm because it causes a very successful mixing of the information between the most fit genoms. In this implementation we execute the operation of tail-swapping which in practice is the swap between the bottom half values of the integer arrays that consist the two genoms.

Mutation of the Population

Mutation is a procedure that give the element of randomness in the genetic procedure. With a curtain chance (given to this implementation by the constants class) a genom may alter one of its value to an adjacent value (+/- 5 meters). This causes an exploration of the possible solution space that would not occur in the case of the simple breeding procedure.

Repeat until generation limit

These procedures are executed in a repetitive manner. By doing so we explore the space of the possible solutions with two basic principles :

- 1 If two solutions are good, their offspring will also be good.
- 2 If a solution is good its mutated form may be even better.

Even if the previous statements aren't absolute, they are correct on average and this causes the total algorithm to incline towards the better suboptimal solutions.

Results

In this chapter we produce indicative results that the genetic algorithm used in this problem model generates logical output by measuring the average fitness of all the population in each generation of the iteration. A genetic algorithm tends to follow a very steep curve until it reaches a point of maturity where all the members of the population have almost identical chromosomes. At that point and forward the search in the solution-space is very limited only by the mutation function.

Input of the algorithm:

Table 24 - Algorithm Input

Number of Cells	7
Generation Limit	100 generations
Population count	100 chromosomes
Mutation Possibility	0.2 / chromosome
Mutation Value	+/- 10 meters

Output of the algorithm:

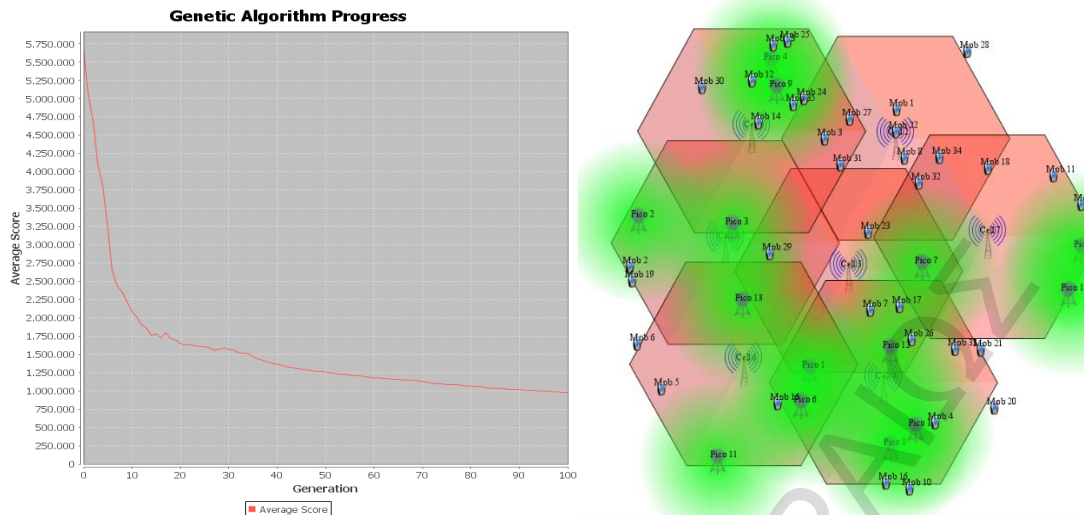


Figure 42 - Genetic Algorithm Results

As expected, the genetic algorithm quickly improves its average score until generation 30 and from that point it performs micro-adjustments according to the mutational variations. It should be noted that as the objective function used to evaluate the chromosomes gets more accurate, the genetic algorithm should perform better. Having said that, this solution is plausible if we take into consideration that the underlying problem is a continuous (not discrete) problem.

Appendix II – Simulator's source code

```

package com.sim;
import java.util.ArrayList;
import java.util.Date;
public class Simulator { //Klasi prosomoiwths
    //Simulator Variables
    private static long time=0;
    private static long eventserial =1;
    public static int simindex =0;
    private static long start=0;
    public static Date startdate;
    private static long stop=0;
    private static boolean running=false;
    private static boolean pause=false;
    public static int wait=0;
    //private Main main;
    private static ArrayList<Event> eventqueue = new ArrayList<Event>();
    private static ArrayList<Event> tobequeue = new ArrayList<Event>();
    //TEST
    private static ArrayList<Event> exitqueue = new ArrayList<Event>();
    //Constants
    public static long MOBILITY_SAMPLING_RATE=0;

    public static final long STEP =1;//long STEP =1 --> 10.000 steps = 1 sec
    public static final long RATIO = 10000; //10.000 simseconds = 1 sec
    public static final long MILISECOND=RATIO/1000;
    public static final long SECOND =RATIO;
    public static final long MINUTE =RATIO*60;
    public static final long LTE_UE_FUNCTION_RATE=MILISECOND*100; //1
function kathe 100 ms
    public static final long HOUR = MINUTE*60;
    public static final long DAY = HOUR*24;
    /**
     * Method that returns the time in Simulation Seconds
     * @return The time.
     */
    public static long now(){
        return time;
    }
    /**
     * Method that returns the start Date in String form
     * @return The String of start Date.
     */
    public static String timeNow(){
        //long seconds = time/Simulator.RATIO;
        //return start+seconds;
        return startdate.toString();
    }
    /**
     * This Method is helping any graphical user element determine the
     * completion percentage of the simulation. The MouseMovementThread
     * uses this method to alter the JProgressBar element to correspond
     * to the current Simulation time.
     * @return The Completion Percentage of the Simulation {0-1}
     */
    public static double percentage(){
        double d =(double)time;//(double) (time/stop);
        double d2=(double)stop;
        double d3=d/d2;
        return d3;
    }
    /**
     * This Method forces the Simulator to Stop by immediately setting the
     internal
     * Clock to 1 sim-second before the initial completion.Of course all the
events
     * scheduled to occur get dismissed and this operation definitely makes
the Simulation
     * lose its credentiality.
     */
    public static void forceStop(){
        stop=now()-STEP;
    }
}

```

```

/**
 * This method is used to real-time-alter the value of the pause variable
 * and pause the operation of the simulator for indeterminate time.
 * @param how True if you want to pause and False if you want to unpaue
 */
public static synchronized void setPause(boolean how){
    pause=how;
}
/**
 * This Method is used to determine if the Simulator is currently Running
 * but has Paused due to the user's keyboard prompt (Keyboard
Space).While
 * the simulator is paused , new events aren't's triggering and the
graphical
 * representation of the system is switched to a blurred-out version of
its
 * original , last view.
 * @return The Boolean value of pause : True of False
 */
public static synchronized boolean isPause(){
    return pause;
}
/**
 * This method is used to determine if the Simulator is currently
Running.
 * The value of this method is altering the Graphical User Interface
Ele-
 * ments in order to change their effect on the simulator i.e. Start to
Stop
 * buttons.
 * @return The Boolean true or false of whether the simulator is running
or not.
 */
public static synchronized boolean isRunning(){
    return running;
}
/**
 * This Method's purpose is to slow down the Simulation
 * in order to give time to the graphical renderer from the
 * Topology Panel class to draw the context.Dramatically slows
 * down the Total Simulation time if used in conjunction with
 * heavily loaded Event Queue's.
 * @param value The delay with every Simulation Iteration Cycle
 */
public static synchronized void setInterval(int value){
    wait=value;
}
/**
 * This method initializes the starting Date and Time of
 * the Simulator.Certain events take not only the differential
 * time as a trigger but the absolute time as well. The data rate
 * modelling we have used takes the absolute time as an input
 * variable in order to match the statistical data from the LTE
 * data rate curve.
 * @param dat the date to start the simulation
 */
public static void setStartTime(Date dat){ //Arxiki Wra tis prosomoiwsis
    startdate=dat;
    start=startdate.getTime();
}
/**
 * This Method sets the Duration of the Simulation in Seconds.
 * This means that the Simulator.start() method will last for
 * as many seconds as this method is set.
 * @param stopvalue How many seconds will the Simulator run
 */
public static void stop(long stopvalue){//Seconds
    stop=stopvalue*Simulator.RATIO;
}
/**
 * The basic Event Queuing procedure. This procedure is
 * crucial for the simulator because it places new events
 * in the pre-event queue area called tobequeue. The simulator

```

```

toqueue.add(e);
e.setSerial(eventserial);
eventserial =eventserial+1;
}
/**
 * This Method is created in order to add events that
 * resolve concurrency automatically. It is not yet
 * function but the idea is to simplify the event queueing
 * procedure.
 * @param e The event to be scheduled in the Simulator.
 */
public static void scheduleDiscrete(Event e){//TODO de leitoyrgei fix
long thetime = e.getTime();
if(thetime<time||thetime>stop){
return;
}
}
ArrayList<Long> times=new ArrayList<Long>();
boolean flag=false;
for(Event ee:eventqueue){
times.add(ee.getTime());
if(ee.getTime()==thetime){
flag=true;
}
}
for(Event ee:toqueue){
times.add(ee.getTime());
if(ee.getTime()==thetime){
flag=true;
}
}
if(flag){
long temptime=thetime;
boolean flag2=false;
do{
temptime = temptime+1;
for(Long l:times){
if(l==temptime){
flag2=true;
break;
}
}
}while(flag2);
e.setTime(temptime);
}
toqueue.add(e);
e.setSerial(eventserial);
eventserial =eventserial+1;
}
/**
 * Simulator's Main method. Begins to run a simulation
 * for the given stop time. During that time all events
 * Scheduled to trigger get activated, impacting the
 * Main class and changing the graphical and system con-
 * text.At the end the Simulator cleans up and gets ready
 * for new simulation.
 */
public static void start(){
long time1 = System.currentTimeMillis();
if(stop==0.0){
System.out.println("No stop Value");
System.exit(0);
}
Simulator.running=true;
int eventcount1=0;//events executed
int eventcount2=0;//events pushed
do {
//Pause
do{
}while(pause);
//Optimization variables
long lowest = Long.MAX_VALUE;
boolean foundevent=false;
//~
ArrayList<Integer>toberemoved = new ArrayList<Integer>();
int counter =0;

```

```

for(Event e:eventqueue){
    if(e.getTime()<=time){
        foundevent=true;
        eventcount1 = eventcount1+1;
        e.execute();
        toberemoved.add(counter);
    }
    else{
        if(e.getTime()<lowest){
            lowest=e.getTime();
        }
        counter = counter +1;
    }
    for(int a=toberemoved.size()-1;a>=0;a--){
        eventqueue.remove(eventqueue.get(toberemoved.get(a)));
    }
    for(Event e:tobequeue){
        if(e.getTime()<lowest){
            lowest=e.getTime();
        }
        eventcount2 = eventcount2 +1;
        eventqueue.add(e);
    }
    //Cleanup of complementary Event queues
    toberemoved.clear();
    tobequeue.clear();
    //Simulator optimization -->skip ton xrono poy den yparxoyn
events!!
    if(foundevent){
        time=time+STEP;
    }
    else{
        time=lowest;
    }
}
//Rythmiseis tis simulation date
startdate = new Date(start+time*1000/Simulator.RATIO);
//Delay logw toy slider
if(wait!=0){
    int a= wait*1000;
    int k=1;
    for(int i=1;i<a;i++){
        k = k*i;
    }
}
}while(time<=stop);
long time2= System.currentTimeMillis();
System.out.println("Time Elapsed: "+(time2-time1)/1000);
cleanup();//cleans up
}
/**
 * Method that re-initialized the Simulator
 */
public static void cleanup(){
    time=0;
    Simulator.running=false;
    Simulator.simindex=Simulator.simindex+1;
    Simulator.eventqueue.clear();
    Simulator.tobequeue.clear();
    Simulator.exitqueue.clear();
    Simulator.startdate=new Date(start);
}
}

```