SON/RRM Functionality for Mobility Load Balancing in LTE Networks

Salvador Navarro Suria

E.T.S.I. Telecomunicaciones

Universidad Politécnica de Valencia

A thesis sumitted for the degree of

Master in Telecommunications Engineering

September 2012

1. Reviewer: Name

2. Reviewer:

Day of the defense:

Signature from head of the committee:

Abstract

Self-Optimization will be an integral part of LTE systems in future deployments as part of the Self-Organizing Networks (SON) features defined in 3GPP LTE Release 9 and beyond. In this context, Mobility Load Balancing (MLB) is introduced as a SON use case. MLB enables the optimization of the intra-LTE mobility parameters to the current load in the cell and in the adjacent cells to improve the system capacity compared to static/non-optimised cell reselection/handover parameters. The idea of the MLB use case is to enable cells that suffer congestion to transfer the load to other cells, which have spare resource. Such transfer must usually be forced against (optimal) radio conditions and hence new mechanisms need to be addressed.

The MLB use case goes hand-in-hand with handover procedure, admission control, and other RRM functionalities. It is foreseen that the MLB SON functionality will optimize the system performance on a long-term basis as compared to the RRM functionality which adapts the system to network conditions on a millisecond basis.

The thesis shall consider the interaction between SON MLB use case and RRM functionalities in order to optimize the system performance. Briefly, the thesis outline may be given as follows:

- Conducted an extensive literature review about RRM and MLB algorithms, identifying current solutions to the problem, and conducting a feasibility analysis for realizing such solutions in real network deployments,
- Getting familiarized with the multi-cell simulator from the company,
- Helping in developing and improving an MLB-RRM concept (with direct support/contribution from the supervisors).

- Building a simulation framework for analyzing load balancing and implementing a framework (if not already there) for any required information exchange between the eNBs as required by the investigated solutions,
- Implementing practically relevant load balancing algorithms,
- Evaluating the performance in terms of user throughput, system spectral efficiency, and other KPIs, and
- Optimizing the algorithms by adapting the parameters and producing final set of results.

To my Family and Girlfriend, for their unconditional support and love.

Acknowledgements

I would like to acknowledge Abdallah Bou Saleh PhD, for supporting this thesis during the last year. His knowledge has been really helpful on the completion of the thesis and in the discussions while making decisions along the thesis. I also would like to acknowledge Pablo Escalle for speeding up the thesis process and for accepting this Thesis as my final ”Proyecto Final de Carrera” for the Masters degree in Telecommunications Engineering.

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

1

Introduction

This thesis discuss the Mobility Load Balancing functionality described on the 3GPP standard for Long Term Evolution (LTE) Networks. This functionality belongs to the set of functionalities available for self-organizing Networks (SON) and applies its enhancements in the Radio Resource Management (RRM) layer of the LTE protocol stack.

The discussion starts with an introduction to the LTE Radio System and its protocol stack. The properties of each layer are summarized in Chapter 2. The goal of this introduction is to give a general overview of the system and explain some characteristics of the system, as well as concepts, that apply directly to the framework of this thesis. If further details is needed, the bibliography is provided at the end of the thesis.

Chapter 3 will introduce Self-Organizing Networks and discuss the need for automated solutions on the network to enhance the network performance and reduce costs. The present 3GPP standard and special features available for SON will be explained for the self-optimizing solutions and specially for the Mobility Load Balancing capabilities standardized in the different releases.

Chapter 4 will define and discuss in detail the framework necessary to implement the Mobility Load Balancing solution. We will take a look at the mathematical framework developed for MLB and the parameters and assumptions necessary to consider for the solution.

Chapter 5 explains the solution chosen and implemented in this thesis. We will take a look at the details of the MLB algorithms and explain how the decision process takes place and which approximations were made to solve the problems for a real system solution.

Chapter 6 will introduce the simulator (AMoRE) used to evaluate the performance of the MLB algorithm. It will discuss the results obtained for different scenarios pointing out the cases where the MLB solution is appropriate and showing the dependency of the performance to the parameters introduced in the MLB algorithm.

Finally, chapter 7 will give an overview of the results obtained for the MLB algorithm and will propose new paths to research to enhance the performance of the MLB algorithm and the performance of the system.

2

LTE Overview

The objective of this chapter is to present an overview of the UMTS Long Term Evolution (LTE) to have some background information and a better understanding of the whole Mobile Communication system. This overview will help understand some of the concepts introduced in further chapters. The following sections are a collection of notes taken from different books and do not describe the system in depth. For a more detailed description, refer to the books appointed in the bibliography.

2.1 General Description

Long Term evolution is a 3GPP project and the last step in the evolution of mobile communications systems which extends and modifies UMTS systems. Figure 2.1 show a first representation of the LTE architecture and explains graphically the terminology that would be used in the architecture description.

LTE is developed under the assumption that all the services are packet based and therefore the radio access technology, as well as the core network (EPC : Evolved Packet Core), are fully packet-switched (PS) with IP connectivity. Together, LTE and EPC, constitute the Evolved Packet System (EPS) which defines a new network architecture that allows the following enhancements:

- A Simple and Flat architecture that favors an optimized usage of the network and minimizes the number of network elements. Due to the fact that all radio network functionalities are located at the eNodeB, there is no need for extra controllers as in UMTS.

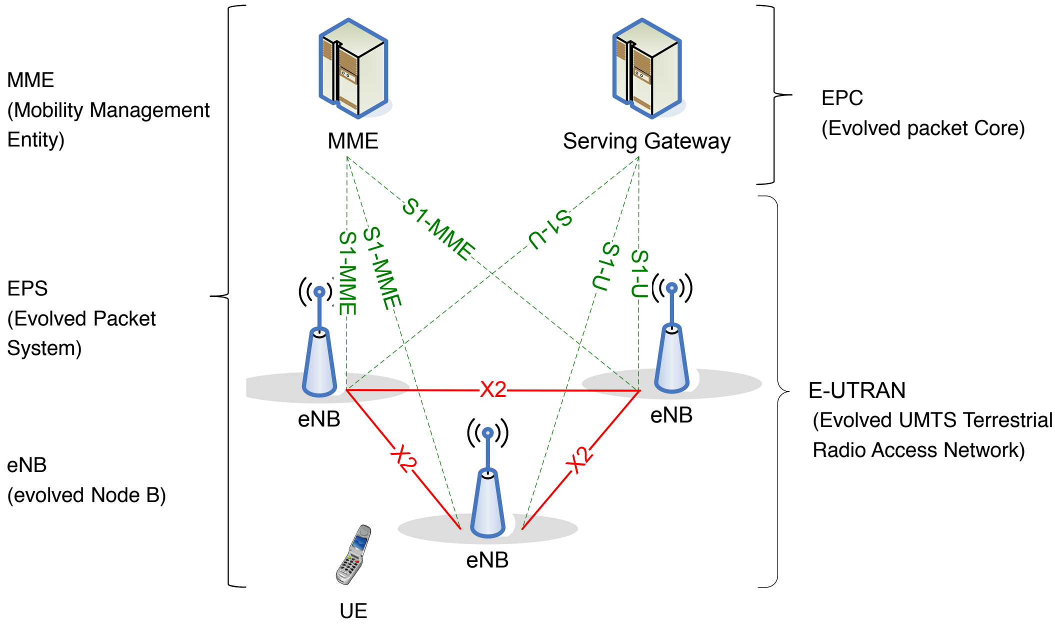


Figure 2.1: LTE Architecture and its terminology - See

- High Data Rate reached due to advanced modulation techniques for optimization of radio frequencies.
- Packet Optimized radio access network
- Enhance User experience for High quality multimedia services by improving cell capacity, end-user throughput and low user plane latency

In the following sections we will be taking a closer look to the EPS system with a brief description of the EPC core network as part of the whole system and we will explain in more detail the E-UTRAN.

2.2 Specifications

The 3GPP (3^{rd} Generation Partnership Project) is a collaborative standardization group formed by several telecommunication associations from different parts of the world and member companies who participate in the standardization process. The 3GPP establish the specifications of the E-UTRAN (Evolved UMTS Terrestrial Radio Access Network), their aim is to guarantee interoperability between multiple vendors, adapt the system to the regulations of the different countries and take into consideration the market needs when defining these specifications. In figure 2.2 and 2.3, we can see the last LTE releases with some of their features.

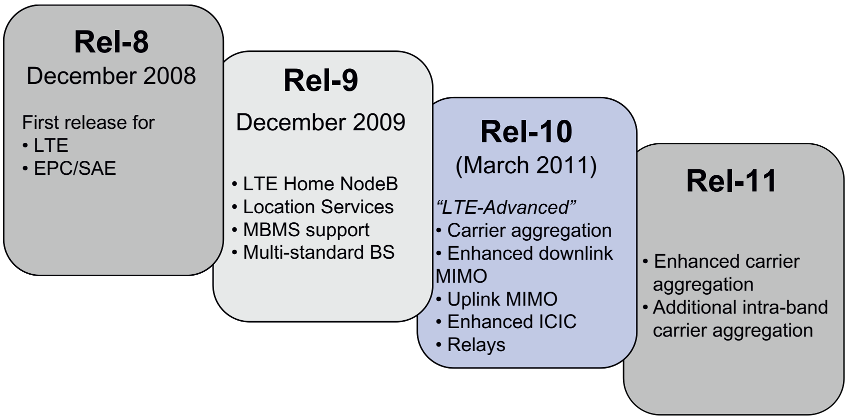


Figure 2.2: Releases of 3GPP specifications for LTE - See (1)

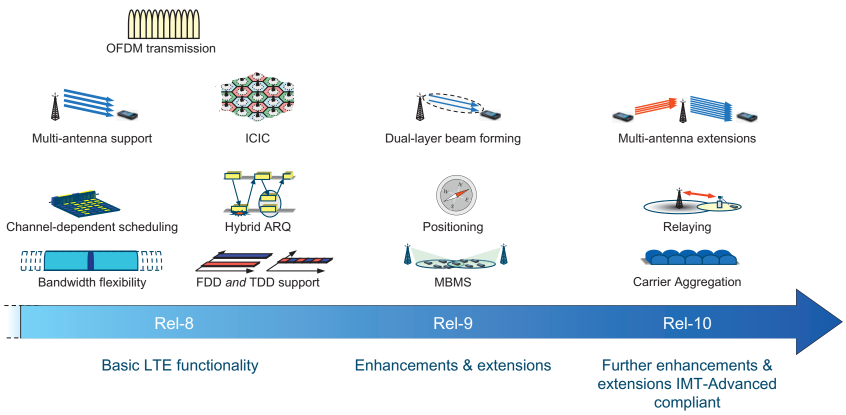


Figure 2.3: LTE and its evolution - See (1)

The first LTE requirements were defined in mid 2005 and the first LTE standardization was finalized in december 2008, know as Release 8. LTE has been enhanced and further standardizations have been made during the last years through releases 9 and 10. At the moment, release 11 is being standardized by the 3GPP. See section 1.5 of (*Dahlman et al.* (1))for further details on 3GPP and the standardization process.

2.3 Overall Architecture

Figure 2.4, takes a closer look at the LTE architecture and the interfaces between the different entities.

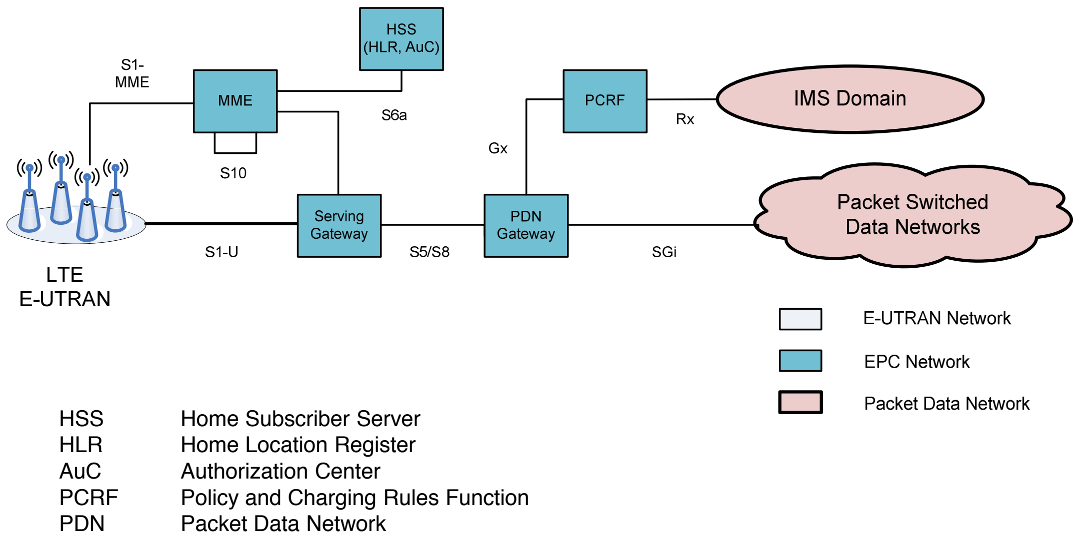


Figure 2.4: Block diagram of LTE Architecture and its evolution -

2.3.1 Evolved Packet Core (EPC)

The EPC as explained in (2), supports only access to the packet switched domain, meaning that it doesnt have support for the circuit switched domain. The EPC contains all functional core network entities . This entities are grouped into control plane entities and user plane entities, that is, MME, HSS and PCRF for the first case and S-GW together with P-GW in the second case . Here is a short explanation of the different entities from the Control and User Plane and some of their more important aspects.

Mobility Management Entity (MME): The MME is the central element of the EPC, it uses a direct logical control plane connection to support the following functions:

- Mobility Management: In Idle and Active mode, UE tracking, MME selection and mobility between 3GPP access networks

- Authentication and security through NAS signaling
- Management of subscription profile and service connectivity
- Packet core Bearer management functions including dedicated bearer establishment

Home Subscriber Server (HSS) : database that stores subscriber data such as User identification, addressing and the user-specific security credentials needed for authentication and ciphering.

Policy and charging rules function (PCRF): responsible for quality-of-service (QoS) handling, Interfaces with the PDN gateway to convey policy decisions to it and charging.

Serving Gateway (S-GW): provides tunneling management between P-GW and eNodeB and switching with some control functions:

- Mobility anchor for inter-3GPP mobility
- Packet routing, forwarding and buffering
- Downlink rate enforcement based on aggregate maximum bit rate (AMBR)

Packet Data Network Gateway (P-GW): is the router that looks to the outside world (Internet). Its main functions are:

- User Equipment IP allocation and routing
- Per user packet filtering
- Charging for UL/DL per UE, per PDN and QoS Class Identifier
- Mobility to non-3GPP RATs.

As shown before in figure 2.1, the S1 interface connects the E-UTRAN with the MME through S1-MME and with the S-GW through the S1-U.

2.3.2 Evolved Universal Terrestrial Radio Access Network (E-UTRAN)

The 3GPP specification (3), gives an overall description of the E-UTRAN. The E-UTRAN is based on a single radio access entity called eNodeB . The eNodeB (eNB) holds all the network functionalities and therefore there is no need for a centralized controller as the RNC (Radio Network Controller) in UMTS. With no separated control entity of the Radio access, the TTI

(Transmission Time Interval) is much shorter than in UMTS allowing a very fast adaptation to the radio environment as well as a very flexible and fast access during handover. This configuration of the radio access permits to have a distributed architecture which reduces the complexity of the whole system. Thus, the Radio Access Network is composed of a mesh of eNodeBs connected to each other through the X2 interface with IP connectivity allowing good scalability of the network, reuse of the backhaul infrastructure and avoidance of single points of failure. The E-UTRAN also define a separation between User Plane and Control Plane which make them independent from each other. This fact influence the latency of the system by lowering it and allows a better scalability.

Figures 2.6 and 2.5 show the differences and similarities between the User Plane and the Control Plane.

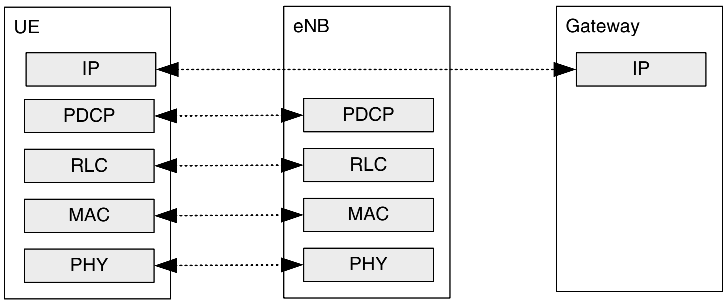


Figure 2.5: E-UTRAN Protocol Architecture - User Plane

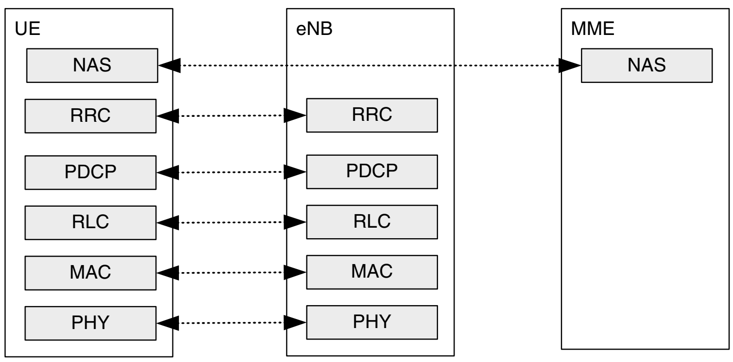


Figure 2.6: E-UTRAN Protocol Architecture - Control Plane

In the following sections, we will be explaining some of the functionalities of the different layers for a better understanding of the E-UTRAN protocol stack.

2.4 Layer 1 : PHY Layer

The E-UTRAN is a very flexible air interface, it supports both frequency division duplex (FDD) and time division duplex (TDD) modes of operation. Thus, most of the design parameters are common to TDD and FDD modes to reduce the complexity of the terminal.

In downlink, LTE uses a new multiple access technology called OFDMA (Orthogonal Frequency Division Multiple Access) which is an extension of OFDM for multi-user communication systems that utilize the spectrum in a more efficient manner than WCDMA in UMTS. OFDM (Orthogonal Frequency Division Multiplexing) is a special case of FDM where the carriers are all made orthogonal with the help of a Fourier transform. This leads to the ability of squeezing subcarrier really tight together. Figure 2.7 shows an example of OFDM subcarrier spacing.

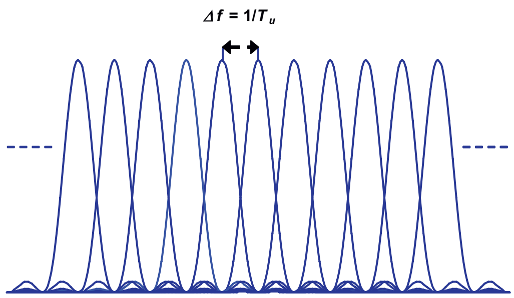


Figure 2.7: OFDM subcarrier spacing -

In LTE, the subcarrier spacing is standardized to $\Delta f = 15$ KHz which is a compromise between the overhead of the CP (Cyclic Prefix), used to reduce ISI (Inter-symbol Interference) due to multi-path propagation, and the sensitivity to frequency offsets due to Doppler spread/shift which produces ICI (Inter-Carrier Interference) breaking the orthogonality of the subcarriers.

As described in (2), the smallest transmission unit are Resource Elements. Each Resource Element contains a symbol of duration $T_{sym} = 66.67\mu s$ transmitted over a single sub-carrier of $\Delta f = 15$ KHz, the number of bits per symbol depend on the modulation scheme used e.g. QPSK, 16QAM or 64QAM. The Downlink and Uplink are divided into Physical Resource Blocks (PRBs), see Figure 2.8, each Resource Blocks (RB) contains 12 consecutive subcarrier and 6 or 7 symbols (depending on the lenght of CP) per subcarrier transmitted in a slot $(0,5$ $ms)$.

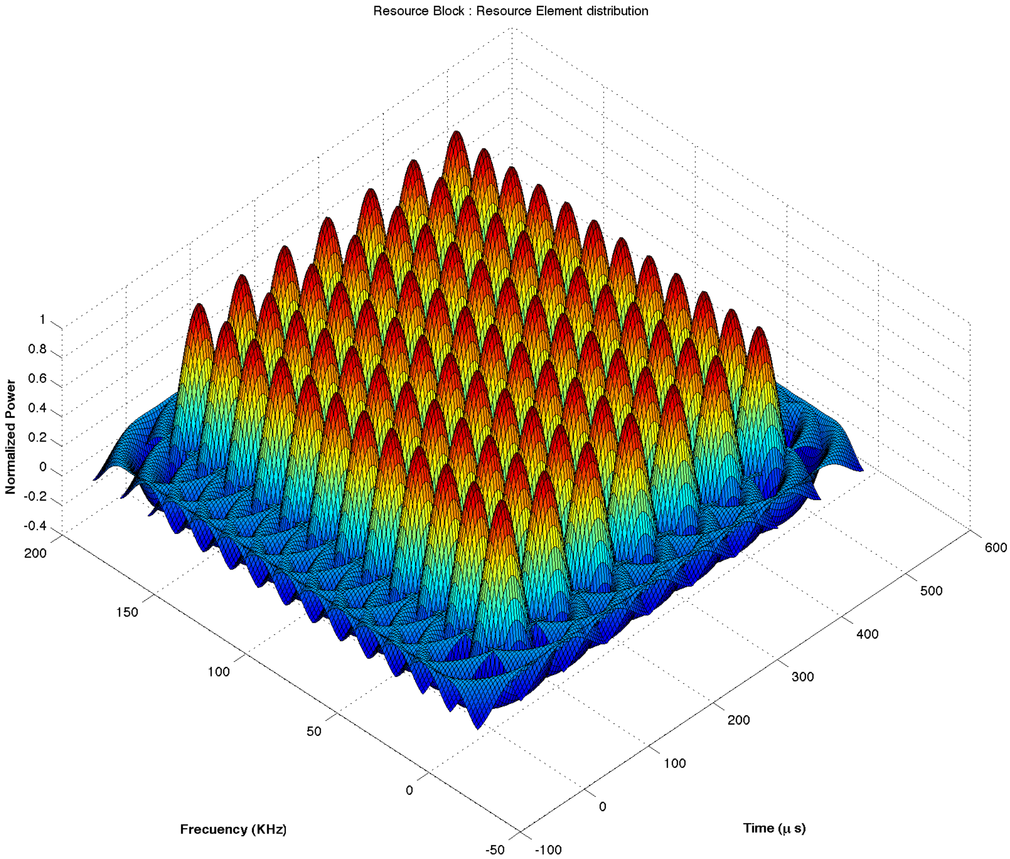


Figure 2.8: Resource Block - Resource Element Distribution

Figure 2.9, explains graphically a subframe, a subframe is the combination of two resource blocks equivalent that occupy 2 slots and a total of 2 RB x (12 subcarriers x 7 symbols), that is 168 Resource Elements. A TTI (Transmission Time Interval) is the smallest scheduling

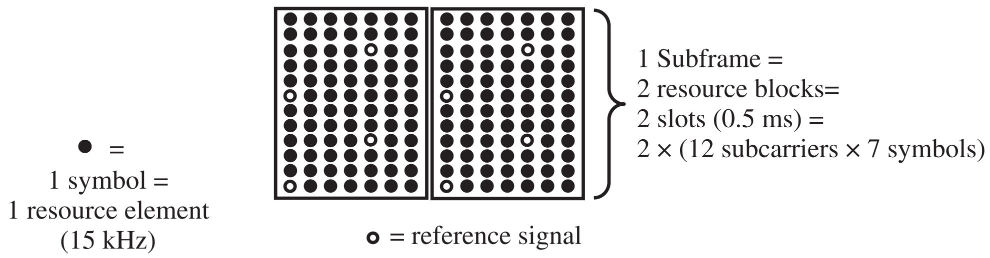


Figure 2.9: Symbols in a Subframe - 2 Resource Blocks in a TTI. See (4)

time in LTE equivalent to 1 ms, equivalent to one subframe or 2 resource blocks. During the scheduling time or TTI the eNodeB decides to which user should be scheduled and which resource blocks are assigned to each one.

Figure 2.10 shows the downlink physical layer design and the assignment to different UEs .

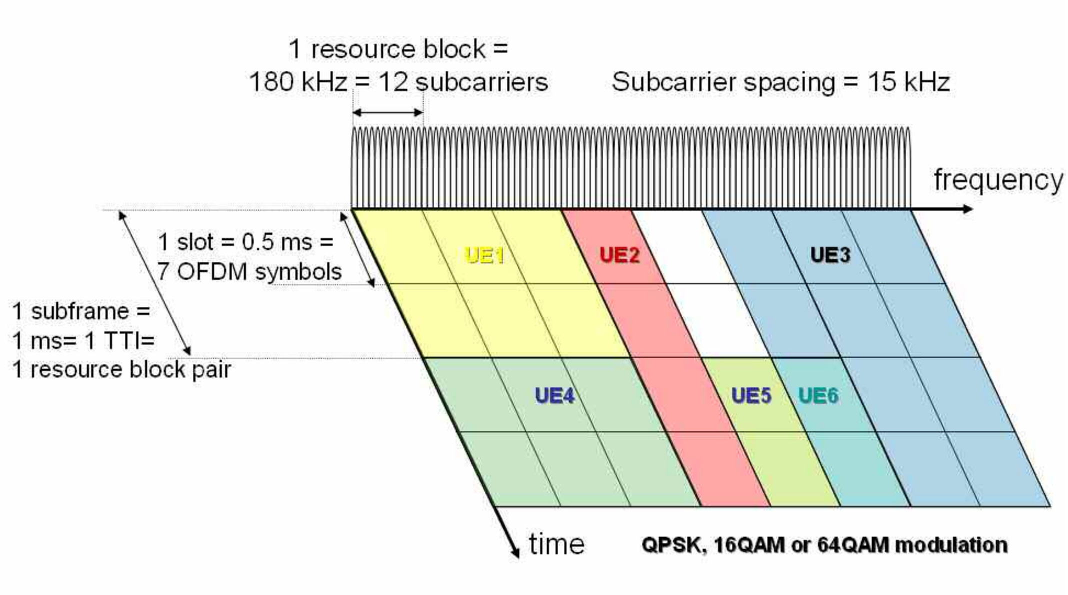


Figure 2.10: OFDMA time frequency multiplexing - See (5)

Since LTE is a very flexible system, the number of PRBs available depends on the Bandwidth of the whole system. As explained in (1), the OFDM falls off very slowly out of the OFDM Bandwidth therefore a 10% guard band is needed. That means that out of a 5 MHz Bandwidth, 4.5 MHz will be actually used for transmission of Resource Blocks. Table 2.1 shows the number of resource blocks depending on the system bandwidth.

Bandwidth	Sub-carriers	Resource Blocks
1.4 MHz	75	6
3 MHz	180	15
5 MHz	300	25
10 MHz	600	50
15 MHz	900	75
20 MHz	1200	100

Table 2.1: Number of Resource Blocks for different System Bandwidth .

In the Uplink, OFMA is not an optimum solution due to the weak peak-to-average power ratio (PAPR) properties of the signal. Instead, SC-FDMA (Single Carrier Frequency Division Multiple Access) with cyclic prefix is used because of its better PAPR properties. It also allows a lower cost of the UE terminals, due to the usage of cheaper power amplifiers, as well as a lower power consumption enlarging the battery life at the UE . The Uplink layer design is the same as in Downlink so the same explanations done before applies also for the uplink case.

From a functional perspective, the role of the PHY Layer is to provide physical channels to the upper RLC and MAC layers. As described in (6), the TTI is a transport channel attribute and can be explicitly given by higher layers through the modulation, the coding scheme and the size of the transport blocks. At each TTI, the physical layer receives a certain number of Transport Blocks for transmission then a CRC (Cyclic Redundancy Check) is added, then protected by a channel-encoding scheme and size adapted by the MAC HARQ process. Moreover, the interleaving process takes place to make it more robust to errors and the Mac layer decides about modulation scheme and finally the data is mapped to the different control or data physical channels.

2.5 Layer 2

Figure 2.11, shows the 3 sublayers contained in Layer 2 and the channel interfaces between each other, Layer 1 and Layer 3:

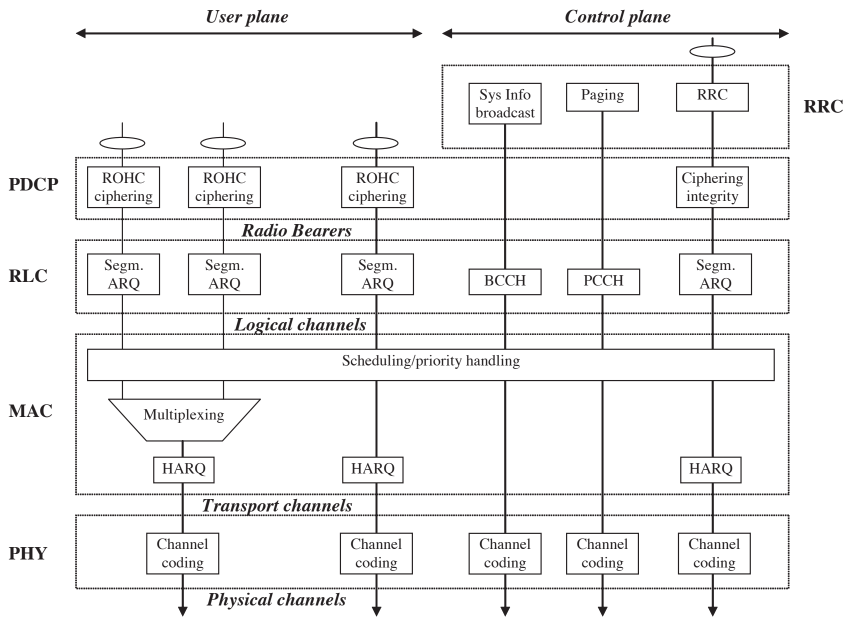


Figure 2.11: Protocol Layer Structure for Downlink - See (6)

The different sublayers and their functions will be explained briefly in the following subsections. For further details on the specifics of each function, please refer to the books cited in the bibliography or the 3GPP specification.(1, 2, 6)

2.5.1 MAC Sublayer

The Medium Access Control (MAC) radio protocol sublayers main purpose is to provide an efficient coupling between RLC services and the physical layer. The main function for the MAC sublayer are:

- Multiplexing of Radio Bearers (Signaling and Data Bearers) : Mapping between Logical channels and transport channels, logical channel identification and transport format selection, reference signals, synchronization signals, broadcast channel and HARQ indicator channel.

- HARQ (Hybrid Automatic Retransmission reQuest): used for Error Correction and allows the network to retransmit faulty packets
- Dynamic Scheduling (UL/DL): decides when, where and what kind of data is scheduled and to which UE the data is sent.
- Priority Handling: between UEs and between logical channels
- QoS Management
- Timing Advance: for synchronization of the mobile transmission
- MAC Control Messaging : PDCCH indicate which resource blocks are allowed to use in the uplink direction (Uplink packet scheduling),
- Power headroom report (UL)
- Buffer Status Report (PUSCH)
- Padding

The MAC layer connects with the RLC over the logical channels, for a better explanation on how the mapping of the logical and transport channels is done, please refer to the 3GPP specification (3) .

2.5.2 RLC Sublayer

The main goal of the Radio Link Control sublayer is to receive and deliver data packet to its peer RLC entity. For that purpose, three different transmission modes are available and assigned to the different logical channels depending on the type of information they carry:

- Transparent Mode (TM): used for general information, it does not alter the upper layer data, no RLC header, just forwards the data.
- Unacknowledged Mode (UM) used for signaling.
- Acknowledge Mode (AM) used for user data.

The main functions of RLC sublayer are performed depending on the transmission mode (UM or AM):

- Segmentation : decides on PDU sizes depending on QoS and available resources
- Automatic Repeat reQuest (ARQ) : ensures the correct delivery of data for AM mode over the air interface.
- Reassembly: (UM and AM) it needs a RLC header in the PDU to know the order of the sequence
- Status Report (AM): indicates if retransmission was lost.

2.5.3 PDCP Sublayer

Figure 2.12, shows the functionality of the Packet Data Convergence Protocol (PDCP) Layer:

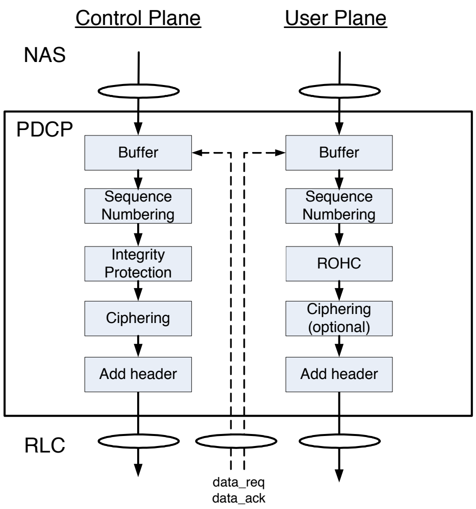


Figure 2.12: PDCP Layer main packet operations -

The main functions of the PDCP Layer are :

- Encapsulation of higher layer protocols
- Packet handling: Buffers packets until they are scheduled by lower layers
- Packet forwarding :Lossless Retransmission of PDCP SDU to support Handover
- Queuing: AQM (Active Queue Management) controls the length of the queues and the delays produce by those queues.

As seen in the figure 2.12, the User plane uses RObust Header Compression (ROHC) and encryption (Data protection) . In the Control plane case , no header compression is needed and ciphering just for dedicated control channels.

2.6 Layer 3

In this section, we describe the Layer 3 protocols of the Control Plane. As seen in figure 2.13, the User Plane does not share this protocols, instead the User Data is directly forward to the PDCP Layer as IP packets.

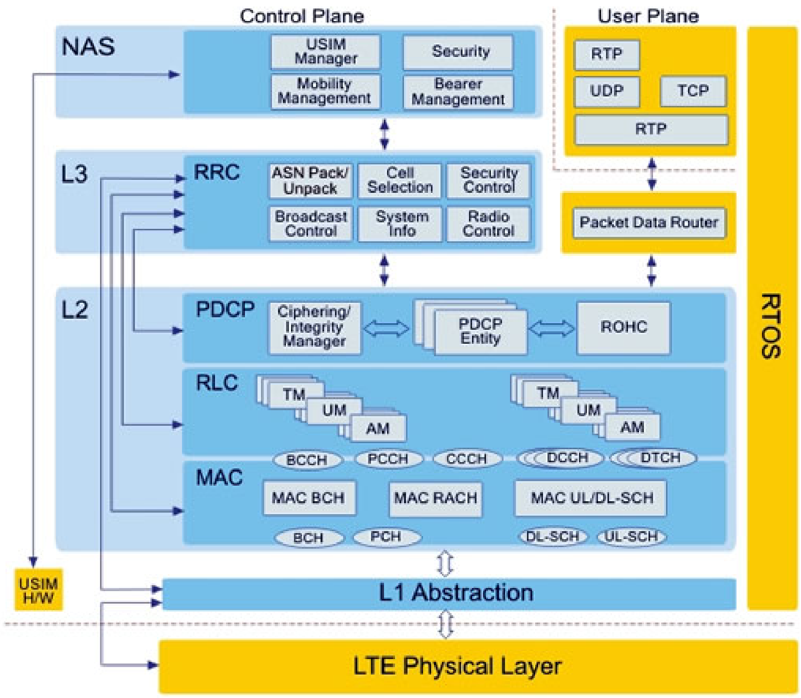


Figure 2.13: Protocol Layer review - Control and User Plane . See (7)

2.6.1 RRC Layer

The Radio Resource Control (RRC), as described in the 3GPP specification (8), is a Layer 3 Access Stratum protocol of the control plane layer that handles the UE management and controls Layer 2 and Layer 1 parameters as well as UE - eNodeB Signaling . Figure 2.14 , shows how the RRC Layer interact with the Lower and Upper layers:

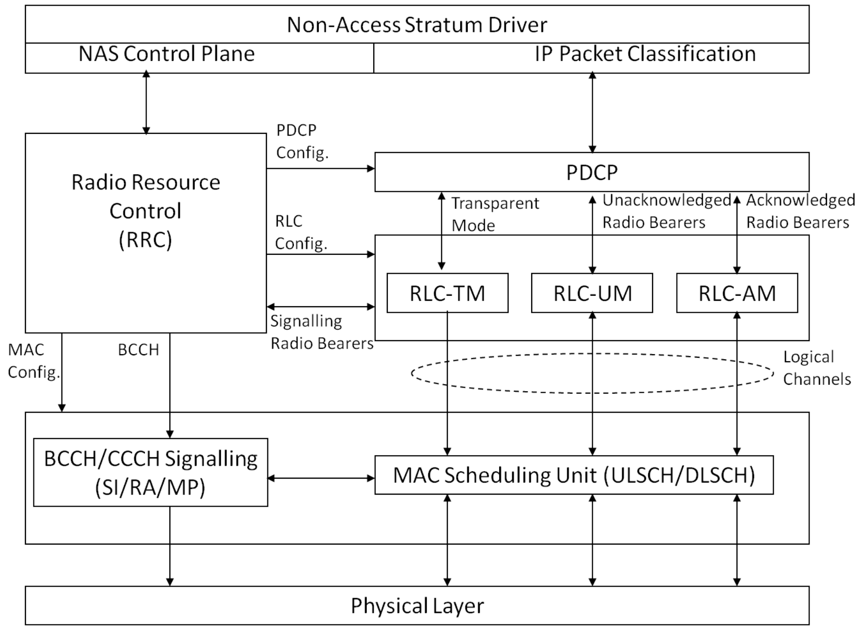


Figure 2.14: RRC control over the different Layer -

The functions of RRC depend on the RRC state of the UE depicted in figure 2.15:

- Applicable to both States:
  - Broadcast of System information (SI): informs the UE about the different configuration parameters necessary to use the transport channels and for mobility purpose
- RRC Idle mode:
  - Paging: allows the UE to detect an incoming call by monitoring the paging channel.
  - UE cell selection and re-selection, controlled by the parameters of the SI.
- RRC Connected mode:
  - RRC Connection management between UE and eNodeB : Radio Resource allocation for the UE and configuration of signaling Radio Bearer (SRB) to send over the control channels .
  - Security functions: such as Key management
  - Quality of service (QoS) management: establishment, maintenance and release of Radio bearers (point to point, MBMS services,..)

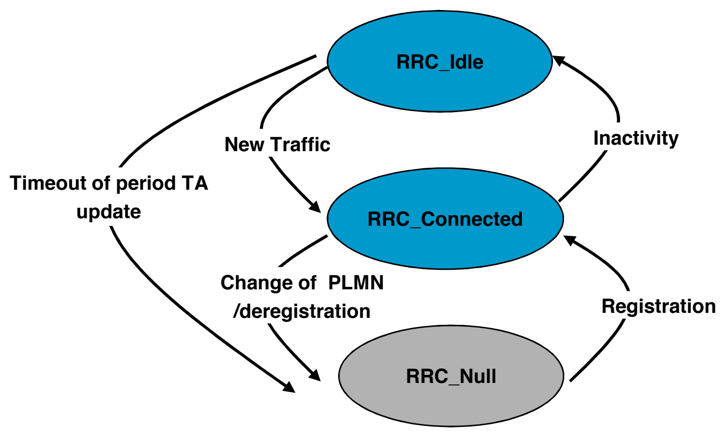


Figure 2.15: RRC States -

- Mobility functions: Handover, inter-cell and inter-RAT mobility and measurements.
- UE measurement reporting configuration and control: buffer status, downlink channel quality, neighboring cell measurements used for mobility procedures support.
- UE context transfer between eNodeB at handover
- Non Access Stratum (NAS) message transfer from/to NAS to/from UE

The RRC messages are mapped into Signaling Radio Bearers (SRB):

- SRB0 : used for connection establishment purpose, it is non-integrity protected and mapped to CCCH (Common Control Channel)
- SRB1: used in RRC connected mode, it is the signaling with higher priority and mapped to DCCH1 (Dedicated Control Channel)
- SRB2: used in RRC connected mode, it is the signaling with lower priority and mapped to DCCH2
- System information (SI) is mapped to BCCH (Broadcast Control CHannel)
- Paging Notifications are mapped to PCCH (Paging Control Channel)

2.6.1.1 System Information

The system information (SI) is common information that is broadcasted over the BCCH channel. The SI is structured into SIBs (System Information Blocks) which contain functionality-related parameters. The SIBs are transmitted over three different types of RRC Messages: MIB, SIB1 and SI messages. The Paging message informs the UE (in Idle mode) of the SI Changes. Table 2.2 shows the different types of messages, their timing requirements and the applicability to both RRC modes.

Message	System Information Blocks Content	Period(ms)	Applicable
MIB	Essential Physical layer Parameters	40 (fixed)	Idle & Conn
SIB1	Cell Access Parameters for Cell reselection	80 (fixed)	Idle & Conn
1^{st}SI	SIB2 : Common & shared channel Configuration	160	Idle & Conn
2^{nd} SI	SIB3: Intra-frequency Serving cell reselection SIB4: Intra-frequency neighboring cell info	320	Idle only

Table 2.2: System Information Blocks: Types and configurations See (2).

The SIB configurations for inter-frequency and inter-RAT (Radio Access Technology) are not included in the previous table. Table 2.3, summarizes some of the parameters available in the system information blocks. For further details and explanation see (7).

Information Block	Key Information
MIB	Downlink Bandwidth, PHICH Configuration SFN (System Frame Number), Number of Transmitting Antennas
SIB 1	SIB Scheduling List, PLMN ID (s), Cell barring, TAC (Tacking Area Code), Cell Selection Parameters, Frequency Bands
SIB 2	Detailed Cell barring, Uplink frequency allocation Uplink Bandwidth, MBSFN details
SIB 3	Cell reselection details
SIB 4	List of Intra-Frequency neighboring cells, $Q_{offset_{s,n}}$ Black List of Intra-Frequency Neighboring cells

Table 2.3: System Information Block Parameters (7).

The RRC Layer handles the messaging for the handover of an UE from one serving cell to another target cell. LTE mobility will be explained in further details due to the fact that the subject of this thesis applies directly to it.

2.6.2 RRM Layer

As explained in (2, 9), the primary goal of the Radio Resource Management (RRM) Layer is to ensure the efficient utilization and optimization of radio resources by using procedures and adaptation techniques for the different Layers. The RRM Layer serves the user according to their minimum QoS requirements to ensure a good user performance. The solutions and algorithms are vendor specific therefore, 3GPP defines just the requirements to support Radio Resource Management such as signaling, QoS requirements and different reporting capabilities.

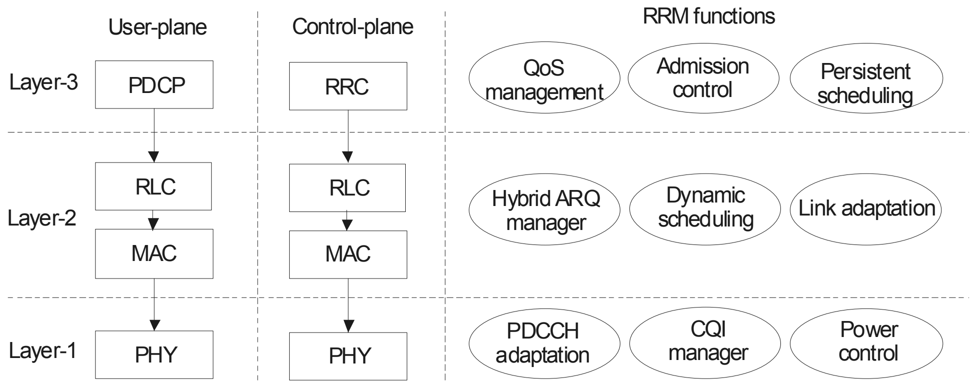


Figure 2.16: Radio Resource Management functions - See (9)

As shown in figure 2.16 ,the main functionalities of RRM are :

- Layer 3: RRC
  - Self-Optimization features: discussed in next chapter
  - RRC Connection management : Establishment, re-establishment and monitoring of connections.
  - Mobility Management :Cell search and reselection, Handover (discuss in next section) for intra-frequency, inter-frequency and inter-RAT.

  - QoS management : to guarantee the minimum QoS requirements of the services required by the users
  - Admission Control: decides on the acceptance of a new Data Bearer from a UE request depending on the available resources and the requested QoS service.
  - Persistent Scheduling (for voice services)
  - Location services

- Layer 2: MAC and RLC
  - Hybrid ARQ management: manages the packet retransmissions.
  - Uplink Adaptation : such as Power control to limit inter-cell interference.
  - Interference management : mechanisms for inter-cell interference coordination (ICIC) such as transmit power adjustments or sending scheduling announcements over the X2 interface.
  - Dynamic Scheduling : allocates PRB in time and frequency to the users depending on the spectral efficiency to maximize cell capacity.
  - Radio link monitoring and adaptation: allows high spectral efficiency by selection of modulation and coding schemes (MCS)

- Layer 1: PHY
  - PDCCH adaptation (Physical Downlink Control CHannel): signalling of the PRB allocation and MCS to the users.
  - Discontinuous reception (DRX) to reduce power consumption
  - Measurement and Report management: CQI reports from the user and Sounding Reference Signals (SRS) for scheduling decisions and Donwlink and Uplink link adaptation, uplink Buffer Status Report (BSR) and uplink Power Headroom (PHR) with their respective measurement configuration.

2.6.2.1 Quality of Service (QoS)

Usually applications and services delivered to a user have different quality of service (QoS) requirements. The eNodeB is responsible of ensuring the minimum requirements of the services to guarantee user satisfaction. For that purpose 3GPP defines different Dedicated Data Bearers types :

- CBR or GBR Bearers (Constant or Guaranteed Bit Rate): defines a type of bearer which guarantees allocation of resources to reach the expected bit rate.
- Non-CBR: which does not guarantee any particular bit rate.

As seen in table 2.4 and defined in 3GPP TS 23.203 , each Bearer has a standardized Quality Class Identifier (QCI) which defines the different priorities, maximum packet delays and packet error loss rate for each required service class.

QCI	Resource type	Priority	Packet delay budget (ms)	Packet error loss rate	Example service
1	GBR	2	100	10^{-2}	Conversational voice
2	GBR	4	150	10^{-3}	Conversational video
3	GBR	5	300	10^{-6}	Non-conversational video
4	GBR	3	50	10^{-3}	Real time gaming
5	Non-GBR	1	100	10^{-6}	IMS signaling
6	Non-GBR	7	100	10^{-3}	Interactive gaming
7	Non-GBR	6	300	10^{-6}	Video
8	Non-GBR	8	300	10^{-6}	TCP based (WWW,e-mail)
9	Non-GBR	9	300	10^{-6}	chat, FTP, p2p file sharing

Table 2.4: Service QCI Characteristics

The QoS influences on the RLC configuration modes and on the MAC Scheduling decisions. Each Data Bearer has a ARP (Allocation and Retention Priority) for admission control to decide whether or not the requested bearer should be establish in case of radio congestion.

2.7 Interfaces

In this section we look at two of the interfaces needed for interconnection of the whole EPS. As said earlier and depicted in figure 2.1 , E-UTRAN is simply a mesh of eNodeBs connected to neighboring eNodeBs with the X2 interface and to the EPC (Evolved Packet Core) through the S1 Interface.

2.7.1 S1 Interface:

The S1 Interface connects the eNodeBs to the EPC over an IP connection using the GTP (GPRS Tunnel protocol) protocol. The S1 interface defines 2 types of connections:

- S1-U: U stand for user plane and the connection is established between the eNodeB and the Serving Gateway(S-GW). This connection uses the GTP-UDP to carry Data Bearers of the users.
- S1-MME: is a control plane connection between the MME and the eNodeB to transmit Signaling Bearers over the SCTP/IP protocol. (Stream Control Transmission Protocol). Its main functions are:
  - SAE Bearer Management (System Architecture Evolution)
  - Paging over S1
  - Mobility over S1:
    - Intra-LTE Handover: just in case there is no X2 interface between eNodeBs. It is similar to UMTS but adding Status transfer to it, just like X2 procedure.
    - Inter-3GPP RAT Handover: mobility towards other RATs (Radio Access Technologies)
  - Load management: controls and prevent overload by balancing the load over different MMEs.
  - NAS (Non-Stratum) signaling transport function
  - Other functions..

2.7.2 X2 Interface:

The X2 interface is a logical point to point interconnection between eNodeBs standardized by 3GPP for multi-vendor operability, see (10). Meaning that there is no dedicated physical connection between eNodeBs which influence the delays of messaging ($5 \sim 20$ ms). This interface uses the X2-AP (Application protocol) based on IP connectivity over SCTP and exchanges application level configuration data. The link between eNodeBs is initialize by the identification of neighbors with the ANRF (Automatic Neighbor Relation Function). Figure 2.17, shows the protocol architecture of the X2 Interface:

- X2-U: transmit data bearers over an unreliable GTP-U transport protocol. This type of connection is used to forward data during handover from one eNodeB to another.
- X2-C: transfers signaling bearers using a reliable SCTP transport protocol. This connection defines 2 types of procedures:

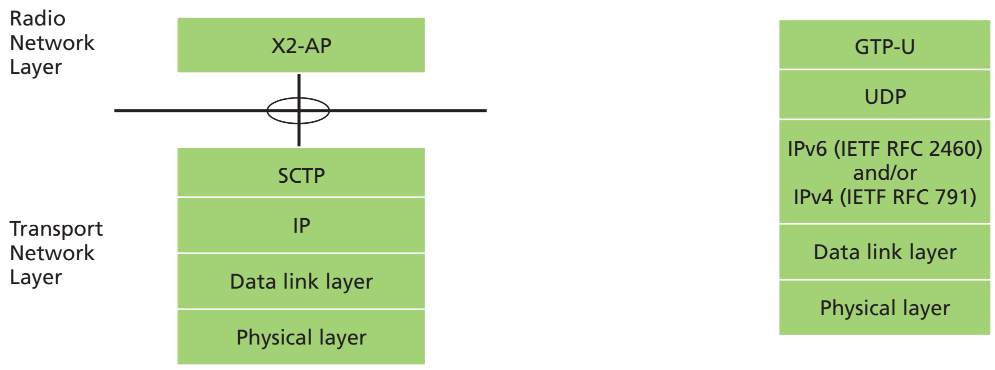


Figure 2.17: X2 Protocol Stack - C-Plane and U-Plane. See (10)

- User specific procedures: mostly mobility procedures to support handovers. Such as:
  - Handover procedure directly performed between two eNodeBs.
  - Handover request to prepare Handover which is the default procedure.
  - Passing historical information of the UEs and the Cell to assist RRM management (e.g. list of visited cells determine to ping-pong) (2)
  - PDCP status report during handover for lossless handover
  - Deletion of context after completion of the procedure
- Global procedures:
  - Setting up X2 interfaces and resetting the link resolving security issues for the exchange of eNodeB configuration data over the link.
  - eNodeB configuration updates
  - Load Management between eNodeBs : that is regular measurements exchange for Load Balancing (as detailed in the latter chapters) and also support of inter-cell interference coordination (ICIC).
  - Error Indication in case error occurs.

2.8 LTE Mobility

In this section we explain in more detail how LTE manages mobility of the UEs. The procedures for maintaining connectivity depend mostly on the UE state, therefore a distinction between idle mode and connected mode will be made along this section. 3GPP aimed to minimize handover delays and disruptions to provide seamless mobility therefore the architecture used for that matter is simple and does not involve management entities unless changes to different RATs or different TA (Tracking Areas) are required. The general description of LTE mobility is presented in (3), idle mode mobility is specified in (11), the performance requirements for radio resource management are defined in (12) and the relevant Radio Resource Control specifications in (8).

2.8.1 Mobility management and User Equipment states:

Mobility procedures are divided into two categories, idle mode and connected mode. The transitions between both states are controlled by the eNodeB.

- UE in Idle Mode : In this state the mobility management is done by the UE, it seeks for the best PLMN and cell to camp on based on parameters provided by the network over the SIBs and its own measurements. Selection and re-selection procedure allows the UE to identify the most appropriated cell or technology for camping.
- UE in Connected Mode: The mobility management is done by the network and it is based on handover. The network controls the mobility decisions based on UE measurements reports from the cells, frequencies and other RAT reachable by the UE. The users satisfaction depends on how these decisions, i.e. finding the best suitable cell, are made and the capabilities of the UE.

2.8.2 Idle mode mobility management: Cell selection and re-selection

As said previously, in idle state, mobility is based on cell selection or re-selection. This procedure is based on finding the strongest cell with quality enough to camp on it. To select the strongest cell the UE needs to measure the different cells that are reachable and suitable based on the S-criterion:

$$S_{rxlev}(dB) > 0 \; and \; S_{qual}(dB) > 0 \tag{2.1}$$

$$S_{rxlev} = Q_{rxlevmeas} - (Q_{rxlevmin} + Q_{rxlevminoffset}) \tag{2.2}$$

$$S_{qual} = Q_{qualmeas} - (Q_{qualmin} + Q_{qualminoffset}) \tag{2.3}$$

Where, $Q_{rxlevmeas}$ is the measured cell received level (RSRP), $Q_{rxlevmin}$ is the minimum required received level [dBm] and $Q_{rxlevminoffset}$ is used when searching for a higher priority PLMN. The same explanation applies to the received signal quality.

The UE retrieves some of those parameters from the SIBs broadcasted over the air interface and then checks the suitability of the cells to make the selection decision based on cell ranking. The following equations describe how the cell ranking is done when the priorities are the same:

$$R_s = Q_{meas,serving} + Q_{hyst,s} \tag{2.4}$$

$$R_n = Q_{meas,neighbor} - Q_{offset_{s,n}} \tag{2.5}$$

where Q_{meas} is the RSRP measurement quantity from either the serving and neighbor cells, $Q_{hyst,s}$ is the power domain hysteresis of the serving to avoid the ping-pong effect and $Q_{offset_{s,n}}$ is an offset value set to control different frequency specific characteristics or cell specific characteristics between the serving and neighboring cell.

Then, the Ranking Algorithm selects the cell:

$$Selected\, Cell = max\{R_s, R_n\} \tag{2.6}$$

Once the UE finds a better candidate from a different tracking area, it de-registers from its actual PLMN and registers to the new one .The Network does not control this process, but can configure different parameters to discourage UEs to camp to a specific cell when overload occurs. Parameters such as frequency prioritization, Neighboring cell list and black listing apply to all camping decisions of the UEs.

2.8.3 Connected mode mobility or Handover:

When an RRC connection exists, the UE is in connected mode . The mobility management in this mode is done by the E-UTRAN by making handover decisions. There are three types of handovers:

- Intra-Frequency Handover: Occurs within the same LTE network Band between different cells.

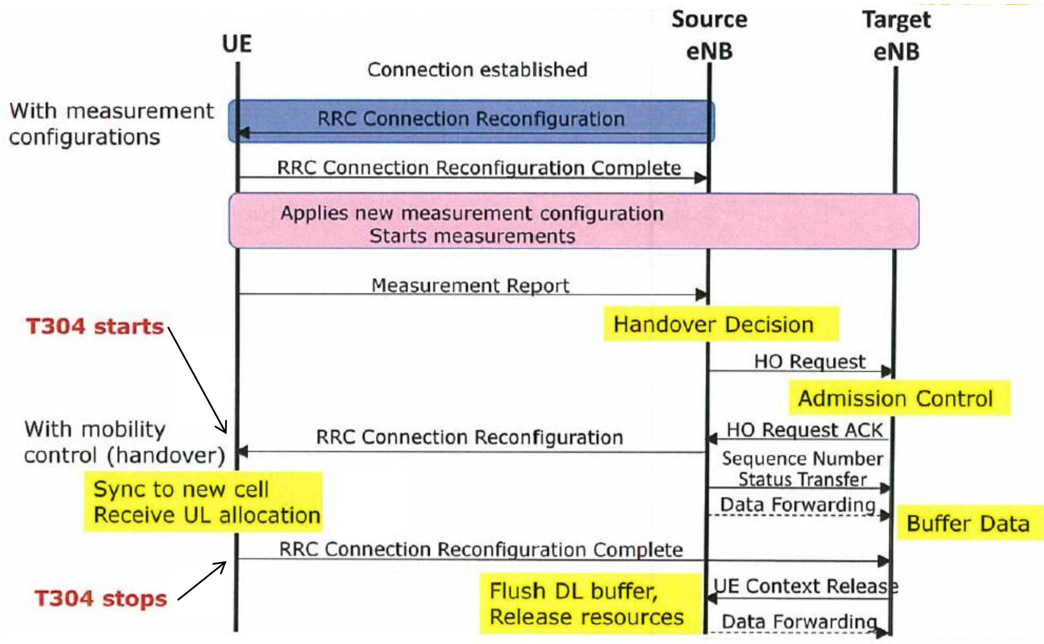
- Inter-Frequency Handover: Occurs between different LTE network Bands to another cell and also vertical Handover within the same cell.

- Inter-RAT : Occurs between different radio access technology (RAT) networks, e.g. WiMAX and LTE, UMTS and LTE, etc.

All have the same basic procedure in common. The UE makes measurements and reports them to the eNodeB. This measurements can be controlled and configured by the eNodeB. Additionally to the UE measurements, the eNodeB makes its own measurements and broadcast information over the corresponding SIBs . Based on all the measurements retrieved and the information available, the handover decisions of a UE to a Target eNodeB is made by the Serving eNodeB . There are two ways to perform handovers, the default and most efficient one based on the X2 interface, and another based on S1 interface when the conditions and configuration require the intervention of the Management Entities.

2.8.3.1 X2 Interface:

As depicted in figure 2.18, the source eNodeB configures the UE measurements with a RRC connection reconfiguration message and the UE responds with a completion message to acknowledge the configuration parameters. While the UE is moving, it measures the different reachable cells. When one of the measurement events is triggered (explained later) the UE sends measurements reports to his serving eNodeB. Based on those measurements and its configurations, the eNodeB decides if the UE should be handed over to a more suitable cell. Once this decision is made, the eNodeB send a Handover request to the target eNodeB. If the target eNodeB supports accepts the handover based on its admission control, it send back to the requested an acknowledgement to the handover request. The Serving eNodeB informs the UE over a Handover command (RRC connection reconfiguration message) to change its serving eNodeB. While the UE synchronize to the new Serving eNodeB, the old eNodeB sends the status information and starts forwarding packets to the new eNodeB who buffers them. When the UE is synchronized it send a completion message to the new eNodeB who sends him back the uplink allocation and timing advanced information. Finally, the new Serving eNodeB send a release context message to the old eNodeB and starts sending the buffered packet to the UE.



T304 expiry = handover failure

Figure 2.18: Handover procedure based on X2 Interface -

2.8.3.2 S1 Interface:

S1 mobility management handles handovers that meet the following conditions :

- No direct connection between neighboring eNodeBs is available, thus no X2 interface has been configured.
- When MME assistance is required to handover the UE such as change in the PLMN, inter-RAT handovers, HeNB handovers or any core-involved handover where the handover procedure is configured to use the S1 interface.

From the UE prospective, the S1 handover does not differ from the default X2 Handover. From the eNodeB side, the request is send to the MME. The MME is responsible for the connection management between the two eNodeBs. Therefore all parameters and handover messaging are managed through the MME over the S1 interface. The procedure is similar to the X2 handover explained earlier. The packet forwarding will be tunneled over the Serving Gateway (S-GW). If the MME changes during handover additional procedures need to be considered. For further details in this proceedings see the bibliography. (1, 2, 9)

3

Self-Organizing Networks (SON)

This section discusses the need for Self-organizing Networks (SON) in the future networks. The process to standardization will be shortly explained, talking about how the need for SON appeared and the evolution of the standardization. Moreover, we discuss the use cases standardized by 3GPP and the challenges faced by the implementation of SON functionalities. Most of the information for this chapter can be found in (13).

3.1 Network Management

Today, Network Management (NM) is mostly done in a centralized OMC (Operation and Maintenance Center) based on a centralized OAM (Operation, Administration and Maintenance) architecture. Meaning that all the control parameters of the network are controlled by a centralized entity. Moreover, the planning and optimization of the networks is managed by semi-automated tools which needs constant human interaction to control the overall performance of the Network. See figure 3.1 A from (14).

With the rapid growth of mobile communications, the deployment and maintenance of mobile networks is more difficult. The complexity of the network increase exponentially as the number of elements increase as well as the interdependencies between their configurations. To be able to maintain a good performance, the network needs to be constantly monitored. This constant monitoring of the network makes the operator to face strong operational challenges in terms of work effort and cost. The work effort applies to the human interaction in the supervision of the management process, which requires highly trained and extensive expertise in the field. Additionally, human interactions tend to increase the response time and the number of errors generated, which impacts directly in the costs of the operators.

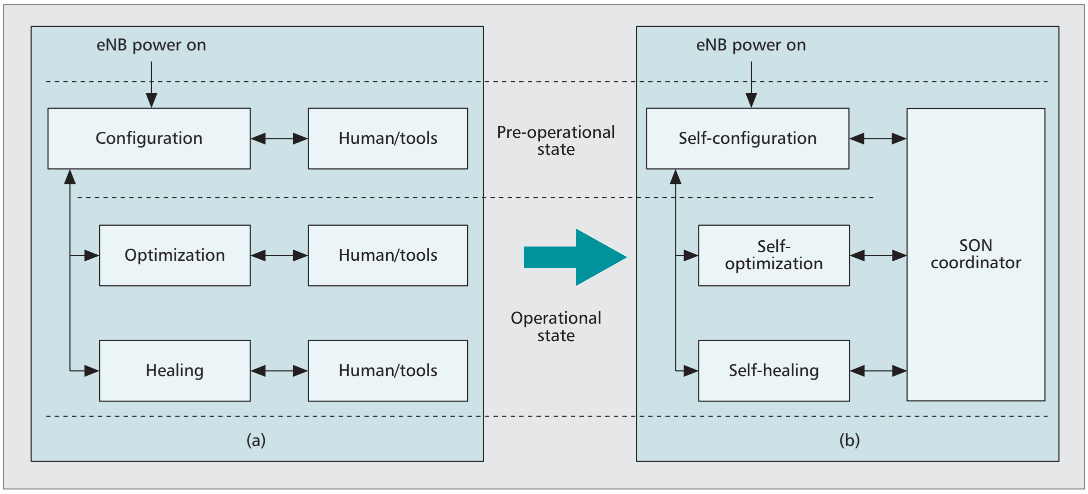


Figure 3.1: Network Operations - A) Today , B) with SON . See(14)

3.2 Self-Organizing Networks (SON)

Self-Organizing Networks aim to reduce complexity of the network by means of increasing automation of the network operations. Making the network more automated will reduce operational effort and work load, as well as constant human interaction. Also, it will protect the network from unexpected errors produced by people and will speed-up the operational processes.

In the early phases, set-up and optimization of the network produce delays that SON tries to minimize, as well as the operational expenditures (OPEX), expenses that will not be compensate by the revenue of the users due to marketing strategies and fierce competition. Thus, cost effective and easy deployment of mobile networks are key factors to shorten the ROI (Return On Investment) of the operators.

Although automation might not be well accepted by network managers, reducing the effort in planning, configuring, optimizing and maintaining multiple network technologies will allow them to shift from low level management tasks to higher level management abstraction. This makes network managers decide about the policies that guide SON functioning based on marketing decisions and leaving the underlaying configurations that need to be applied to the SON functionalities. The configurations will optimize the settings of every individual parameter in each network element resulting in a global enhancement of the performance of the wireless network.

The Self-Organization is divided in 3 parts :

- Self-Configuration: enables automatic configuration for the Initial Deployment and planning , software installation, updates, test, parameter setup, authentication, configurations, neighbor eNB identification and Plug & Play functionality for HeNBs.
- Self-Optimization: automatic real-time control of radio parameters to support the dynamic character of mobile networks, environmental changes, changes on the landscape, changes in user distribution and any other changes that affect the network performance. The optimization decisions depend on the operators preferences and policies. Here are some of the tunable parameters:
  - Radio parameters: handover parameters, neighbor lists, RACH, QoS parameters,..
  - Transport parameters: optimization of S1 and X2 and routing
- Self-Healing: Aims to minimize the harm that failures produce in the network performance. It applies to daily operations of mobile networks, hardware checking for replacement, software updates, network monitoring by measurements and performance analyses, failure recovery and alarm setting for fault detection and triggering of healing mechanisms

3.3 SON Standardization and use cases

In this section we describe how the idea of SON started and its evolution to the 3GPP standardization. We will also give a quick explanation of the different use cases of SON.

3.3.1 NGMN: Next Generation Mobile Networks

As discussed in (13), NGMN is a industry forum created in 2006 and formed by operators. Their objective is to provide business requirements to the new technologies developed. They described the operational use cases, from problems faced by operators in their day-to-day operations, that expected automatic or autonomous solutions to reduce work force and enhance the performance of the network. This use cases as seen in figure 3.2, cover multiple aspects of the network operations, including planning, deployment, optimization and maintenance.

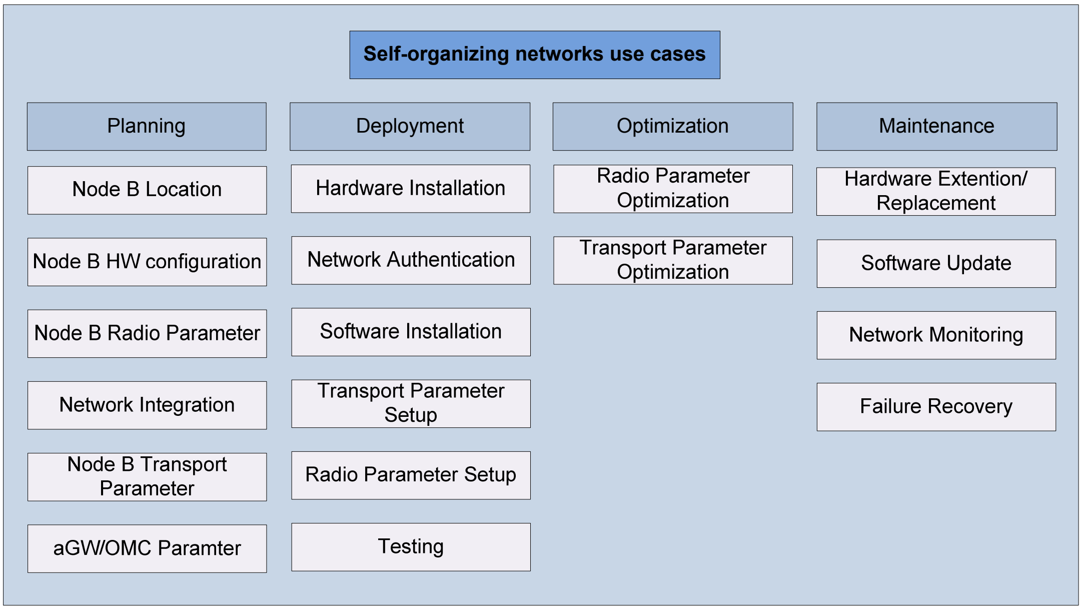


Figure 3.2: NGMN use cases - See (15)

The generation of SON-specific requirements by the NGMN contributed to the adoption of the SON concept by the 3GPP. NGMN defines high level use cases that provide a recommendations and guidance to the 3GPP specifications. Some of the use cases have already been standardized and some of them are in the process.

Some other research projects have influenced the standardization process such as GANDALF, E^3 and SOCRATES which would be refer in the following chapter due to the contribution to Mobility Load Balancing.

3.3.2 3GPP and SON Use cases

In 3GPP, SA5 work group was given the task to study the work items for SON. The goal of this group is to define use cases, measurements, procedures and open interfaces to support interoperability in a multi-vendor environment . This standardization process has been done along Release-8, Release-9, Release-10 and Release-11which still under discussion. Task like standardization of information exchange between network elements such as X2 and Itf-N(open management interface), definition of the use cases or control of SON functionalities by AOM Policies for monitoring and reaching operators targets.

Figure 3.3, shows the evolution of the SON standardization over the different releases.

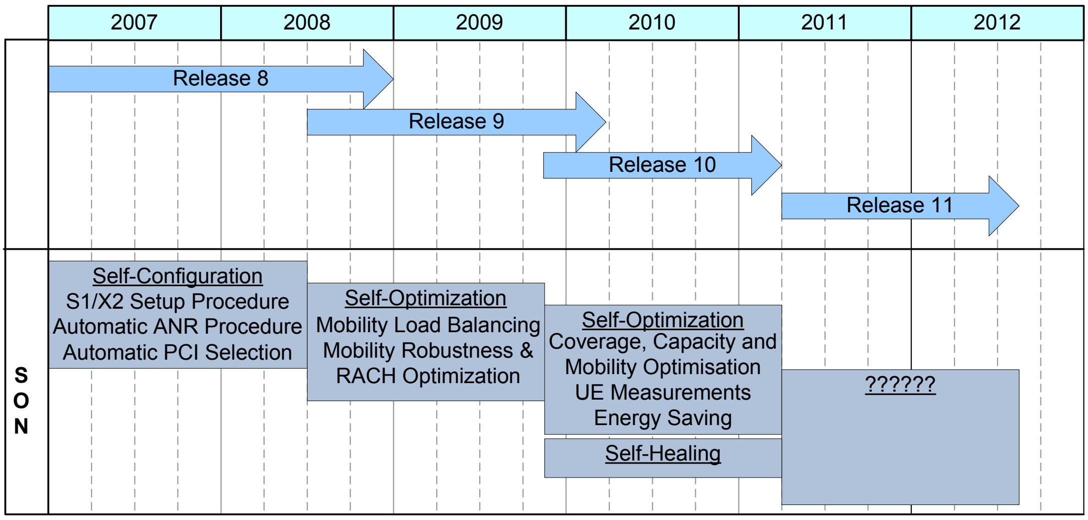


Figure 3.3: 3GPP SON Standardization evolution - See (15)

In 2008, the standardization process of SON functionalities started in the 3GPP with Release 8 specifications introducing Self-Configuration features. Release 8 defines procedures associated with initial equipment installation and integration to support the commercial deployment of LTE networks. The procedures defined are:

- Automatic Neighbor Relation Function (ANR): which enables a cell to maintain information on its neighbors and define operates based on information available at an eNB.
- Automated Configuration of Physical Cell Identity : that enables a cell to select automatically its PCI from an allowed range and avoids PCIs that are reported from outside.
- Load reporting of current load information for radio, Transport Network Layer (TNL) and hardware.
- Dynamic Radio Configuration (DRC) : Automatic configuration of initial radio transmission parameters which allows the base station to become adaptive to the current radio network topology.

- S1 and X2 setup: which allows the eNodeB to automatically setup the connections between MMEs and Neighbor eNodeBs.

In the 3GPP Release 9, RAN3 WG (Work Group) continued work on SON functions described in Release 8 and also newly defined in Release 9. This functions are based on Self-Optimization. Here is a list of the different use cases:

- Mobility load balancing optimization (MLB) which allow network to force users to Handover against the radio condition but due to load situation. Such LB procedure can be executed as an intra-LTE Handover as well as inter RAT HO. In Release 9, for the purposes of Load Balacing, procedures have been defined to negotiate Handover settings between eNBs (intra LTE), and procedures to identify appropriate cause values for Handover request or signalling have been defined. MLB will be further discuss in the following chapters as can be deduced from the topic of this thesis.

- Mobility Robustness Optimization (MRO) which allow on automatic detection and correction of wrong handover settings which lead to Radio Link Failures (RLF).

- RACH Optimization which allows UE to report RACH activity to the eNB . The eNB based on received reports is able to optimize RACH resources and synchronize RACH settings with other eNB by exchanging information over X2 and mitigate interference.

- Coverage and Capacity optimization (CCO) which enables the network to detect capacity and coverage problems (e.g. coverage holes). For the purposes of this function, WG3 defined required information exchanged between eNBs, this task is also related with interference reduction techniques.

In 3GPP Release 10, SON includes the continuation of the work done in Release 9. Enhancements in Mobility Robustness with the focus on inter-RAT Handover and Mobility Load Balancing enhancements to improve intra and inter-RAT procedures. As a new use case Energy saving has been added to SON work item (WI).

3.4 SON Architecture

The SON architecture depends on the location of the optimization functionalities since the algorithms will be influenced by the way data is acquired, the knowledge of the network and the capabilities in each location. As depicted in figure 3.4 , the 3 main architectures are:

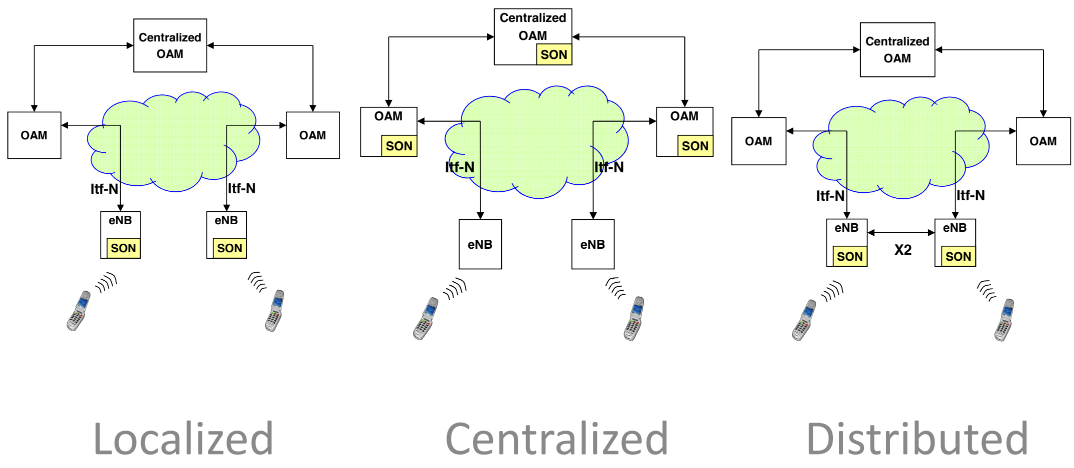


Figure 3.4: SON Architectures - See (16)

- Localized: The functionalities are executed independently from the surrounding eNodeBs. Therefore, SON functionalities are multi-vendor specific and does not require any standardization. This configuration has no interoperability problems nor signaling overhead and the delay is very low. In the other hand, it just applies to isolated problems in single cells.

- Centralized architecture: All optimization algorithms and functionalities executed in OAM systems. Easy to deploy, vendor specific solutions, slow optimization, enormous amount of data. Thus the response time is slow. No need for inter-eNodeB communication but requires It-N interface support.

- Distributed Architecture: All functionality is executed at the eNodeB, costs a lot of deployment effort to support and coordinate lots of eNodeBs thus Convergence and stability might be difficult . In the other hands , fast solutions are possible because the response time in much smaller. The X2 interface should be extended to allow intercommunication between eNodeBs and standardized for multi-vendor compatibility.

Since each one of the different architectures has its advantages and disadvantages, the real SON solution utilize an Hybrid Architecture approach due that the implementation approach depends on the specific SON functionality. Figure 3.5 , depicts the Hybrid architecture and its interfaces (X2 and Itf-N).

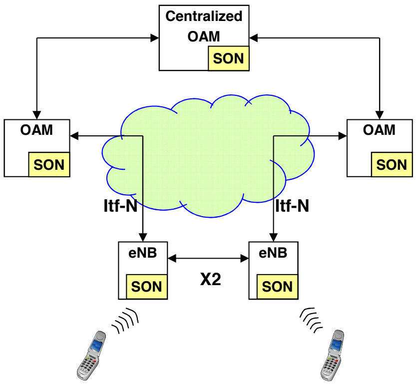


Figure 3.5: Real SON Hybrid Architecture - See (16)

- Hybrid architecture: The functionalities that need faster response will be implemented closer to the eNodeB and the ones that need a better or wider understanding of the network status will be implemented away from the eNodeB. The time delays acceptable, from collecting the information and application of a solution, depend on the requirements of the different functionalities therefore delays are a key factor when deciding about their locations. Then part of the functionalities are executed in the OAM system and part in the eNodeB which allows a lot of flexibility and diversity in the optimization process. It also needs to support multi-vendor optimizations hence standardization of X2 interface information exchange and Itf-N are required.

Figure 3.6, shows the SON architecture approach of Nokia Siemens Networks (NSN) for his SON solutions.

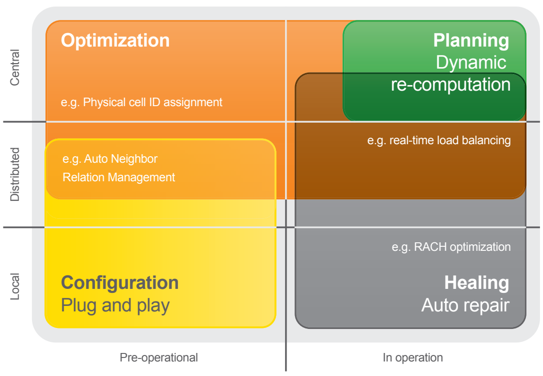


Figure 3.6: Nokia Siemens Network Architecture - See (17)

3.5 SON Challenges

Paper (18) , describes the challenges faced by SON functionalities when implementing solutions for the different use cases. This challenges should be taken into consideration in the design of Algorithms for SON.

- Interrelation of use cases : some of the decisions made by the different SON use cases may generate opposite solutions or may apply to the same parameters, these situations might not get to a desirable solution therefore some coordination of the use cases is needed. Figure 3.7, shows some of this interrelations.

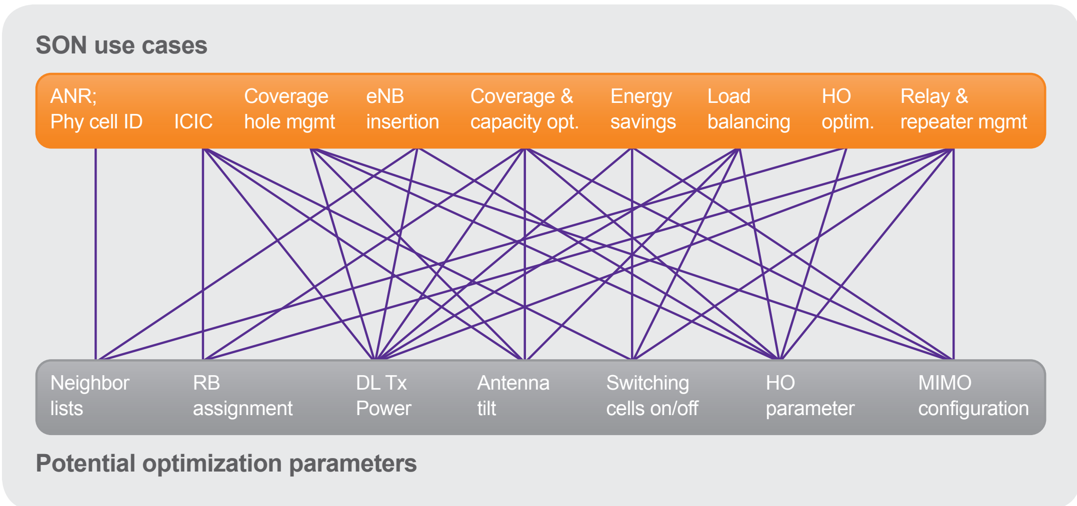


Figure 3.7: SON use cases interrelation - See (17)

- Information availability, consistency and reliability in the network elements would be a major requirement to lead the decisions made in the different use cases, nevertheless a trade-off between delay and overhead needs to be taken into consideration.

- Algorithm design: due to the fact that the information might have delays and error prone. The design of a deterministic solution might not be the best the best approach, a probability approach should be better suited.

- Stability and Convergence: need to be ensure in the solutions especially in dynamic scenarios such as those in the real word

- Evaluation Aspects: evaluation of the performance on the use cases might be difficult to analyze since the use of real networks is prohibitive in terms of costs and most of the information retrieve is confidential. System level simulators also need to make contain many different scenarios and configurations such as user distributions, mobility models, traffic intensities and need to generate their own data to give an estimation of the performance of this use cases.

4

Mobility Load Balancing (MLB)

As explained in (19), the conventional network planning assumes load concentrations of different base stations at the same time. Being able to balance network load to get a suitable network performance is one of the goals of network planning. Therefore, higher density of cells is introduced in areas where higher traffic is expected.

Even with detailed network planning , dynamic random changes of the load over time cannot be taken into account. Higher traffic in office locations during work hours versus higher traffic in residential areas during the evening or sudden affluence of traffic in certain transportation facilities after a special event in the surroundings produce common resource shortage.

The dynamic adjustment of network parameters is considered to be out of the scope of network operations. In this scenarios, Load balancing plays an important role to balance the load between neighbor cells and a utilize resources in a more efficient way. Thus, the ability to Dynamically optimize parameters should lead to a more economic network design and probably saving some base stations.

4.1 Definition

Mobility Load Balancing (MLB) is part of the Self-Optimization functionalities of Self-Organizing Networks. The objective of Mobility Load Balancing, is to counteract local traffic load imbalance when an overload situation appears due to high concentration of users in a specific cell. Due that the spatial distribution of users and traffic properties determine the network load in the cell, MLB aims to optimally distribute the users between the underloaded neighboring cells according to the load conditions of the network and the velocity, Quality of Service or Energy consumption of the user.

Moreover, MLB moves traffic from high loaded cells to less loaded neighbors as far as interference and coverage situations allow in a certain geographical area by means of controlling mobility parameters and configurations including UE thresholds. As depicted in figure 4.1 , MLB changes the mobility parameters, by applying an offset to them. The effect produced by this offset is that, if the user is connected to cell A and he remeasures the received power from the different neighbor eNodeBs, then the best suitable cell might have changed for the user if he is close enough to the border of the cell, as happens for one of the users in the figure.

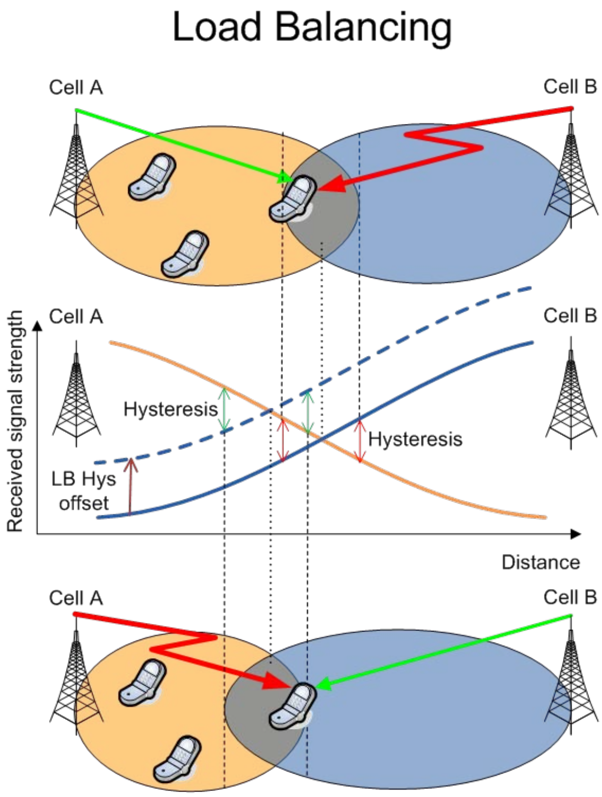


Figure 4.1: Load Balancing Solution - See (20)

MLB solutions should improve QoS, accessability, and resource utilization within the whole network rather than only on a simple base of neighbor cell relations.Thus, improving the overall system capacity and reduce the congestion in overloaded cells.

4.2 Operational Modes in Mobility Load Balancing

As described in (13), MLB can be subdivided into modes of operations, which propose different solutions to overcome the overload problem in a specific cell. Depending on the state of the UE, the different modes of operation are:

- **Idle mode**: The main purpose is to prevent the worsening of the overload situation by adjusting cell specific parameters that impact the camping decisions of the UEs in cell selection/reselection procedures by biasing, prioritizing or restricting access to certain RATs. The UE normally tries to connect to the cell where he is camping whenever it changes to connected mode. Thus, allows optimization of camping decisions to the most suitable cell, depending on the network status.

- **Redirection during connection**: also know as Traffic steering, is another prevention technique that impacts the connection establishment procedure via rejection and redirection of traffic to different Layer/RAT. It can also offload traffic to lower power cells (Wifi, HeNBs) due to unavailability of resources or while balancing the load in overloaded scenarios. It also prevents hard-handovers and lowers the amount of signaling required for load balancing.

- **Connected mode**: In this mode, MLB reaches the load balanced situation by shifting UEs from the border of the cell to less congested cells. This impacts the handover procedures through biasing of cell specific offset and forced handovers. Here, the signaling and setup time are increased.

In this thesis, we focus on load balancing in connected mode since during that state the eNodeB has full control over the users.

4.3 Load balancing mechanism in connected mode

Load balancing in connected is based on changing handover parameters and forcing handovers of edge user to less loaded cells. Depending on the type of handover made and the handover parameters being changed, we can distinguish three different kinds of Mobility Load Balancing solutions:

- **Intra-frequency**: in this case, the handover is made on the same system bandwidth as its neighbor and they need to be forced against radio conditions. This will limit the area where MLB is feasible between 2 cells in a certain frequency.

- **Inter-frequency**: here, the forced handover is produced between different frequencies of the LTE network if available. The advantage is that there is no mutual interference problems between overloaded frequencies and underloaded frequencies. The down-side is that the UEs need to be configured to report other frequencies then the one using, that means that the eNodeB has to signal the configurations and the UE needs to measure different frequencies which cost lots of effort to the UE.

- **Inter-RAT**: Similar to inter-frequency, but handovers happen between different Radio access technologies. This allows also vertical handover in the same cell between different overlapping technologies. The problem in this case is that the old technologies use different methodologies for handover and measurements, and do not possess the information exchange mechanism available in LTE.

We will mainly focus on intra-frequency Mobility Load Balancing on this thesis and we will try to manage the interference problems that happen mostly on the uplink due to the use of frequency reuse 1 usually utilized in LTE networks.

4.4 3GPP Standardization evolution of Mobility Load Balancing

As explained in the previous chapter, the evolution of MLB has gone along with different releases. Here we retake this evolution to focus on the features that apply directly to MLB and the ones that indirectly help MLB achieve his optimization process.

In release 8, as we said earlier, ANR is necessary to maintain track of the neighbor eNodeBs and with the automated X2 Setup, the links between eNodeB is easily established. With the X2 interface established, information can be transmitted to the different neighbors allowing information exchange between them such as load reporting of current load information, TNL or Hardware Load.

In release 9, new features were added to support more accurate MLB solutions. New parameters such as Composite Available Capacity (CAC), basic inter-RAT load information exchange, as well as new procedures to Negotiate information and settings information (explained later).

Release 10, continued enhancing the standard to support intra- and inter-RAT procedures. It also introduced some new features related to energy saving and coordination of the different SON functionalities. As well, new parameters to the X2 messages where introduced for exchanging information to support Uplink MLB, those parameters included P0 and the alpha parameter for path loss compensation.

Finally, release 11 is focused on detection and prevention of intra-LTE rapid Handovers and inter-RAT too late handovers. To optimize those handover problems, information and parameter exchange between different RATs is being standardized.

4.5 Relevant measurements for Load Balancing

While planning the network, the configuration parameters are set in a sub-optimal manner. This is due to the fact that they are set based on models and not on real measurements. Therefore, some optimizations are needed to improve the performance of those configurations.

For that purpose, we need to have measurements from the real networks that allow us to fine tune the pre-configured parameters in a more accurate manner and that help us reveal problems that might happen during the operational phase.

In this section, we describe some of the measurements, as in (21) and (22), needed to support MLB and reach a certain level of accuracy in the decisions made by the algorithms.

- **Reference Signal Received Power (RSRP)**: part of the UE physical layer measurements and is the linear average (in watts) of the downlink reference signals (RS) across the channel bandwidth. It measures the coverage of the LTE cell in the Downlink. It is used to determine the best cell on the Downlink radio interface and select this cell as the serving cell. It is also used for inter-LTE handovers.

$$RSRP_{c,ue}(dB) = P_c - L_{ue} - L_{fad} \tag{4.1}$$

- **Reference Signal Received Quality (RSRQ)**: gives an indication of the signal quality and determines the best cell for LTE radio connection in a certain geographic area. Mostly used for initial cell re-selection and handover.

$$RSRQ(dB) = 10 \cdot \log\left(\frac{RSRP}{RSSI}\right) \tag{4.2}$$

The Received Signal Strength Indicator (RSSI) represents the entire received power including the wanted power from the serving cell as well as all co-channel power and other sources of noise.

- **Total PRB usage (DL/UL)**: Average value calculated from the statistic measurements, in time and frequency, of the usage of PRBs resources in the scheduler, by all the users (MAC Layer). This information is signaled over X2 interface between eNodeBs.

$$PRB_{usage}^{tot}(\%) = \left\lfloor \frac{PRB_{usage}(T)}{N_{PRB}^{tot}(T)} * 100 \right\rfloor \tag{4.3}$$

Where, $PRB_{usage}(T)$ is the PRB usage over a period of time T and $N_{PRB}^{tot}(T)$ is the total number of PRBs available over the same period of time.

- **Composite Available Capacity (CAC)**: indicates the amount of overall resources that the reporting node is ready to accept.“ Composite ” means that the reporting node takes into consideration multiple internal resources criteria, via a proprietary evaluation, to build up his report. “Available” estimate of the amount of non-GBR traffic that can be handed over into the cell controlled by the eNodeB. The load transferred should not exceed the capacity reported as available by the TeNB.

  CAC can have different formats since the calculations are vendor specific. Here are two possible options:

  1. Simple percentage of the total E-UTRAN resources available (total cell uplink or downlink bandwidth known from the X2 setup procedure).
  2. A percentage weighted according to a cell capacity class value which classifies the cell capacity with respect to the other cell available in the network.

  It is assumed that the node indicating available capacity is ready to accept the corresponding traffic, but it is not mandatory. It is also assumed that the algorithm to calculate the available capacity indicator is vendor-specific and runs in the eNodeB that provides the indication.

4.6 Mathematical Framework

In this section, we describe the mathematical framework in which load balancing is based and which was proposed in (23). The goal of this framework is to define the target functions that we would like to optimize and that allows us to judge that a solution is better than another one.

The target functions discuss in the following are Capacity-based target functions and they are based on the signal-to-interference and noise ratio of the users.

The definition of the SINR per user in a cell is as follows:

$$SINR_u = \frac{P_{serving} \cdot L_{serv}(\vec{q_u})}{N + \sum\limits_{c \neq Serv} \rho_c \cdot P_c \cdot L_c(\vec{q_u})} \tag{4.4}$$

Where, $\vec{q_u}$ is the positions of the user, $P_{serving}$ the power of the serving cell, $L_{serv,c}$ the overall loss of either the serving cell or the other neighboring cells, N is the Thermal noise and ρ_c the load of cell c that is a value between [0,1] that represents the probability of a cell using the same PRB as the user.

The definition of the cell load depends on the type of UE services required, the scheduler design and the SINR distribution over the cell. Therefore, if the user has a full buffer type of service, the load of the cell will always be ρ_c=1 , since there is no limit to the amount of traffic required by the user.

If the user requires a Constant Bit Rate (CBR) type of service, also called CBR users, the cell load is defined as:

$$\rho_c = \frac{\sum\limits_{u=1}^{u_T} N_u}{N_{tot}} \tag{4.5}$$

Where N_u is the number of resources (PRBs) used by user u, N_{tot} is the total number of resources available and u_T the total number of users in the cell.

If we take a closer look at equations 4.5, we can see that it assumes that all the services of the users are satisfied by the cell and that no user needs more resources that the ones he is getting. Therefore, its value can not be bigger than 1 . The truth is that, in some scenarios, as in Load Balancing, the cell might be overloaded and may not be able to satisfied the services required by all the users.

For that purpose, we define the amount of required resources by a user u to satisfy his service requirement given a $SINR_u$. That is:

$$\hat{N_u} = \frac{D_u}{R(SINR_u)} \tag{4.6}$$

Where D_u is the data rate requirement of the user u depending on his CBR service and $R(SINR_u)$ is the mapping of the data rate reached by a single PRB given an $SINR_u$. This mapping depends on the MCS (modulation coding scheme), the code rate and the $SINR_u$ as just appointed. Figure 4.2, show a data rate mapping example.

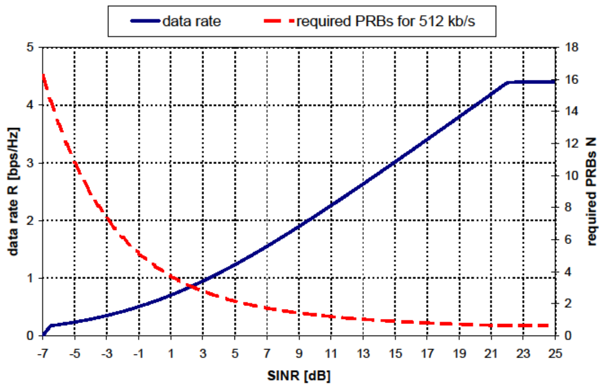


Figure 4.2: Required PRBs for 512kbps for a given SINR - See (20)

Following the idea in equation 4.6, we can define a new expression that will give us an idea of the amount of load that would be necessary to fulfill the service requirements of all the users in the cell if some users are not satisfied. This expression is called *virtual load* $(\hat{\rho}_c)$ and it is define as follows:

$$\hat{\rho}_c = \frac{\sum_{u=1}^{u_T} \hat{N}_u}{N_{tot}} \tag{4.7}$$

Since the *virtual load* $(\hat{\rho}_c)$ can be higher than 1, it gives a better indication of how overloaded the cell is and if $\hat{\rho}_c \leq 1$ then it means that all the users are satisfied.

Actually, from the virtual load definition we can also calculate the number of unsatisfied user in the cell c as :

$$Z = \max\big(0, N_c \cdot (1 - \frac{1}{\hat{\rho}_c})\big) \tag{4.8}$$

Where N_c is the total number of users in the cell and $\hat{\rho}_c$ the virtual load in cell c.

4.7 Framework for Intra-frequency Load Balancing

In figure 4.3, we can take a look at the different features needed to implement the MLB Algorithms into the standard LTE specification specified in (24).

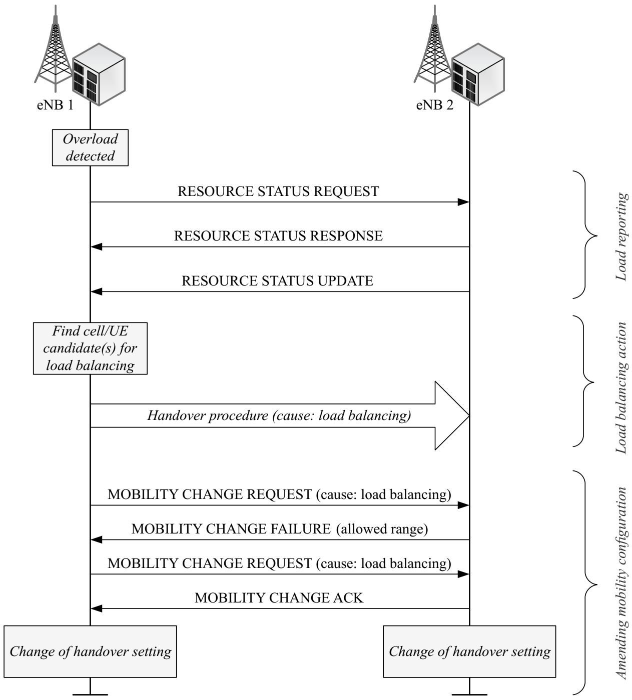


Figure 4.3: Load Balancing Standardized Framework Solution - See (13)

In this section, we will be describing the parts that constitute the overload detection, collection of information, application of the MLB Algorithms and how their output is negotiated between the corresponding eNodeBs.

4.7.1 Overload Detection

As explained in (25) , the first step on the Load Balancing Framework is to detect the overload in the cell. For that purpose, the eNodeB has to monitor the load in the controlled cell and exchanges related information over X2 or S1 with neighboring eNodeBs. This cell status information is normally periodically exchange but to avoid this continuous exchange of information we define some load threshold to reduce the control information overhead of the X2 interface and to trigger the Load Balancing functionalities. Figure 4.4 , shows the different threshold that are defined for Mobility Load Balancing and a imaginary virtual load of the cell.

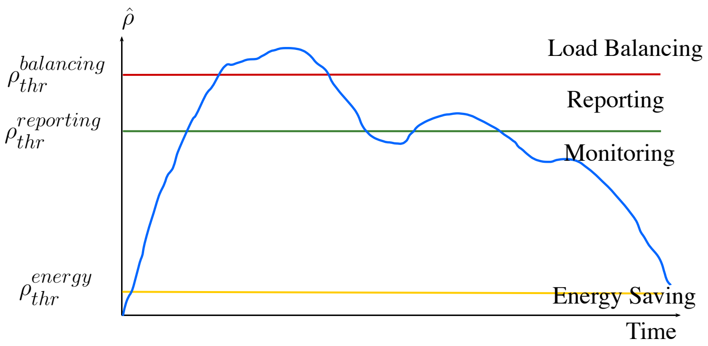


Figure 4.4: Load Balancing Thresholds - See(25)

$\rho_{thr}^{reporting}$ is the reporting threshold, while the virtual load is lower than this threshold the eNodeB does not send his cell load status unless he is been requested by another eNodeB. If the virtual load exceed the reporting threshold, the eNodeB requests the cell load status from other eNodeBs and send his own information to them. $\rho_{thr}^{balancing}$ is the Load Balancing threshold. When the virtual load is higher than this thresholds, the eNodeB triggers the Load Balancing functionalities and the MLB Algorithms are launched. ρ_{thr}^{energy} is the threshold define to trigger Energy saving mechanism for reduction of power consumption when there is a high probability of " long time low load". In this thesis, we will not consider the energy saving case.

4.7.2 Load Information exchange for load balancing

As proposed in (24), the load information of the eNodeBs should be used for load balancing purposes. Besides its own load, an eNB must know the load in the neighboring cells to be able to decide on the appropriate candidate cell for LB actions. The load of the different neighbor can be provided with a Load Information exchange based on a client-server mechanism over the X2 interface.

As depicted in figure 4.3, the requesting eNodeB (client) send a "RESOURCE STATUS REQUEST" to request load reports from his neighbors (servers). This message can request multiple types of measurements within one message. The neighbors that receive the request report the requested load measurements information via the "RESOURCE STATUS RESPONSE" message or the "RESOURCE STATUS UPDATE" message if the reporting is configured to be periodically send to the requesting eNodeB.

The load information exchange carried for Intra-LTE load balancing is as follows:

- the current radio resource usage (UL / DL GBR PRB usage, UL/DL non-GBR PRB usage, UL/DL total PRB usage)
- the current hardware load indicator (UL/DL HW load: low, mid, high, overload),
- the current S1 Transport load indicator (UL/DL TNL load: low, mid, high, overload).
- a composite available capacity indicator (UL / DL) as explained in section 4.5 .
- a cell capacity class indicator (UL / DL).

This measurements give the requesting eNodeB a global view of the current load situation and the willingness of his neighbors to accept extra traffic to facilitate the MLB procedures.The signalling of composite available capacity impacts the X2AP protocol and the necessary support is introduced in the Resource Status Reporting procedures.

4.7.3 MLB Decisions

Once identified the need to distribute the load of the cell towards either adjacent or co-located cells, and having the information from the neighboring cells, load balancing actions can take place. The LB actions are based on algorithms that decide how to distribute the UEs in order to balance the traffic load. This can be done by comparing the load among the cells, the type of ongoing services or the cell configurations. The goal pursued by MLB algorithms may be achieved by delaying or advancing the handing over of the UEs between cells or by changing the mobility parameters between neighboring cells. In any case, the Load Balancing solutions proposed should improve QoS, accessibility, and resource utilization within the whole network rather than only in a single base of neighbor cell relations.

The solutions proposed in this thesis will be described in the following chapter.

4.7.4 Negotiation of mobility/handover parameters: Mobility change Procedure

As explained before, in the intra-LTE case, the MLB algorithm estimates if the Handover parameter settings need to be modified. If the algorithm decides that the Handover parameter need to be changed, communication between the involved eNodeBs needs to take place so the requesting eNodeB can propose changes of the Handover trigger settings to a neighbor eNodeB. In the X2AP protocol, it is defined a Mobility Settings Change procedure to allow the negotiation of Handover trigger thresholds and guarantees that users offloaded to neighboring cells will not be sent back due to radio handover conditions. This means that the ping pong effect can be avoided with this parameter negotiation.

The Mobility Settings Change procedure enables one eNodeB to send a “Mobility CHANGE REQUEST” message to another eNodeB, as depicted in figure 4.3, to indicate the handover parameter shift that his MLB algorithm considers would be more suitable to overcome the overload situation. If the neighbor eNodeB accepts the proposed handover trigger parameter modification then he sends back a “MOBILITY CHANGE ACKNOWLEDGE” message. In case the neighbor eNodeB finds any constraints to the proposed change, it would send a “MOBILITY CHANGE FAILURE” message rejecting the proposed changes and indicating an agreeable range to the parameter modification.

The handover trigger thresholds changes values can be configured in +- 0.5dB delta steps between the new value and the current handover trigger. As specified in (3), the OAM (Operation and Maintenance) entity might also preconfigure the allowed range for the handover parameter changes.

4.7.5 Autonomous Adjustment

Finally, after the network adjusts its handover thresholds with its neighbors to overcome the overload situation, the eNodeB starts the handover procedures of the UEs that fulfill the requirements with the new settings. Cell re-selection configuration may be also adjusted to reflect changes in the Handover settings.

5

MLB Algorithm Implementation

In this Chapter, we explain the implementation chosen for the MLB algorithm solution. The goal is to go into the implementation details to have a better understanding on how the decisions are made and how some of the difficulties are surpassed.

5.1 Algorithm Implementation Milestones

As studied before, the MLB functionalities are implemented in a distributed architecture. This means that each eNodeB has his own vendor specific MLB solution running by its self inside each eNodeB.

As explained in (18), the solutions implemented should follow some basic properties to be suitable solutions for real world scenarios. Here are some of the desirable properties for MLB algorithms:

- Decentralized Solution to support the distributed architecture.
- Stability and Convergence : to minimize oscillations and unstable behavior
- Availability, Consistency and Reliability of the information introduced in the algorithm.
- Overhead and delay reduction : reduce exchange of information and have the information available whenever needed.
- Adaptation mechanisms to different time scales, configuration settings and scenarios.
- Coordination between neighbor eNodeBs and other SON functionalities to prevent conflicts and continuous parameter changes.

- Predictability on the algorithm to avoid unknown behavior.

In the implementation explained in the following sections, this goals have been taken into consideration since the purpose of the solution was to embed it in a real-world solution.

5.2 Mobility Load Balancing States

As in explained in chapter 4, the eNodeB has to monitor the virtual load to detect an overload situation at the eNodeB. Depending on the virtual load of the cell ($\hat{\rho}_c$), we can define a state-machine with 3 different states, as depicted in figure 5.1.

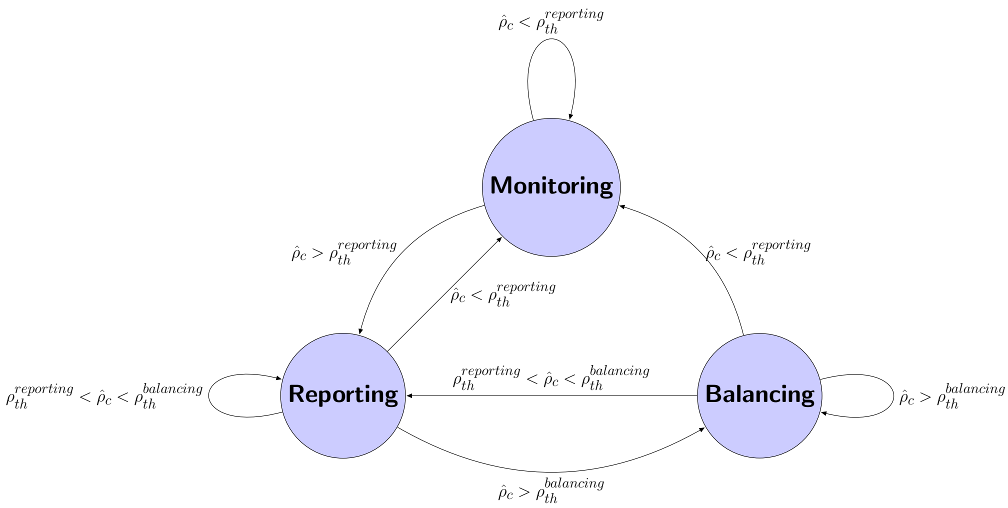


Figure 5.1: Mobility Load Balancing States-Machine -

The three states are :

- Monitoring : The MLB algorithm monitors the virtual load. Similar to an idle state where the algorithm still needs to make some calculation to have the corresponding information available.
- Reporting : MLB starts collecting the information necessary from the neighbor eNodeBs and the UEs before it is needed, to reduce the delay and the overhead.
- Balancing : The MLB algorithm considers the information available and makes the corresponding optimization decisions to solve the overload situation.

In the next sections, the different states will be explained in further details.

5.3 MLB Monitoring State

The following flow chart describes the actions taken in the monitoring state of the MLB Algorithm implemented.

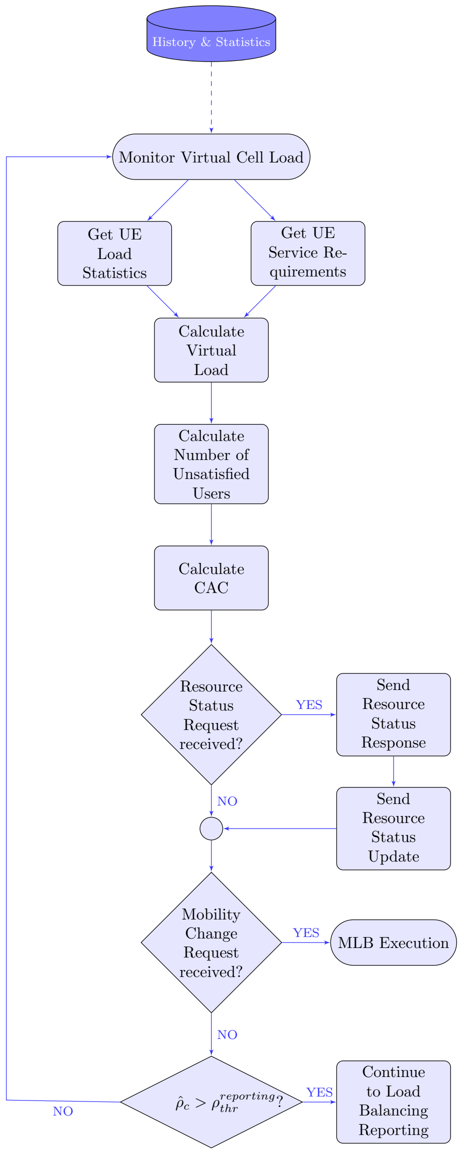


Figure 5.2: Monitoring State Flow Chart -

As depicted in figure 5.2, the MLB Algorithm needs to have some parameters available even though the optimization process is not yet been triggered. This parameters are needed to control the MLB state-machine, to evaluate the performance of the cell and to report load measurements to other neighbor eNodeBs.

5.3.1 Virtual Load Implementation

The virtual load is one of the most important parameters since it will allows the MLB state-machine to go to other states depending on his value compared with the different load thresholds already explained. In order to make the required calculations for the Virtual Load of the cell, the eNodeB is required to keep track of all the UEs load statistics and have the information of each UE CBR service type requirements available to calculate the Virtual Load.

Once we have this information, we need to estimate the required number of PRBs ($\hat{N}_u$) to calculate the virtual load as defined in the original equation 4.7:

$$\hat{\rho}_c = \frac{\sum_{u=1}^{u_T} \hat{N}_u}{N_{tot}} \tag{5.1}$$

To estimate $\hat{N}_u$, we use a simple cross-multiplication and we establish a limit for the worst case ($\hat{N}_{max}$) considered just for CBR users. For each user we calculate $\hat{N}_u$ as:

$$\hat{N}_u = N_u^{alloc} \cdot \frac{D_{u,CBR}^{req}}{D_u^{alloc}} \qquad \text{where } 0 \leq \hat{N}_u \leq \hat{N}_{max} \tag{5.2}$$

This estimation assumes that the SINR in the $\hat{N}_u$ will be the same as the average SINR of the allocated PRBs of the user u (N_u^{alloc}). With this in mind, we can estimate the number of

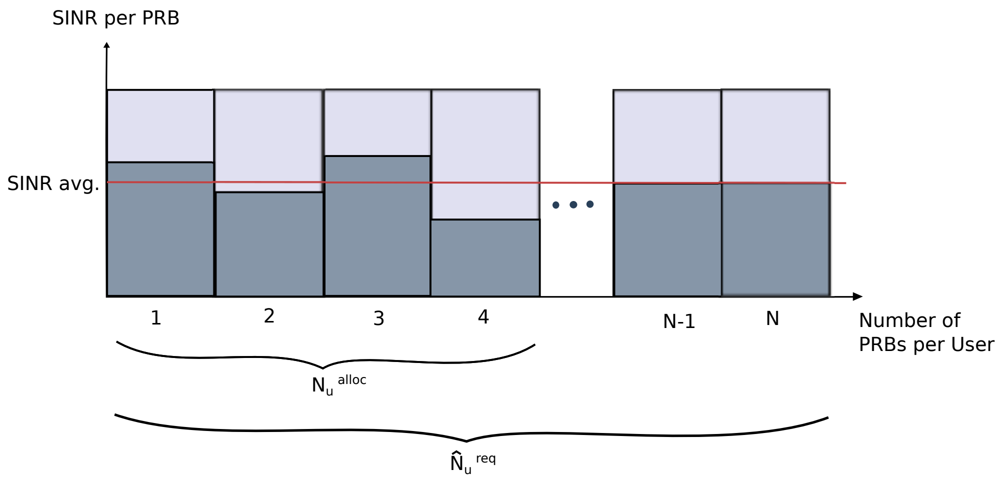


Figure 5.3: SINR Estimation of unallocated PRBs -

PRB required comparing the data rate required by the CBR service ($D^{req}_{u,CBR}$) against the data rate achieved (D^{alloc}_{u}) by the PRBs allocated.

5.3.2 Number of unsatisfied Users

Another interesting parameter to calculate is the number of unsatisfied users in the cell. From a network management point of view, this parameter is a good KPI (Key Performance Indicator) to track since it measures the user satisfaction inside the cell. We will be using this parameter to study the performance of the MLB solution on the next chapter.

The number of unsatisfied users can be calculated in two different ways. One way would be as describe in equation 4.8, that is calculating it directly from the virtual load. A second way, as it is done in the implementation, would be to count the number of unsatisfied users one by one. In the latter approach, we consider that a user is satisfied if:

$$D_u^{GBR} \geq 0.98 \cdot D_u^{CBR} \quad \text{then user } u \text{ is satisfied} \tag{5.3}$$

That is, when the Guaranteed Bit Rate, the data rate performed by the user, is bigger or equal to the Constant Bit Rate requirement of his CBR service type. In this case, the user will be considered as satisfied. This second approach of calculating the number of unsatisfied users allows to give consistency and reliability to the data retrieved from the simulator.

5.3.3 Composite Available Capacity (CAC)

The Composite Available Capacity is another specific MLB parameter that needs to be calculated when the eNodeB is not overloaded. This parameter needs to be available in the monitoring state so it can be signaled to other neighbor in case any of them request it. The CAC confirms the willingness of the eNodeB to accept extra traffic from its neighbors in case overload happens in any of them.

In the implementation, the CAC is signaled in the following way:

$$CAC = \begin{cases} 0 & \text{if } \hat{\rho}_c \geq \rho_{thr}^{reporting} \\ \rho_{thr}^{reporting} - \hat{\rho}_c & \text{if } \hat{\rho}_c < \rho_{thr}^{reporting} \end{cases} \tag{5.4}$$

This means that if the virtual load in the cell ($\hat{\rho}_c$) is bigger that the reporting threshold, the MLB would signal as if he was overloaded. In the other case, the spare capacity between the virtual load of the cell and the reporting threshold will be signaled as the amount of traffic that the eNodeB is willing to accept from overloaded neighbors. It is referenced to the reporting threshold to leave a margin of load so the eNodeB does not get into overloaded mode when traffic is offloaded from a neighbor eNodeB. The latter situation would produce oscillations on the Handover settings of both eNodeBs and bad optimization decisions as studied in (26).

5.4 MLB Reporting State

The following flow diagram describes the actions followed in the MLB Reporting State:

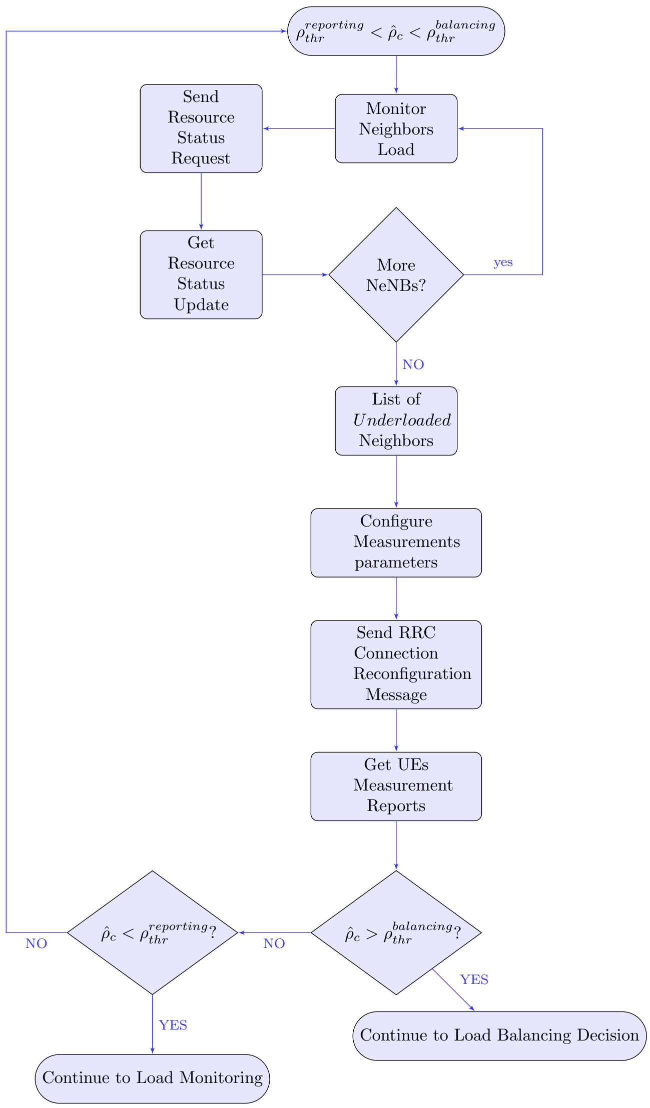


Figure 5.4: Reporting State Flow Chart -

Whenever the Virtual Load of the cell reaches the reporting threshold ($\rho_{thr}^{reporting}$), the MLB Framework changes its status to a reporting state, as shown in figure 5.4. This state pretends to make the information available before it is needed by the MLB Algorithm, but not to soon, to prevent overhead and unnecessary information exchange or continuous configuration changes.

Once the virtual load is greater than the reporting threshold, the eNodeB starts monitoring the load status of its neighboring cells. For that matter, the eNodeB sends Resource Status Request over the X2 interface to his neighbors and waits for a Resource Status Update from each one of them. Once all the information is available, the MLB framework composes a List of underloaded Neighbors. Those neighbors included on his list will be considered as possible eNodeB candidates that would help him lower his overload in case it appears.

The list of underloaded cells also helps in real LTE systems to configure the measurement parameters of the different UEs depending on their closeness to the neighbor eNodeBs. This measurement parameters are sent in the RRC Connection Reconfiguration message over the air interface to the UEs. Once the UEs are configured, the eNodeB collects all the UE measurement reports from the different UEs.

As noted earlier, this stage allows the eNodeB to have the necessary information before the Load Balancing functionality is triggered.

5.5 MLB Balancing State

In this section, the Mobility Load Balancing Algorithm chosen for the thesis , and depicted in figure 5.5, is described in thorough detail.

The MLB Balancing State is attained whenever the Virtual Load in the Cell ($\hat{\rho}_c$) is bigger than the MLB threshold ($\hat{\rho}_{MLB,Thres}$). We will suppose that this algorithm has been running continuously and that the cell has become overloaded over a period of time. Thus, the information necessary for the algorithm is available to be examined and processed since the reporting state has already been reached and the information has also been retrieved from the different sources.

The first thing to do when the state machine is in Balancing Mode is to set the Composite Available Capacity (CAC) equal to zero. By doing so, the eNodeB will signal to the Neighbor eNodeBs, whenever queried, that its load is reaching the limit and therefore no extra load will be admitted.

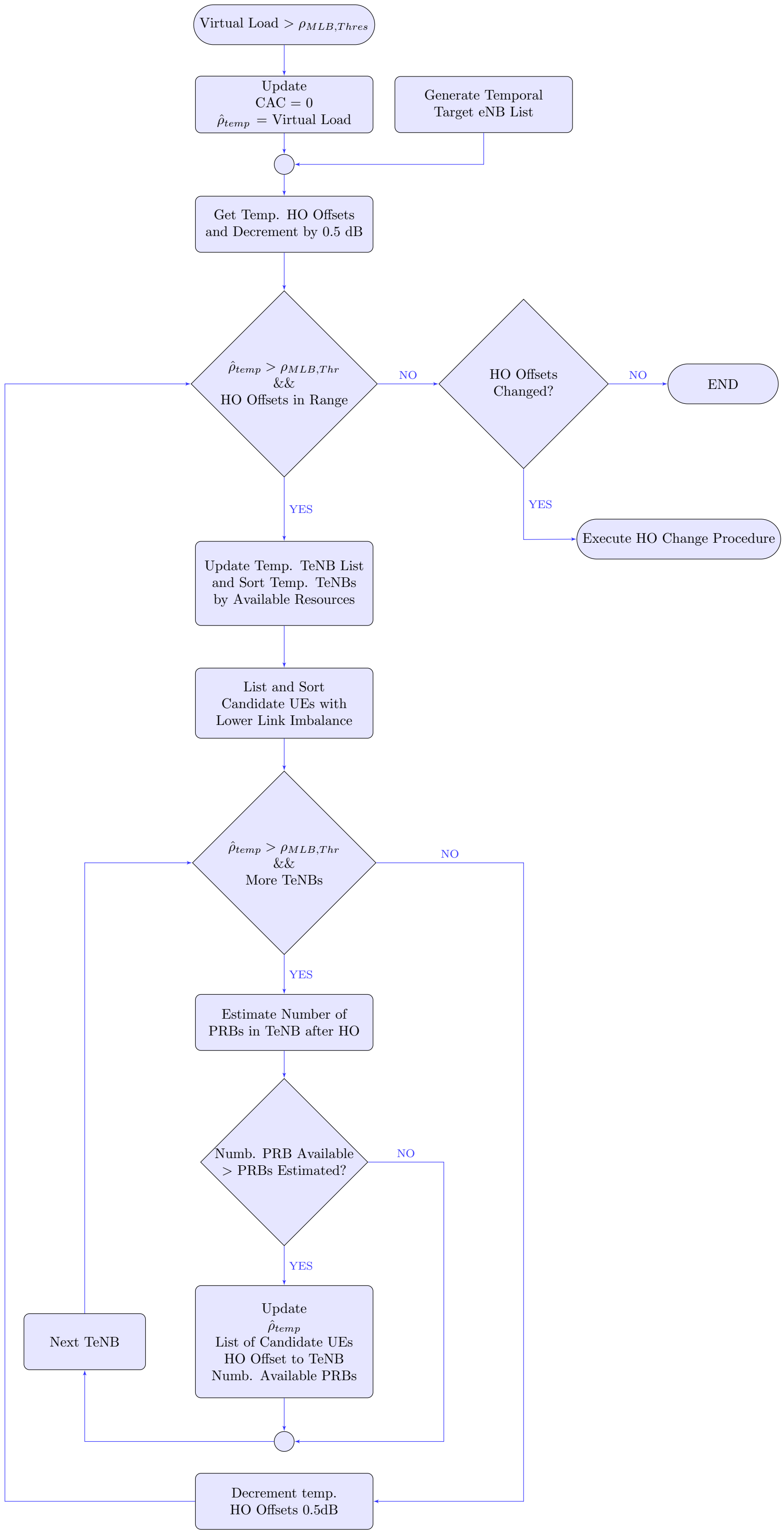


Figure 5.5: MLB Algorithm Flow Chart -

At this point, the eNodeB creates a temporary Target eNodeB (TeNB) list with all the required information received from its neighbor eNodeBs (NeNB). Here, Target eNodeB is the name given to the candidate neighbor eNodeB that is willing to accept extra load from its neighbors. Thus, being its CAC greater than zero.

The information available for each target eNodeB is:

- Composite Available Capacity (CAC)
- Number of available Physical Resource Blocks (PRBs) at TeNB
- List of candidate UEs to Handover
- Number of PRBs needed after Handover
- Number of Users to Offload
- Handover Offset with Target eNB

The temporary TeNB list and the temporary virtual load, set to the actual virtual load, will be used along the algorithm to make decisions regarding changes on the handover offsets. Those changes will be tracked and stored in a temporary Handover Offset vector, which will help negotiate the changes in the parameter change procedure at the end of the MLB decision process.

Now that the framework has been described lets proceed to explain how the MLB decision process is implemented.

Once the information needed is available, the next thing to do is to decrement the temporary Handover Offsets by 0.5 dB. The step (0.5 dB) by which the Handover Offsets are decremented is specified by the 3GPP standard, see (10). The goal of decrementing the temporary Handover Offsets is to help the algorithm estimate the behavior of the system if the changes had to be made. From now on the decrements to the temporary Handover Offset will be made to a specific eNodeB as long as it keeps within the limits established by the standard. That is between -10 dB and 10 dB.

Following figure 5.5, the algorithm starts looking for the best solution while two conditions are met. The first condition is to keep looking for a better solution while the temporary virtual load still above the MLB threshold. Thus, until the future estimated virtual load is low enough to make the eNodeB switch to a reporting state and therefore, the eNodeB will not be overloaded anymore. The second condition to stop looking for the best possible solution is to check if the Handover Offsets of at least one of target eNodeBs is within the limits established by the standard ([-10 dB, 10 dB]). This condition checks if any of the Handover Offset of the remaining target eNodeBs could be forced 0.5 dB more to offload extra UEs.

When the two conditions are met, the algorithm updates the target eNodeB list since one or more Handover Offsets of the target eNBs could have reached their limits and therefore, should not be taken into consideration for future optimizations. Once the Target eNodeBs are well known, the algorithm sorts them regarding their CAC, from greater to lower values. This is equivalent to sort the TeNBs considering their available load capacity and willingness to accept UEs from its neighbors.

The next step is to locate all UEs that are close to a specific TeNB and make a list for each TeNB. To find out whether the UEs are close to a target eNodeB we need to calculate the link imbalances [1] of each UE in the coverage area to the different TeNBs with the corresponding temporary Handover Offset. The UEs with the available RSRP measurements to the TeNB will be added to the corresponding list of Candidate UEs of that particular TeNB and then, the algorithm will make sure that all the UEs on the lists are sorted from the ones with lower link imbalance to the ones with greater link imbalance.

The next stage on the algorithm will start the estimation process. This process will take one TeNB of the list at a time and will run until either the temporal virtual load is lower than the MLB threshold or there are no more TeNB to consider in the list . The estimation process will take the list of temporal UE candidates from the target eNodeB and will estimate for each UE what is the Number of PRBs necessary to offload to the target. The estimation calculation is explained in further details in section 5.5.2. Once, all the estimations are calculated, the algorithm will sum up all the results and then this sum will be compared to the actual available capacity of the target eNodeB.

If the available capacity is lower than the estimated required capacity then the algorithm will continue trying with the next target eNodeB on the list.

If, on the other side, the available capacity is higher than the estimated required capacity, the algorithm will update a set of variables. Those variables are:

- Number of Available PRBs on the Target eNB: Considering that the Handover Offset will be changed, the algorithm needs to subtract the number of PRBs that it estimated necessary to offload the candidate UE from the number of available PRBs on the Target.

[1] The link imbalance is explained in section 5.5.1.

This new value will be used in future optimization when considering if the target eNodeB has enough capacity to accept more UEs from its neighbors.

- Temporary virtual load ($\hat{\rho}_{temp}$): the average number of PRBs used by the UEs that will be offloaded, need to be subtracted from the temporary virtual load. This new temporary virtual load will be reevaluate to decide if further optimizations are needed.

- List of Candidate UEs : After estimating which UEs on the list will be offloaded, the algorithm needs to make sure that this UEs will not be considered again in future optimization from the target eNodeB as well as from from another TeNB. Therefore, it needs to make sure that the UEs already considered as offloaded, do not appear in any other candidate list.

- Handover Offset to Target eNodeB: Due that the estimations have been made based on a preconceived situation. The algorithm needs to keep track of all the change that will need to be made once the optimization process is done. Therefore, the Handover Offsets that need to be changed are stored in the Target eNodeB list.

Once all the variables are updated, the algorithm continues evaluating the temporary virtual load and if it is still greater than the MLB thresholds, the algorithm restart the estimation process for the next Target eNodeB. When the algorithm runs out of Target eNodeBs, it decrements the temporary Handover Offset by 0.5 dB and starts again evaluating the link imbalances of the UEs and making new estimations after offloading the corresponding UEs. If any new enhancements can be made to the decision process, the algorithm will update for each decremental step until the overload has disappeared or the Handover Offset of the TeNBs have reached their limits.

Since we have been keeping track of the changes necessary to make to the Handover Offsets of the Target eNodeBs, whenever any changes happen the algorithm launches the handover parameter negotiation by the parameter change procedure explained on the next section. Otherwise, the algorithm ends its executions if no changes have occurred.

The next 2 subsections explain the two concepts necessary to complete the explanation and that have not been explained in this section.

5.5.1 Link Imbalance

To explain the link imbalance we need to recall equations 2.4 and 2.5, used on the mobility management of the UEs to select the most suitable cell, to which the UE should be connected to, based on measurements performed by the UE:

Measurement to the Serving eNodeB:

$$R_s = Q_{meas,serving} + Q_{hyst,s} \tag{5.5}$$

Measurement to the Neighbor eNodeB:

$$R_n = Q_{meas,neighbor} - Q_{offset_{s,n}} \tag{5.6}$$

and the Cell Ranking Algorithm that selects a cell by the following Criteria:

$$Selected\ Cell = max\{R_s, R_n\} \tag{5.7}$$

Thus, whenever an UE meets the condition, $R_n > R_s$, the UE will be handover to the neighbor eNodeB with the strongest received signal. That is,

$$R_n > R_s \Rightarrow Q_{meas,neighbor} - Q_{offset_{s,n}} > Q_{meas,serving} + Q_{hyst,s} \tag{5.8}$$

Re-arranging the terms,

$$Q_{meas,neighbor} > Q_{meas,serving} + Q_{hyst,s} + Q_{offset_{s,n}} \tag{5.9}$$

Meaning that, in order to make handover the UE to a NeNB the received signal from the neighbor has to be greater than the sum of the received signal from the connected eNB, the hysteresis parameter and the Handover Offset between both eNodeBs.

To favor that the UE hands over to a Neighbor eNodeB we need to adjust the Handover Offset to meet the previous condition. If we develop the condition:

$$Q_{offset_{s,n}} < Q_{meas,neighbor} - [Q_{meas,serving} + Q_{hyst,s}] \tag{5.10}$$

So the Handover Offset needs to be lower that the difference between $Q_{meas,neighbor}$ and R_s in order to make the UE handover to neighbor cell. When connected to a certain cell, R_s is greater than $Q_{meas,neighbor}$ producing a negative value of the right side of equation 5.10. Therefore, in oder to meet the condition, the Handover Offset needs to be more negative than the right side

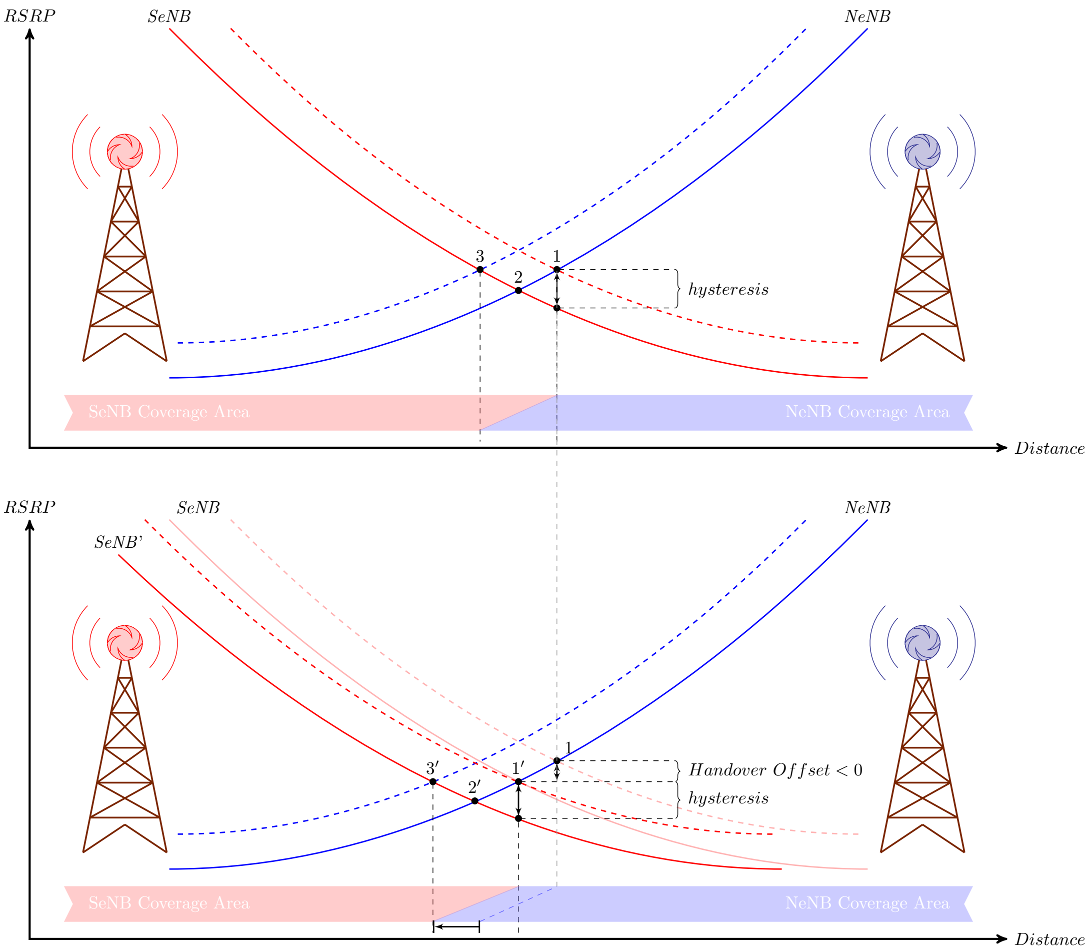


Figure 5.6: Effects of Handover Offset decrease - *Above without Handover Offset and Below with negative Handover Offset*

and that is the reason why in the algorithm the Handover Offset is reduced by 0.5 db step to find an optimal solution.

Figure 5.6 depicts the effects produced by the handover offset decrease between 2 neighbor eNodeBs. In the graph below we can see how the coverage area of the SeNB is decreased and at the same time the coverage area from the NeNB increases. This is exactly what we expected since when we consider a constant UE density on the area, if the area is reduced then the number of UE in the reduced area also reduce. We can also see that the RSRP levels for handover are lower than before and that the border area of both eNodeBs are bigger. This fact influence the number of drop calls in a negative way. To counteract the spread on the border area it will be required to coordinate the changes done on the handover offset for both eNodeBs. The procedure to synchronize changes in mobility parameters will be explained in the next section.

Now, to make it easier for the algorithm, and for us, to estimate when the handover condition will be met and thus, estimate when the UE will be switched to the NeNB we use the Link Imbalance of that particular UE. The link Imbalance is define as follows:

$$Link\ Imbalance = R_s - R_n = \begin{cases} \geq 0 & \text{when SeNB more suitable cell} \\ < 0 & \text{when NeNB more suitable cell} \end{cases} \tag{5.11}$$

We can conclude that, when a UE is connected to its Serving eNB and its link imbalance is negative, the NeNB should be considered as a more suitable base station for the UE. When estimating which UEs will be handover if an hypothetic handover Offset change occurs, the algorithm will consider as candidates UE to offload, the ones with a negative link imbalance.

The link imbalance, when positive, could also be thought as the gap or amount effort to be made to offload a particular UE to a particular eNodeB. This issue can be helpful to reduce the number of operations in the algorithm when the effort to be made is higher than the one permitted by the handover offset limits.

5.5.2 Load Estimation on Neighbor eNodeB

The aim of this section is to explain in detail how it is estimated the impact on the load when offloading a UE to a particular NeNB.

The eNodeB has knowledge of the types of services that each UE requests. The type of service defines the requirements needed for a particular service. This requirements are

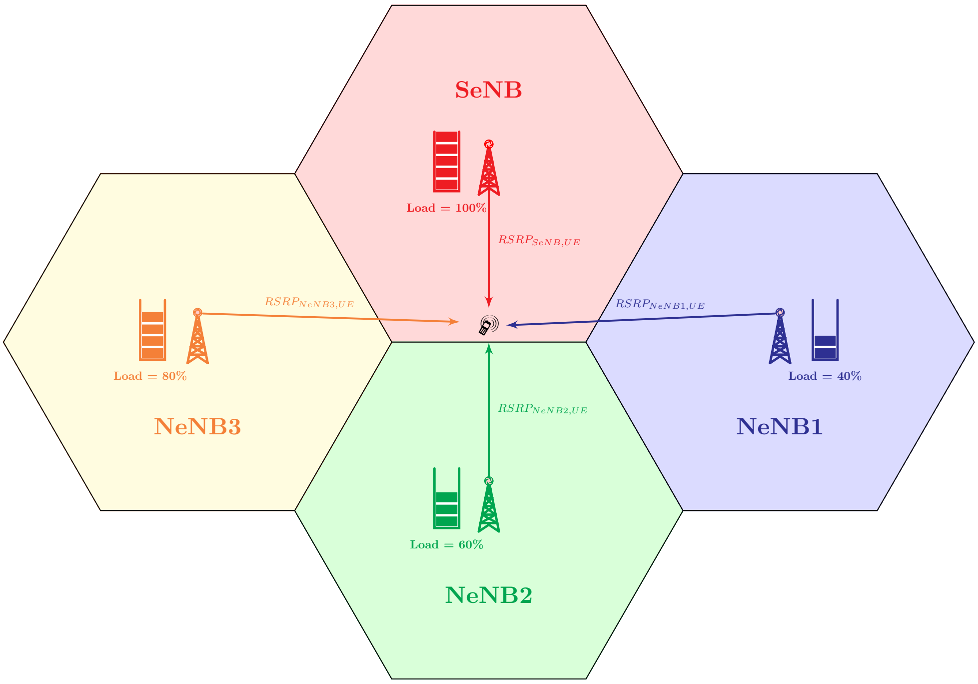


Figure 5.7: SINR estimation in a Neighbor Cell -

translated into guaranteed bit rates that need to be fulfill to satisfied the service requested by the UE.

The goal of the algorithm is to determine the requirements needed to fulfill a service requested by a UE in a Target eNodeB. This estimation needs to be made, to decide wether the TeNB will be capable of fulfilling the service of a UE that is been considered to be offloaded to that same TeNB. If the NeNB is capable of supporting the service of the UE, then the UE will have great chances to be offloaded to the Target eNodeB if the algorithm considers that changes to the handover offset are beneficial to counteract the overloading situation at the Serving eNodeB.

To estimate the requirements needed on the TeNB to satisfy the UE service, the algorithm estimates the SINR per PRB that the UE will perceived whenever a it is handover to the Target eNodeB.

In figure 5.7, it is depicted the situation encounter by a UE located in the border between 2 cells with 3 Neighbor eNodeBs. The measurements made by the UE, requested by the SeNB, to the Neighbor eNodeBs are represented by the arrows with the RSRP labels. Due that the SeNB connect to its Neighbor eNodeBs through the X2 interface, the SeNB has knowledge of the loads of its Neighbors.

With the information that the SeNB holds, the algorithm can estimate the SINR per PRB of the UE in a TeNB with the following formula:

$$\widehat{SINR}_{TeNB,UE} = \frac{RSRP_{TeNB,UE}}{N(BW) + \sum_{n=1}^{N_{NeNB}} \rho_n \cdot RSRP_{NeNB_n,UE}} \tag{5.12}$$

Where, N_{NeNB} is the number of Neighbor eNodeBs that the UE can measure from its location including the Serving eNodeB. ρ_n is the ponderation given to each RSRP measurement, its value depends on the load of the measured eNodeB. This factor represents the probability that a PRB being used by the UE, is also being used by another eNodeB in another cell and thus, interfering with him. The Noise figure $N(BW)$ is a function of the bandwidth of the PRB which is 180 kHz and approximated in the algorithm to -174 dBm/Hz.

Once we have the SINR per PRB estimation, we can map the SINR with the modulation coding scheme (MCS) to have an idea of the Data Rate per PRB ($R(SINR_{TeNB,UE})$) reachable with the actual interference conditions, as explained in section 4.6.

Since the algorithm knows the type of service and therefore, the data rate that needs to be guaranteed for the UE to be satisfied, we can calculate the number of PRBs required in a neighbor eNodeB with the Data Rate per PRB at the current position of the UE. The formula as in the previous chapter is:

$$\widehat{N}_{UE,TeNB} = \frac{D_{UE,CBR}}{R(\widehat{SINR}_{TeNB,UE})} \tag{5.13}$$

$D_{UE,CBR}$ is the guaranteed bit rate for the CBR service requested and $R(SINR_{TeNB,UE})$ the data rate per PRB. This formula considers that all the PRBs allocated to the UE in the NeNB will have the same SINR conditions and that might not be the case in real system.

The last step in the estimation of load in the neighbor eNodeB will be to sum up all the PRBs needed for all the UEs that will handover to the NeNB if the the handover offset changes occur. That is,

$$\widehat{N}_{Total,TeNB} = \sum_{u=1}^{N_{UE}} \widehat{N}_{u,TeNB} \tag{5.14}$$

$\widehat{N}_{Total,TeNB}$ is the value that will be considered when checking for load availability in the NeNB and will decide if changes to the handover offset need to be made.

5.5.3 MLB Handover Offset Adjustment

When the algorithm decides to make changes to the handover offsets between its neighbor eNodeBs, the handover offset change procedure is triggered as explained in section 4.7.4. The procedure negotiates the handover offset changes between each Neighbor eNodeB. The NeNB can reply with a negative response which would leave the handover offset intact or it can be reduced by the negotiation process. Although it would be interesting to implement this mechanism, for the purpose of this thesis, the procedure is not implement. This means that the changes on the handover offsets are accepted and therefore change every time an eNodeB request it.

The important issue in this change procedure is that the handover offsets need to be synchronized. This means that a reduction of the handover offset in one eNodeB will need to be followed by an increase on the handover offset of the neighbor eNodeB with whom it shares that parameter. This helps avoid handovers with lower received power as depicted in figure 5.6. Where we can see the effect on the received power when the change is just done in the Serving eNodeB.

Once the handover offsets are changed, the algorithm informs the UEs so that the handover procedures happen automatically. This allows a natural flow of UE between eNodeBs without forcing extreme conditions and it makes the algorithm independent from the mobility patterns of the UE. If we wanted to force individual UE to handover to specific eNodeB, the algorithm would need to take into consideration the mobility patterns of the UEs.

6

Results

This chapter contains the results from the evaluation of the MLB algorithm done in a real-time LTE Mobile Radio System Simulator called AMoRE. This was developed by Nomor Research GmbH and is exclusively owned by the Research and Development department of Nokia Siemens Networks.

In the first section, a short introduction to the simulator is provided to better understand his purpose and usage. Then, the results of the performance of the MLB algorithm obtained from the simulator will be evaluated and the effects on the network performance discussed in further details.

6.1 Amore Simulator

AMoRE (= Advanced Mobile Radio Realtime Experience) is a system-level simulator for Mobile Radio Access Networks that executes in (or near) real-time. The simulator allows for interactive change of configuration parameters, as well as, conditions and the immediate observation of the resulting impacts on the system behavior with the selected performance measurements.

Starting already in 2005, AMoRE has originally been developed as a demonstrator tool for WCDMA, WiMAX and LTE radio technologies to provide insights into the operation of radio features in loaded networks, the interaction between protocol layers and ultimately the effects on the user perception of system performance. The latter is facilitated by the realtime capability of AMoRE that allows to run live applications (like Web browsing or video streaming) across the simulated and synthetically loaded radio network and observe its end-to-end behaviour under different radio and traffic conditions and in different feature configurations.

Besides the intended demo value for an intuitive explanation and promotion of technology and feature benefits, AMoRE has also turned out highly useful for training purposes by quickly and illustratively conveying complex features and interactions. Nonetheless, it also presents a valuable research tool by facilitating experiments to quickly confirm hypotheses and gaining insights in often unexpected effects that would be hard to detect by offline simulations alone.

While the original version of the AMoRE simulator was limited to single cell simulations (with the surrounding cells modeled as synthetical interferers), the new generation allows for more complex multi-cell simulations in single or multi-layer deployments thus facilitating realistic models for coordinated transmission or scheduling schemes between multiple cells. It still executes on a transportable laptop PC under Linux, but now requires high performance hardware with a fast quad-core processor and a high-end graphics card that is heavily employed for parallel channel computations in the radio simulation.

The LTE Radio Emulator provides a multi-cell, multi-user system-level simulation in real-time, also taking into account the interference from surrounding cells in a cellular environment. The simulated users experience variable radio conditions according to a detailed statistical and physical model. User mobility between cells is only considered in the simulated area of interest. Data traffic of the users in downlink and uplink is provided from artificial traffic generators for the majority of users, and from live applications for one or few users. Each user can be assigned multiple data flows with different traffic patterns. The Radio Emulator models a RAN protocol stack for the user plane, through which the entire user traffic is routed. In this sense, it is not a pure radio emulator, but must rather be regarded as a RAN emulator including major functions of the Radio Access Network in addition to the radio interface. Nonetheless, we continue to use the term Radio Emulator as it has already been introduced in the project plan and related project documents. The protocol implementation is simplified to the main functionality in order to allow for real-time performance, a particular focus in terms of functional accuracy is laid on the MAC layer simulation. The PHY layer is emulated in its effects on the packet loss, making use of off-line link-level simulation results. The Emulator executes the steady state of a given radio link configuration, i.e. user plane data transmissions through existing and pre-configured radio links. Users do not appear or vanish during the simulation (birth/death model), however, there may occur silence periods within the users traffic patterns. C-plane functionality for the reconfiguration of established radio channels is not supported.

6.2 Simulation Scenario Showcase and Parameter Configuration

The scenario showcase simulated for the evaluation of the MLB algorithm has 7 simulated cells, one central cell and six cells surrounding the one on the center. The purpose of the simulation is to study the behavior of the MLB algorithm whenever overload occurs. For that purpose, the highest distribution of users will be placed in the central cell where overload will be forced . In this scenario, 10 users will be placed on the surrounding cells and the remaining users in the central.

The simulations made during the evaluation had between 90 and 120 users simulated simultaneously. To increment the load on the cells, different types of services are configure to analyze the performance depending on the type of service. The type of services used for the simulations are constant bit rate services (CBR). The services employed go from 512 Kbps up to 3 Mbps on the downlink and on the uplink.

The simulator has a huge amount of parameters to configure. Due to the extended list of parameters, just the basic physical parameters will be describe in Table 6.1.

Parameter	**Simulated**
Duplex Mode	Full Duplex FDD
DL Access Mode	OFDMA
UL Access Mode	SC-FDMA
Carrier Frequency	2 GHz
Bandwidth	10 MHz
TTI length	2 ms
Antenna configuration	1x2: (Rx Diversity)
Subcarrier Spacing	15 kHz
Chunk Width	180 kHz (Frec. domain), 1 TTI (Time domain)
FEC for data channels	Turbo Code
Modulation Schemes	QPSK, 16QAM, 64QAM

Table 6.1: AMoRE Simulation Parameter Configuration .

6.3 Performance Evaluation of the MLB Algorithm

In this section, the results obtained from the simulator will be explained in further detail. The figures shown in the following subsection are the outcome from the post-processing of the data in Matlab. To show the performance of the users and the whole system, the cumulative distribution function (CDF) will be depicted to analyze how the throughput is distributed among users and how the algorithm enhances the performance of the whole system. The figures depicted compare the performance of the users and the system when the MLB algorithm is not been used and when the algorithm is performing optimizations.

6.3.1 90 Users Simulations

The simulation shown in this section correspond to the scenario where 10 users are located randomly in the 6 surrounding cells and 30 users are located in the center cell randomly distributed. The user move randomly over time whit a Random Walk mobility model. The simulations are running during 15 minutes of simulations time, to allow the user to move around and see the way the MLB algorithm adapts to the changing conditions.

6.3.1.1 90 users: DL CBR 1024/512, UL CBR 1024/512

In this scenario, the users are configured to produce a constant bit rate service type. The user in the center cell have 2 different service types. 15 of the central user have a CBR service of 512 Kbps in downlink and uplink, and the other 15 users a CBR service of 1024 in both link directions.

Figure 6.1, shows the cumulative distribution function of the downlink PDCP throughput of the users in the downlink. As it is shown, the difference between both situations is minimal. This is due to the fact that the MLB Algorithm does not reach the MLB Threshold easily, not overloaded, and thus the MLB Algorithm do not run and therefore a small amount of optimizations are happening.

In figure 6.2, we can see the effect on the downlink PDCP throughput of the whole system. Here, we can see that the mean downlink PDCP throughput varies a very small amount. Nonetheless, the downlink 5%-ile PDCP Throughput, the minimum PDCP throughput reached by 95% of the users, increases when the MLB optimizations are enabled.

Figure 6.3, depicts the effect on the uplink of the PDCP Throughput per user. We can see here that the uplink suffers a decrease in the performance when the MLB algorithm is enabled in both mean uplink PDCP throughput and 5%-ile uplink PDCP throughput. This means that forcing the users on the edges of the cell produces a deterioration of the uplink performance.

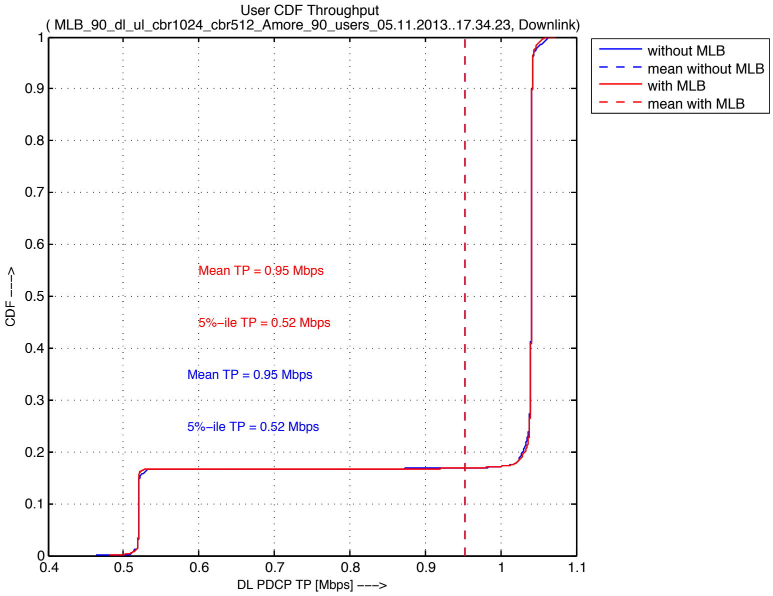


Figure 6.1: DL User Throughput: 90 users CBR1024+512 in DL and UL -

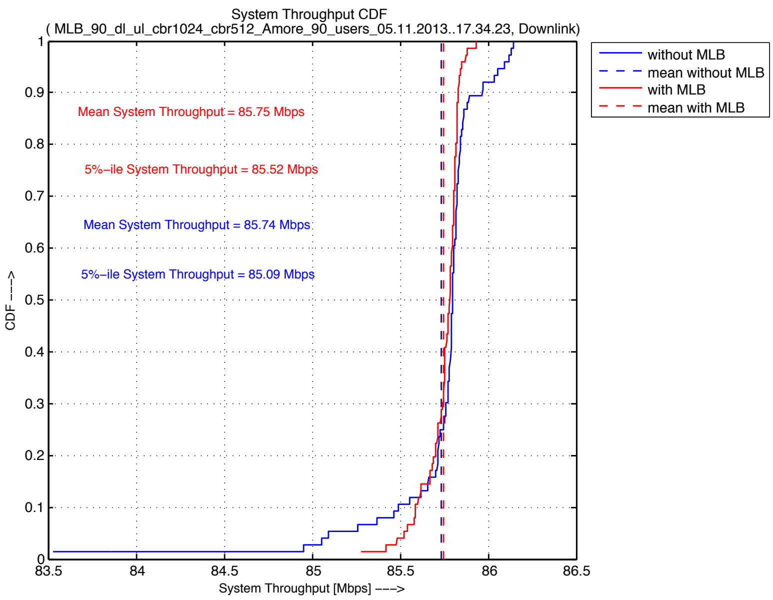


Figure 6.2: DL System Throughput: 90 users CBR1024+512 in DL and UL -

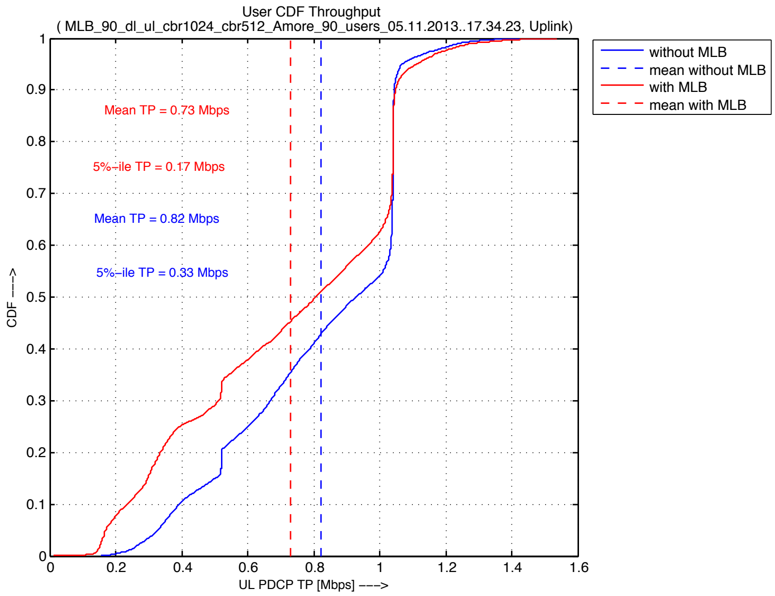


Figure 6.3: UL User Throughput: 90 users CBR1024+512 in DL and UL -

Figure 6.4, shows a much clear picture of the deterioration of the uplink for the whole system in both mean uplink PDCP throughput and 5%-ile PDCP throughput.

6.3.1.2 90 users: DL CBR 1024/2048, UL CBR 1024/2048

In this scenario, the users are configured to produce a constant bit rate service type. The user in the center cell have 2 different service types. 15 users of the central user have a CBR service of 1024 Kbps in downlink and uplink, and the other 15 users a CBR service of 2048 in both link directions.

Figure 6.5, shows an increment on the mean downlink PDCP throughput when the MLB Algorithm is enable. We can also see the increment on the 5%-ile downlink PDCP throughput which increases considerably with the MLB Algorithm. Meaning that the user on the edge of the cell, which normally would perform poorly than the ones closer to the eNodeB, get a better quality of service when the MLB algorithm is running.

In figure 6.6, we can see more clearly the effect on the overall system performance on the downlink. Since the users on the edge are handover to a different cell, the users that remain on the cell would have more resources available and therefore the whole system performs much better.

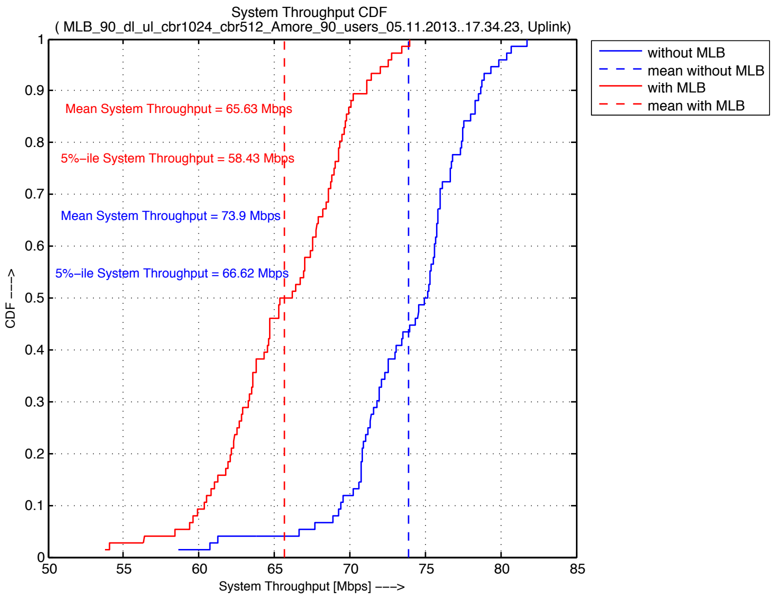


Figure 6.4: UL System Throughput: 90 users CBR1024+512 in DL and UL -

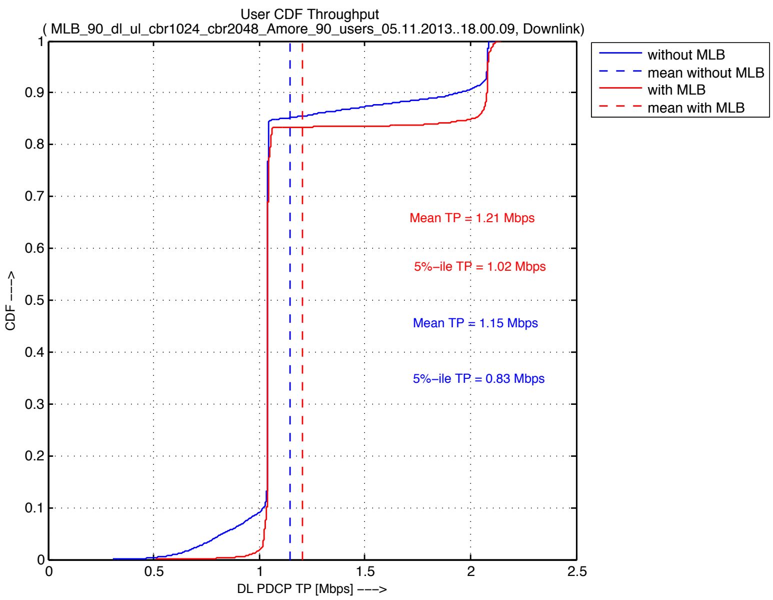


Figure 6.5: DL User Throughput: 90 users CBR1024+2048 in DL and UL -

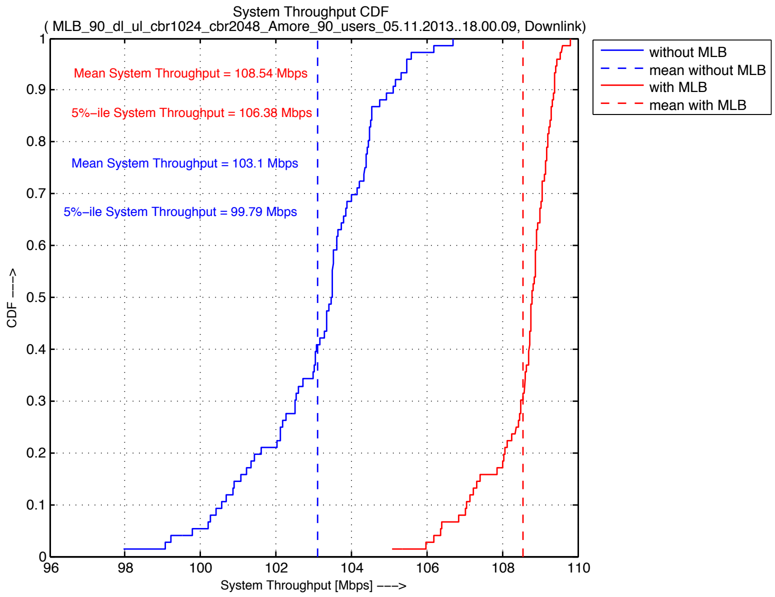


Figure 6.6: DL System Throughput: 90 users CBR1024+2048 in DL and UL -

On the other hand, as already seen in the previous scenario, the mean uplink PDCP throughput and the 5%-ile uplink throughput are reduce when MLB optimizations take place. This is shown in figure 6.7.

This fact is translated to the overall uplink performance as depicted in figure 6.8 where both paramaters decrease.

6.3.1.3 90 users: DL CBR 1800, UL CBR 1800

In this scenario, the users are also configured to produce a constant bit rate service type. The user in the center cell have the same service type. All users have a CBR service of 2048 kbps in both, downlink and uplink. In this scenario, we will see that the MLB algorithm runs more frequently than in the previous scenario.

Figure 6.9, shows the CDF of the user PDCP throughput measurements. We can see that the the user user that had more difficulties to satisfy its service perform better. This can be seen on the increase of the 5%-ile PDCP throughput with the MLB Algorithm.

The effect on the whole system is depicted in 6.10. There, the enhancements produced in the PDCP throughput are represented. We can notice that the increase on PDCP throughput, although considerable, is much lower than the increase with CBR 1024/2048 in the previous

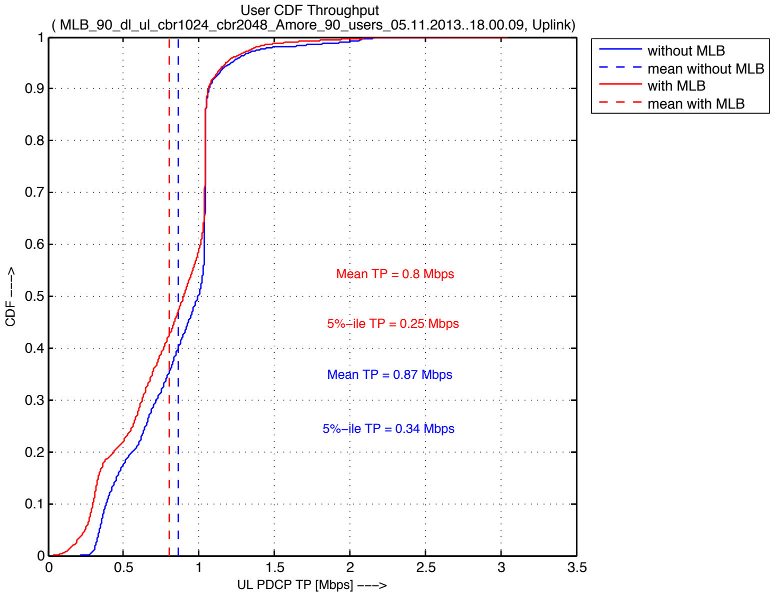


Figure 6.7: UL User Throughput: 90 users CBR1024+2048 in DL and UL -

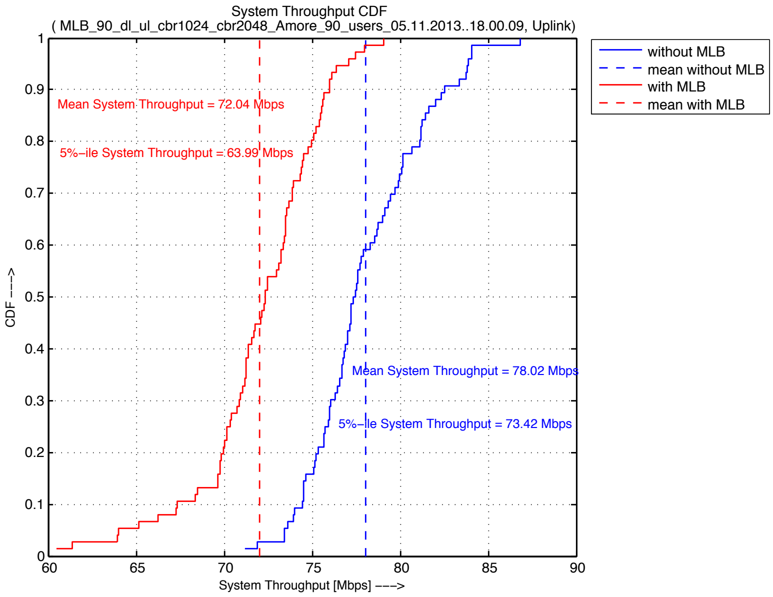


Figure 6.8: UL System Throughput: 90 users CBR1024+2048 in DL and UL -

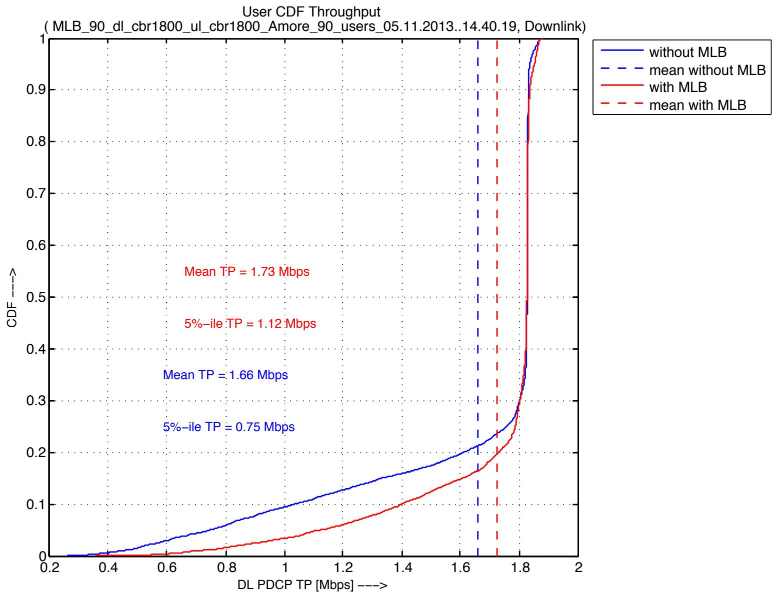


Figure 6.9: DL User Throughput: 90 users CBR1800 in DL and UL -

case. This is due to the fact that the offloading decisions made by the MLB Algorithm depends on the availability of resources on the neighbor eNodeB. Therefore, the bigger the service type, the harder would be to find available resource for the users. This will be much clear when we increase the number of users in the cell.

Figures 6.11 and 6.12 show the effect on the uplink with a CBR of 1800 kbps. We can see in this cases that the mean throughputs and 5%-ile throughputs decrease as in the previous cases.

6.3.2 100 Users Simulations

The simulation shown in this section correspond to the scenario where 10 users are located randomly in the 6 surrounding cells and 40 users are located in the center cell randomly distributed. The user move randomly over time whit a Random Walk mobility model. The simulations are running during 15 minutes of simulations time, to allow the user to move around and see the way the MLB algorithm adapts to the changing conditions.

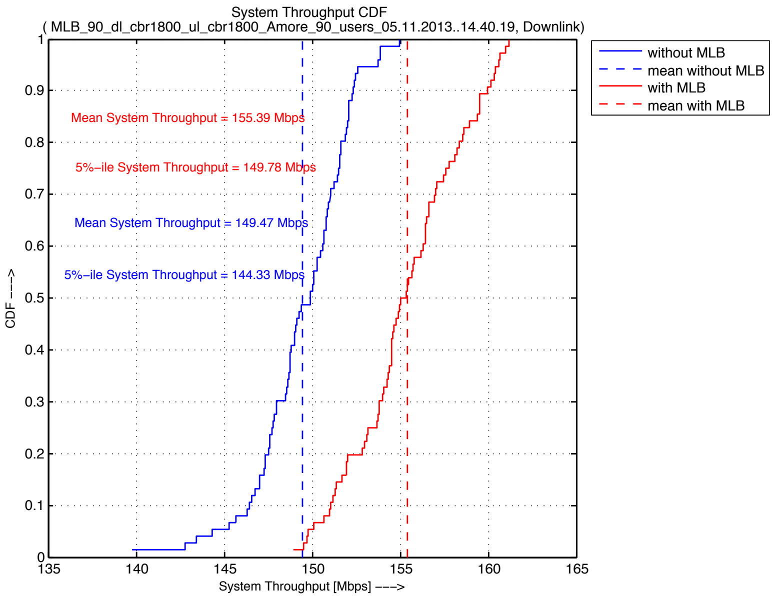


Figure 6.10: DL System Throughput: 90 users CBR1800 in DL and UL -

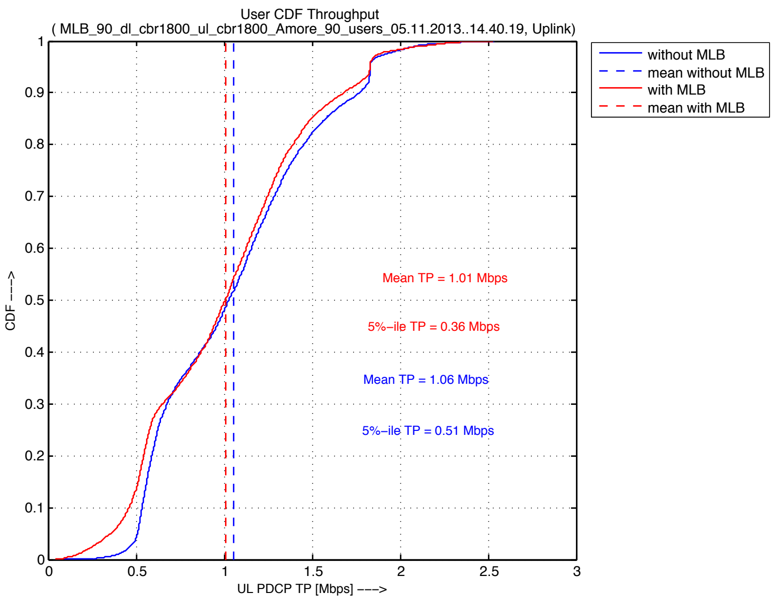


Figure 6.11: UL User Throughput: 90 users CBR 1800 in DL and UL -

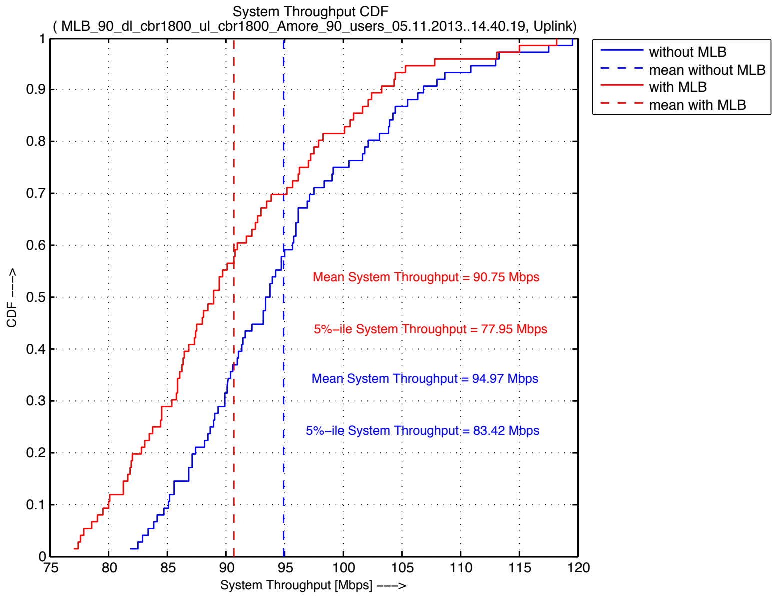


Figure 6.12: UL System Throughput: 90 users CBR1800 in DL and UL -

6.3.2.1 100 users: DL CBR 768, UL CBR 768

In this scenario, the users are also configured to produce a constant bit rate service type. The user in the center cell have the same service type. All users have a CBR service of 768 kbps in both, downlink and uplink.

Figure 6.13, shows the cumulative distribution function of the downlink PDCP throughput of the users in the downlink. As it is shown, the difference between both situations is minimal. This is due to the fact that the MLB Algorithm does not reach the MLB Threshold easily, not overloaded, and thus the MLB Algorithm do not run and therefore a small amount of optimizations are happening.

In figure 6.14, we can see the effect on the downlink PDCP throughput of the whole system. Here, we can see that the mean downlink PDCP throughput varies a very small amount. Nonetheless, the downlink 5%-ile PDCP Throughput, the minimum PDCP throughput reached by 95% of the users, increases when the MLB optimizations are enabled.

Figure 6.15, depicts the effect on the uplink of the PDCP Throughput per user. We can see here that the uplink suffers a decrease in the performance when the MLB algorithm is enabled in both mean uplink PDCP throughput and 5%-ile uplink PDCP throughput. This means that forcing the users on the edges of the cell produces a deterioration of the uplink performance.

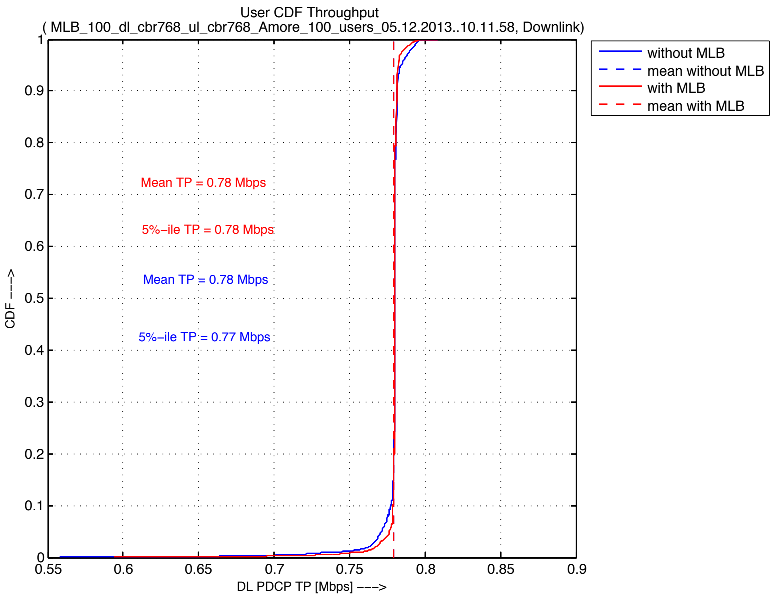


Figure 6.13: DL User Throughput: 100 users CBR768 in DL and UL -

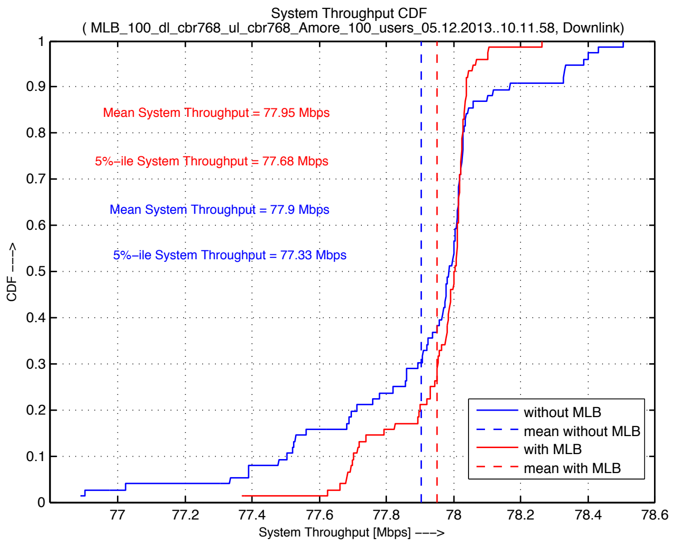


Figure 6.14: DL System Throughput: 100 users CBR768 in DL and UL -

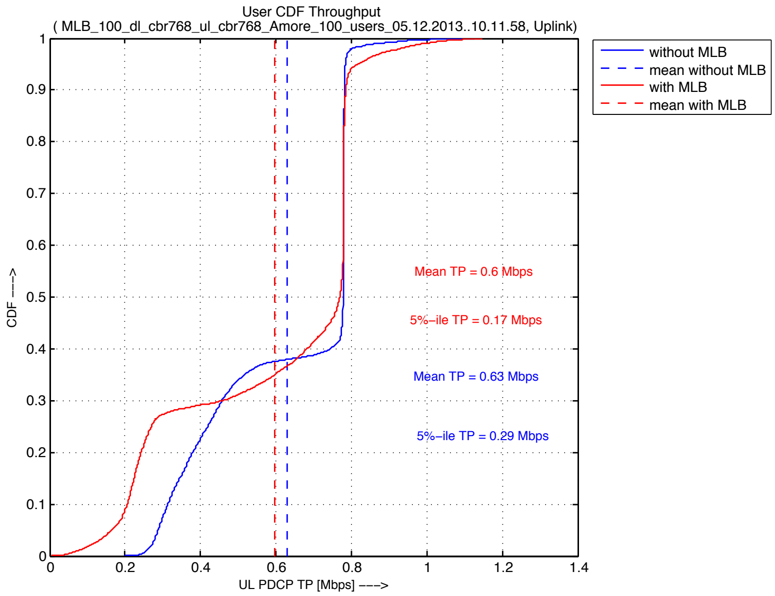


Figure 6.15: UL User Throughput: 100 users CBR 768 in DL and UL -

Figure 6.16, shows a much clear picture of the deterioration of the uplink for the whole system in both mean uplink PDCP throughput and 5%-ile PDCP throughput.

6.3.2.2 100 users simulations: DL CBR 1024, UL CBR 1024

In this scenario, the users are configured to produce a constant bit rate service type. The user in the center cell have the same service type. All users have a CBR service of 1024 kbps in both, downlink and uplink. In this scenario, we will see that the MLB algorithm runs more frequently than in the previous scenario.

Figure 6.17, shows an increment on the mean downlink PDCP throughput when the MLB Algorithm is enable. We can also see the increment on the 5%-ile downlink PDCP throughput which increases considerably with the MLB Algorithm. Meaning that the user on the edge of the cell, which normally would perform poorly than the ones closer to the eNodeB, get a better quality of service when the MLB algorithm is running.

In figure 6.18, we can see more clearly the effect on the overall system performance on the downlink. Since the users on the edge are handover to a different cell, the users that remain on the cell would have more resources available and therefore the whole system performs much better.

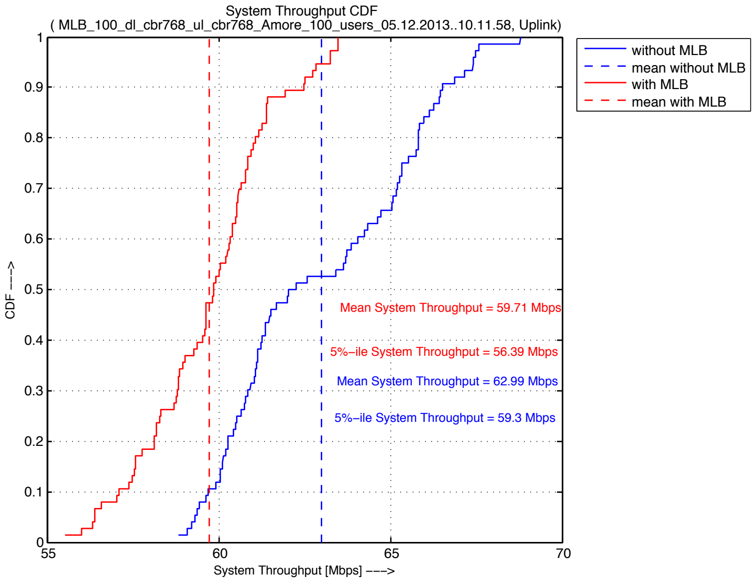


Figure 6.16: UL System Throughput: 100 users CBR768 in DL and UL -

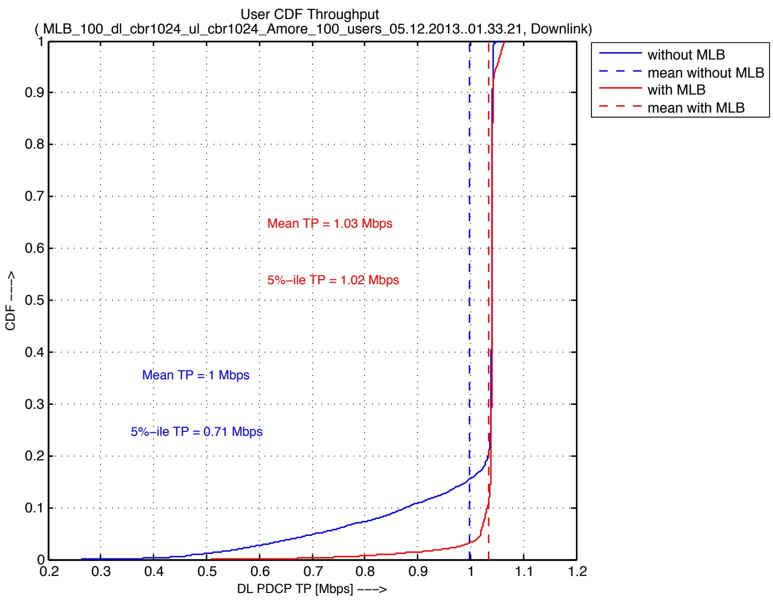


Figure 6.17: DL User Throughput: 100 users CBR1024 in DL and UL -

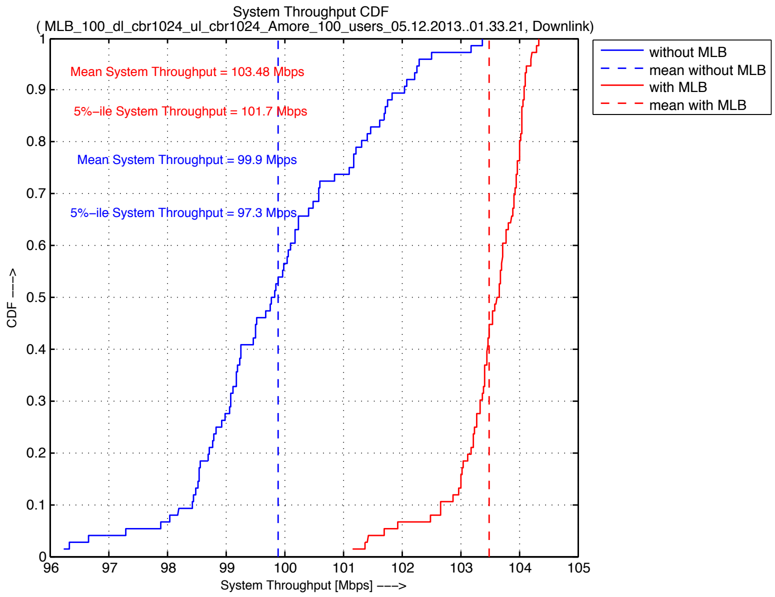


Figure 6.18: DL System Throughput: 100 users CBR1024 in DL and UL -

On the other hand, as already seen in the previous scenario, the mean uplink PDCP throughput and the 5%-ile uplink throughput are reduce when MLB optimizations take place. This is shown in figure 6.19.

This fact is translated to the overall uplink performance as depicted in figure 6.20 where both parameters decrease.

6.3.2.3 100 users simulations: DL CBR 1800, UL CBR 1800

In this scenario, the users are also configured to produce a constant bit rate service type. The user in the center cell have the same service type. All users have a CBR service of 1800 kbps in both, downlink and uplink. In this scenario, we will see that the MLB algorithm runs more frequently than in the previous scenario.

In this scenario, we can see that the MLB algorithm does not make many optimizations thus, the performance of to the user and system throughput stays the same for both uplink and downlink. This happens because the throughput of the users for the expected service type is bigger than the available capacity on the neighbor eNodeBs.

Figures 6.21 and 6.22 depict the downlink PDCP throughput measurements and figures 6.23 and 6.24 the ones for the uplink case.

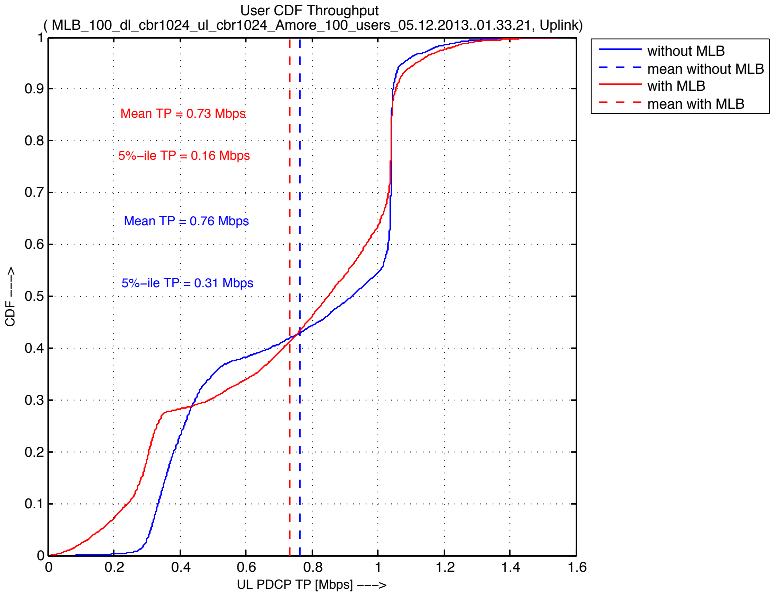


Figure 6.19: UL User Throughput: 100 users CBR1024 in DL and UL -

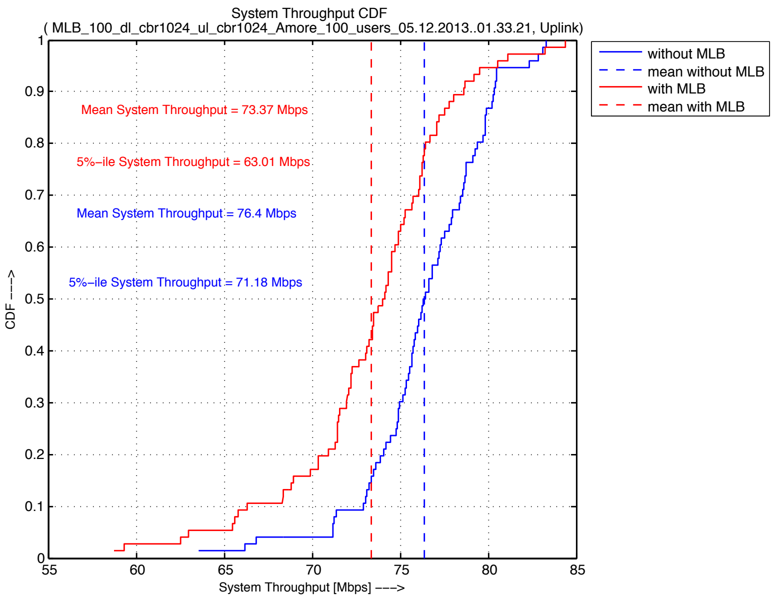


Figure 6.20: UL System Throughput: 100 users CBR1024 in DL and UL -

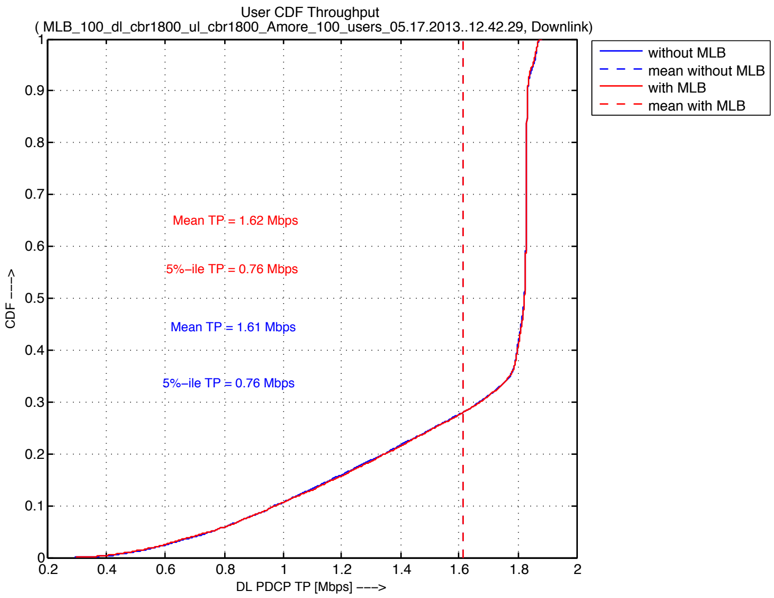


Figure 6.21: DL User Throughput: 100 users CBR1800 in DL and UL -

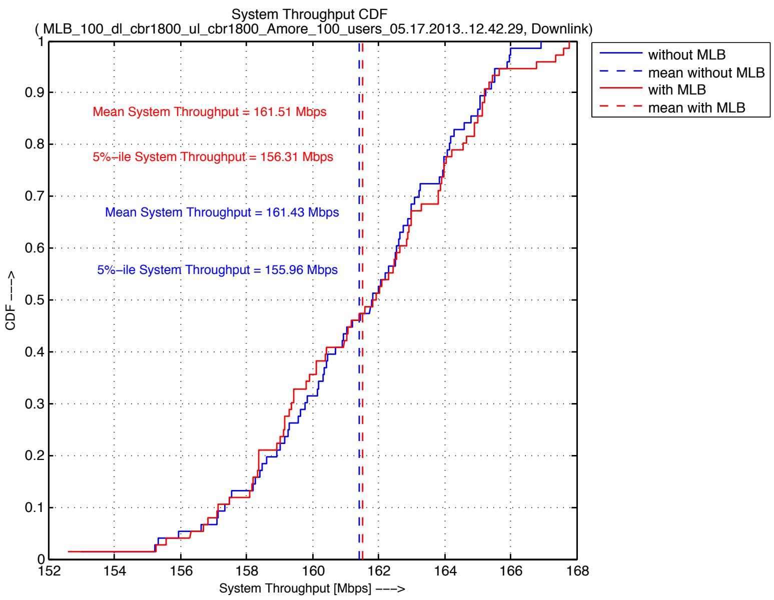


Figure 6.22: DL System Throughput: 100 users CBR1800 in DL and UL -

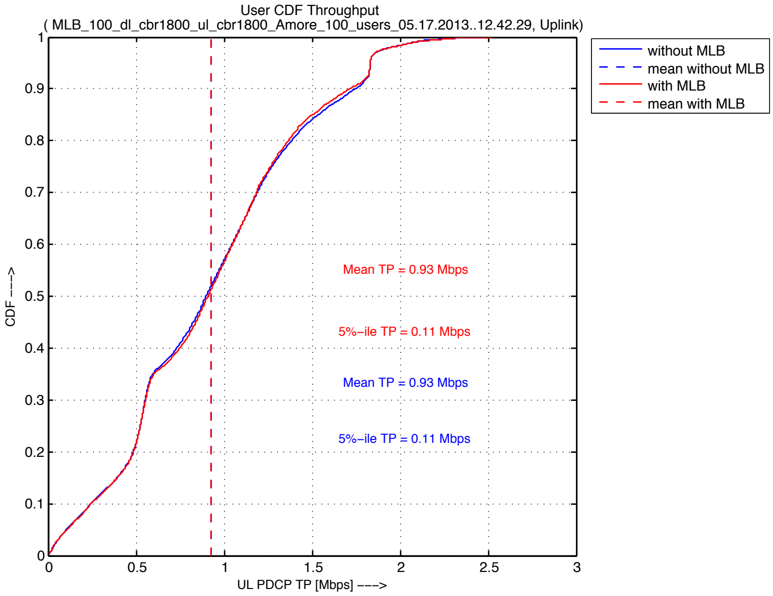


Figure 6.23: UL User Throughput: 100 users CBR 1800 in DL and UL -

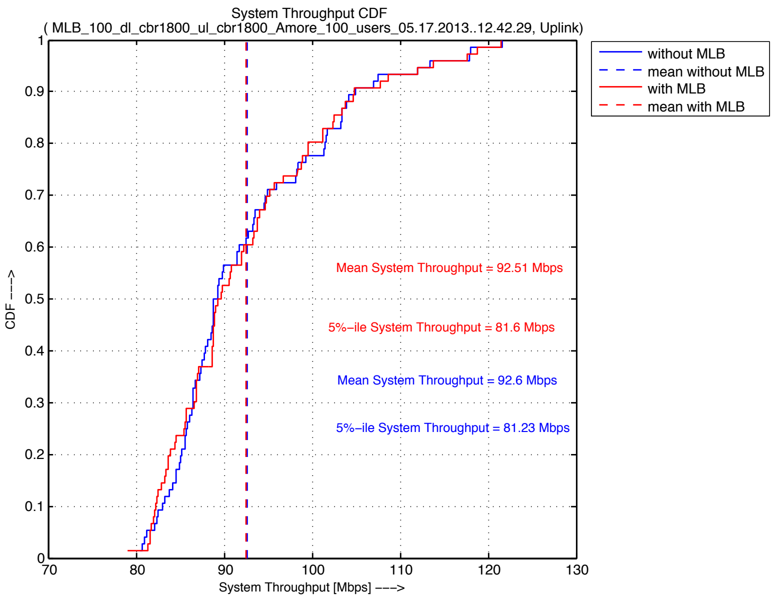


Figure 6.24: UL System Throughput: 100 users CBR1800 in DL and UL -

6.3.3 110 Users Simulations

The 110 user simulations are similar to the previous scenarios simulated. When the overloaded periods of time are small, the enhancements produce to the users and system does not differ much with the case where the MLB algorithm is not present.

Nevertheless, the uplink is performing worst than the case where the algorithm is not present. This is due to the fact that the changes perform to the Handover Offset remain unchanged until further optimizations are performed.

6.3.4 Consideration of Simulations

Now that we have seen the common effects produced by the MLB algorithm in the downlink and uplink, we will compare the results obtained for the scenarios already show and some other scenarios not shown. This scenarios do not appear to avoid repetition of the same behavior where the conclusions can be observed.

6.3.4.1 MLB Gain

MLB Gain is a much more appealing parameter to study since it compares different scenarios with different number of users. This parameter is the relative gain that the user and system gets whenever the MLB algorithm is used. It gives insight on the situations where the algorithm might be more suitable and where it will enhance better the system performance.

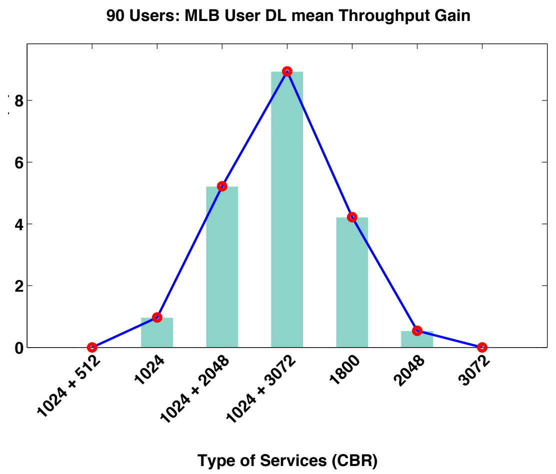


Figure 6.25: 90 users: DL mean user Throughput Gain -

Figure 6.25 and 6.26 give an idea of the performance on the downlink of the MLB algorithm when 90 users are available in the system. We have to remember that 30 of then are in the center cell with is the one being overloaded. The different bars represent the types of services required by the user in the center cell. In this case CBR 1024+512 means that half of the user require a CBR of 1024 kbps and the other half 512 kbps. The service type CBR 1024 means that all the users in the center cell require a CBR service type with 1024 kbps. The same logic applies to the rest of the scenarios. In both figures we can see that the MLB algorithm perform poorly

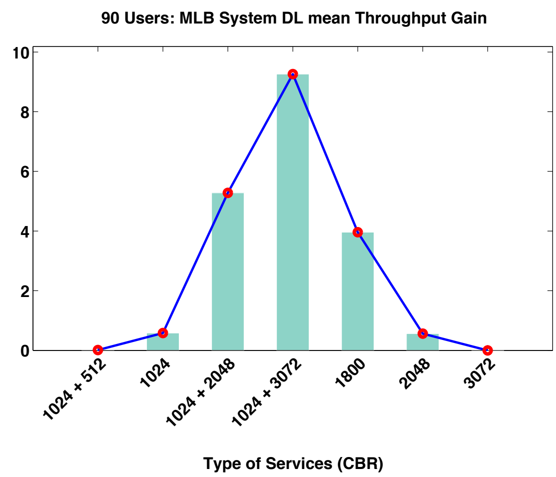


Figure 6.26: 90 users: DL mean System Throughput Gain -

for low service type and increases when the service types requires are higher. This happens because 90 users requiring low service types do not overload the cell and therefore the MLB algorithm is not triggered regularly. The figures also show that when the service is too high the MLB algorithm starts getting trouble obtaining gain with the algorithm. This effect can also be seen in the 5%-ile mean Throughput depicted in figure 6.27 for the user case and figure 6.28 for the system case.

When we increase the number of users to 110, that is 50 users on the center cell and 10 on the others, we can see the same type of behavior as the previously observed. The difference between the one with 90 users and this one with 110 users is that the gain appears to be shifted towards lower service types. Due that with higher number of user the cell overload can be reached easier with lower service types.

Figures 6.29 and 6.30 show the behavior for 110 users.

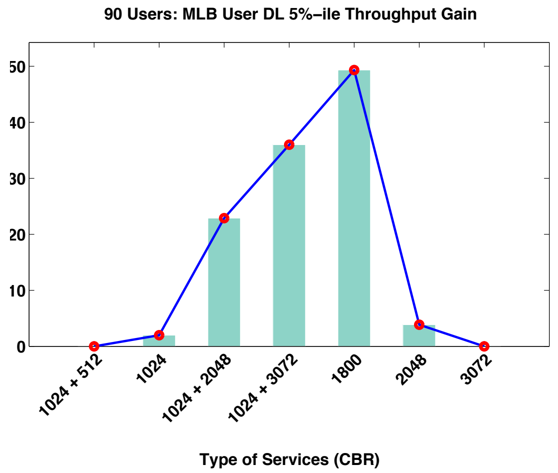


Figure 6.27: 90 users: DL 5%-ile user Throughput Gain -

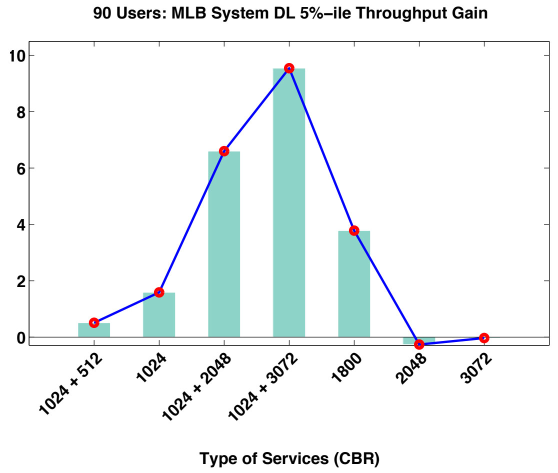


Figure 6.28: 90 users: DL 5%-ile System Throughput Gain -

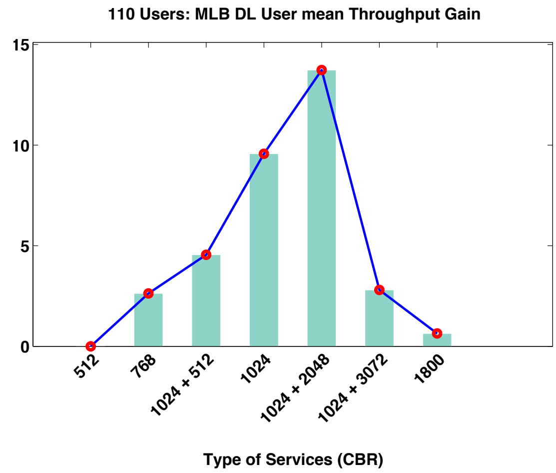


Figure 6.29: 110 users: DL mean user Throughput Gain -

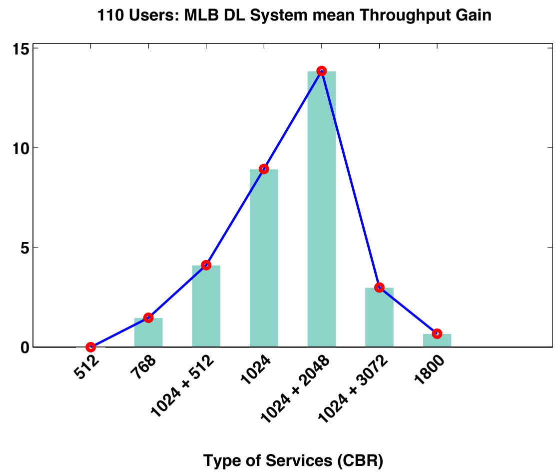


Figure 6.30: 110 users: DL mean System Throughput Gain -

We can also observe greater gain for the scenarios with 110 if we compare the top gains independently of the service required. The reason for this behavior is that a greeter number of users allow for a greater number of decision in the MLB algorithm and normally the more options you have the better the solution applied.

The same shifting to smaller service types is also observed in the 5%-ile throughput for both user and system performance. Since it does not add extra information the plots will not be represented.

On the Uplink side, depicted in figure 6.31 for the 90 users case and in figure 6.32 for the 110 user case, we can see that when the number of users is lower and the MLB algorithm forces the users to handover, the effect is worst for the uplink throughput. When the number of users is higher, as in the 110 user case, the effects of the MLB algorithm on the uplink throughput are smoother for the service types where MLB performs best on the downlink.

We can also observe, as previously in the cumulative distributions functions, that when the MLB algorithm makes less optimizations, thus forcing less handovers, the uplink throughput loses less throughput.

Similarly, the same conclusions can be seen for the 5%-ile uplink Throughput.

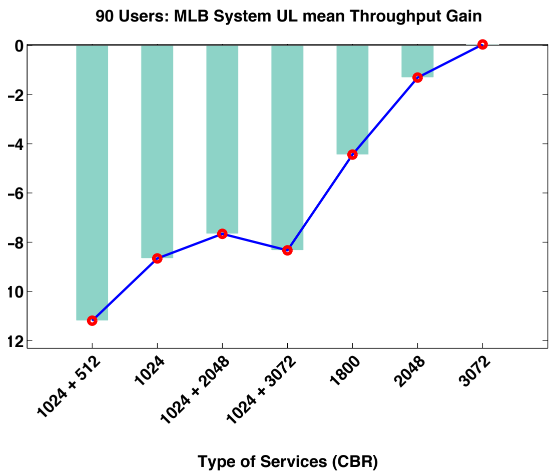


Figure 6.31: 90 users: UL mean System Throughput Gain -

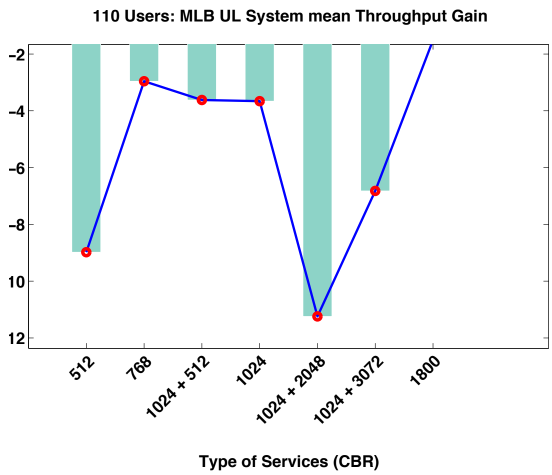


Figure 6.32: 110 users: UL mean System Throughput Gain -

6.3.4.2 Ratios

6.4 Effect of MLB Period on the Performance

In this section we discuss the effects of the deciding the MLB Period on the overall system performance. The MLB period is the lapse of time between 2 consecutive optimizations produced by the MLB algorithm. This parameter can be configured by the network operator to reduce the amount of measurements required by the eNodeB to produce MLB optimizations.

In figure 6.33, we can see the effect on the downlink system throughput when we vary the MLB period. We can observe that the smaller the MLB period is, the better the algorithm perform. As explain earlier, decreasing the MLB period makes the algorithm run more often producing more optimization decisions in a shorter amount of time.

The trade-off here is that at a lower MLB period, the eNodeB needs to asks the user about their measurements to their neighboring cells. This produces a overhead on the system and makes the user make measurements more often than usual reducing their battery life and producing saturation in the uplink control channels when measurements are send to the eNodeB.

On the Uplink, depicted in figure 6.34 the effect is, as already discussed in previous sections, that the mean uplink system throughput and the 5%-ile system throughput are reduced when the MLB periods are smaller.

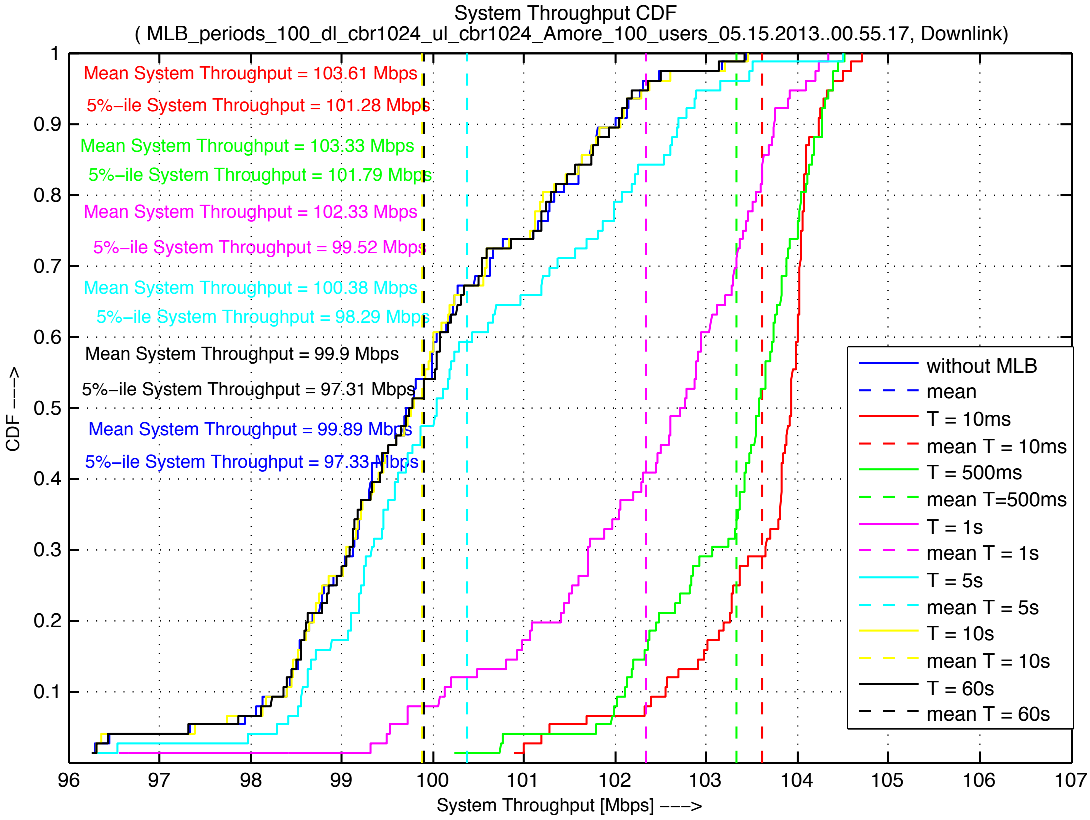


Figure 6.33: DL mean System Throughput for different MLB Periods -

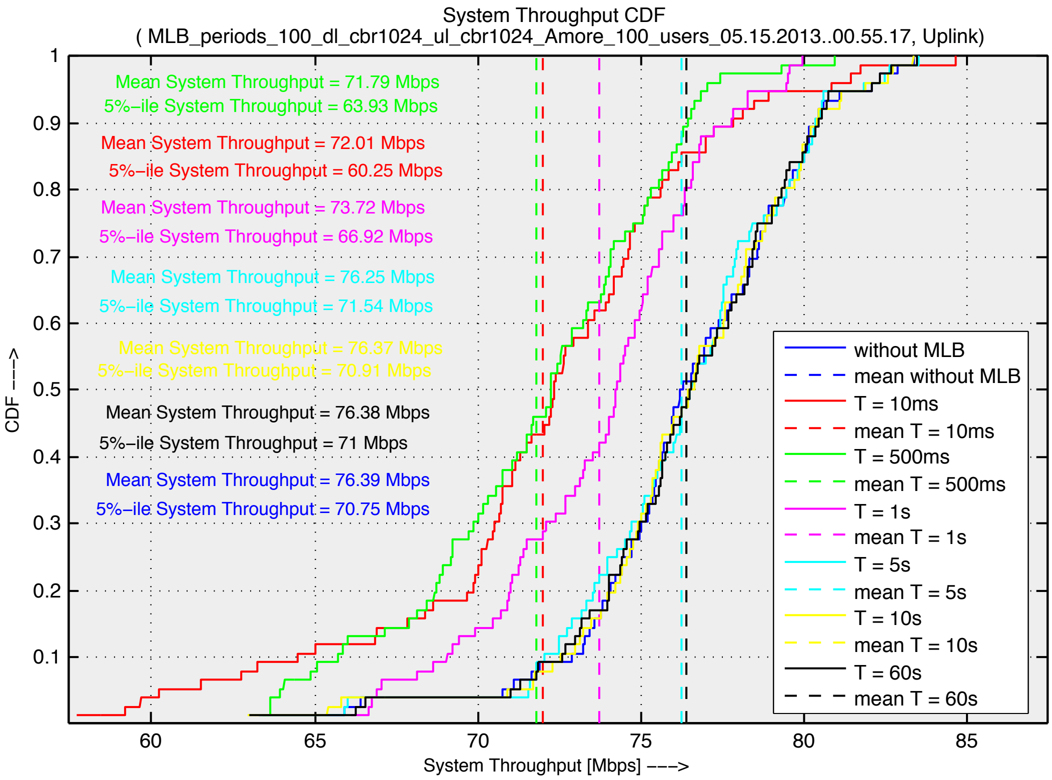


Figure 6.34: UL mean System Throughput for different MLB Periods -

6.5 Effect of MLB Thresholds on the Performance

Whenever the virtual load reaches a certain point, the MLB algorithm starts optimizing the handover offsets of the cells. This point is called MLB Threshold. For the previous simulations, the MLB Threshold was set to the same value. In this section we take a look at the implications of deciding on one MLB Threshold over another.

First, we will take a look at the downlink system throughput for different number of users when selecting different MLB Thresholds. Figure 6.35, shows the case for 100 users with different MLB Thresholds and figure 6.36 shows the same study for 120 users.

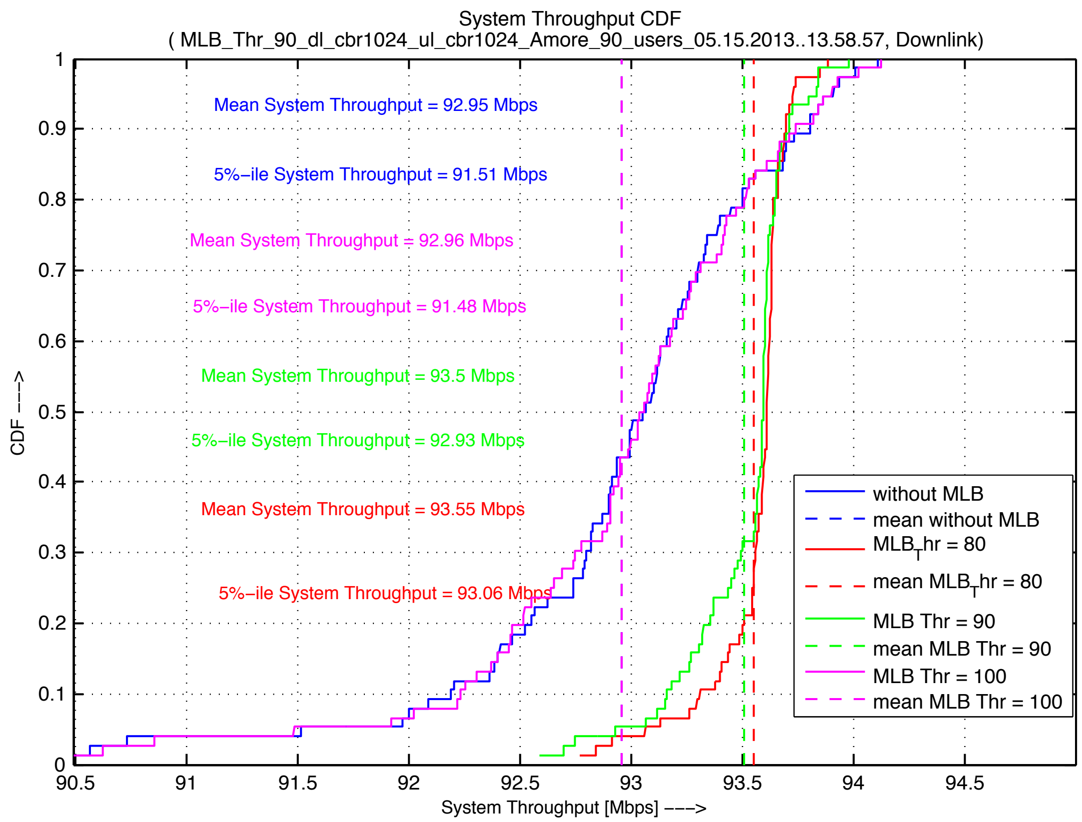


Figure 6.35: Effect of MLB Threshold in the DL System Throughput with 90 users -

In figure 6.35, with 90 users, we can see that the overall downlink system PDCP Throughput performs better with lower MLB Thresholds.

In figure 6.36, with 120 users, the previous statement does not hold as we can see for the case where the MLB Threshold is equal to 90 which perform slightly better then the case where

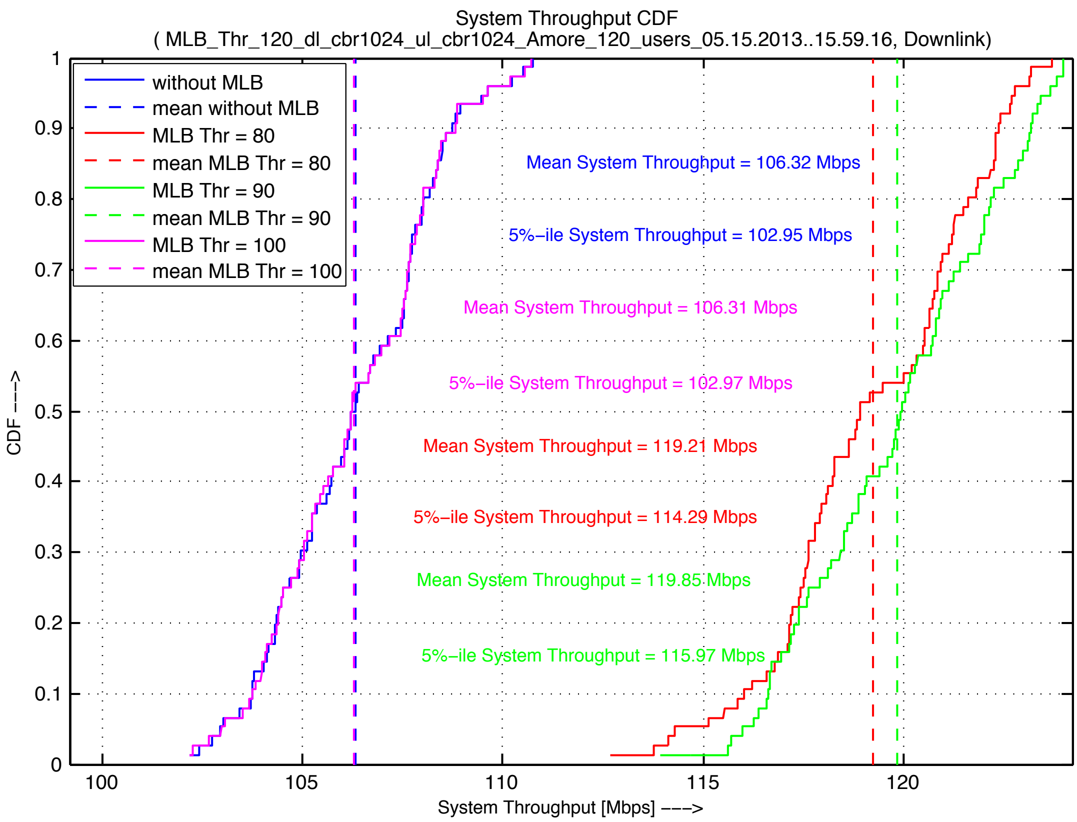


Figure 6.36: Effect of MLB Threshold in the DL System Throughput with 120 users -

the MLB Threshold is equal to 80.

For the MLB Threshold we can conclude that depending on the situation and the number of user the system behaves in different ways. The reason for this behavior is that whenever the MLB Threshold is lower, it allows the MLB algorithm to force more extreme situations than when the MLB is higher. This is due to the fact that the composite available capacity depends on this parameter to signal the available capacity to its neighbor eNodeBs. When the capacity signaled is higher the MLB algorithm can make the PRB estimations on the Neighbor eNodeB in a more rough estimation producing more extreme situations for the users on the edges.

6.6 SINR Fine Tuning

Finally, we will take a look at the effect of tuning the SINR estimations made by the MLB algorithm. The tuning of the SINR depends on the characteristics of the scenario since each scenario is different from another.

In figure 6.37, we can see the results obtained for our scenario and how the downlink system PDCP throughput varies for the different changes in the SINR estimation.

We can see that for the case where we decrease the SINR estimations by 2dB (-2dB) the mean system PDCP throughput and the 5%-ile system throughput performs better than the others.

Additionally, in figure 6.38, we can see that the case where we decrease the SINR estimation by 2dB (-2dB) the mean system PDCP throughput and the 5%-ile system throughput also performs better than the others.

This means that making adjusting the estimations made by the algorithm, the performance of the MLB algorithm will perform much better in both , uplink and downlink.

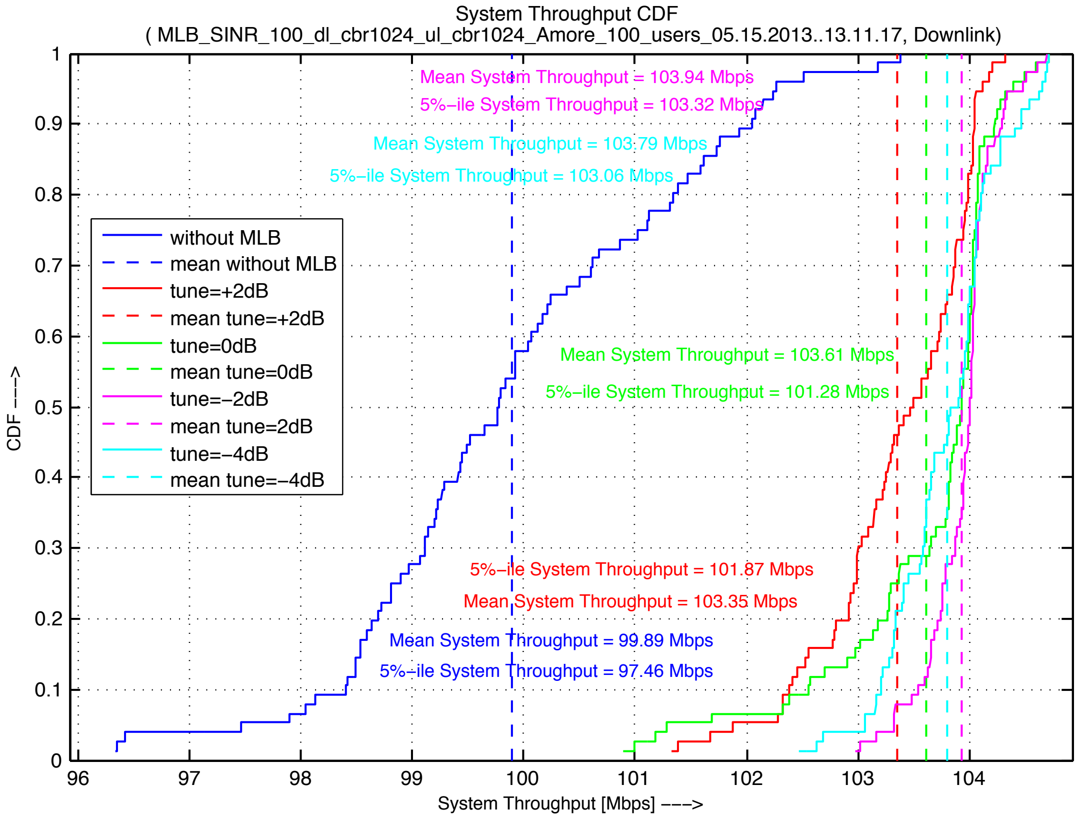


Figure 6.37: Effect of the SINR tuning in the DL System Throughput with 100 users -

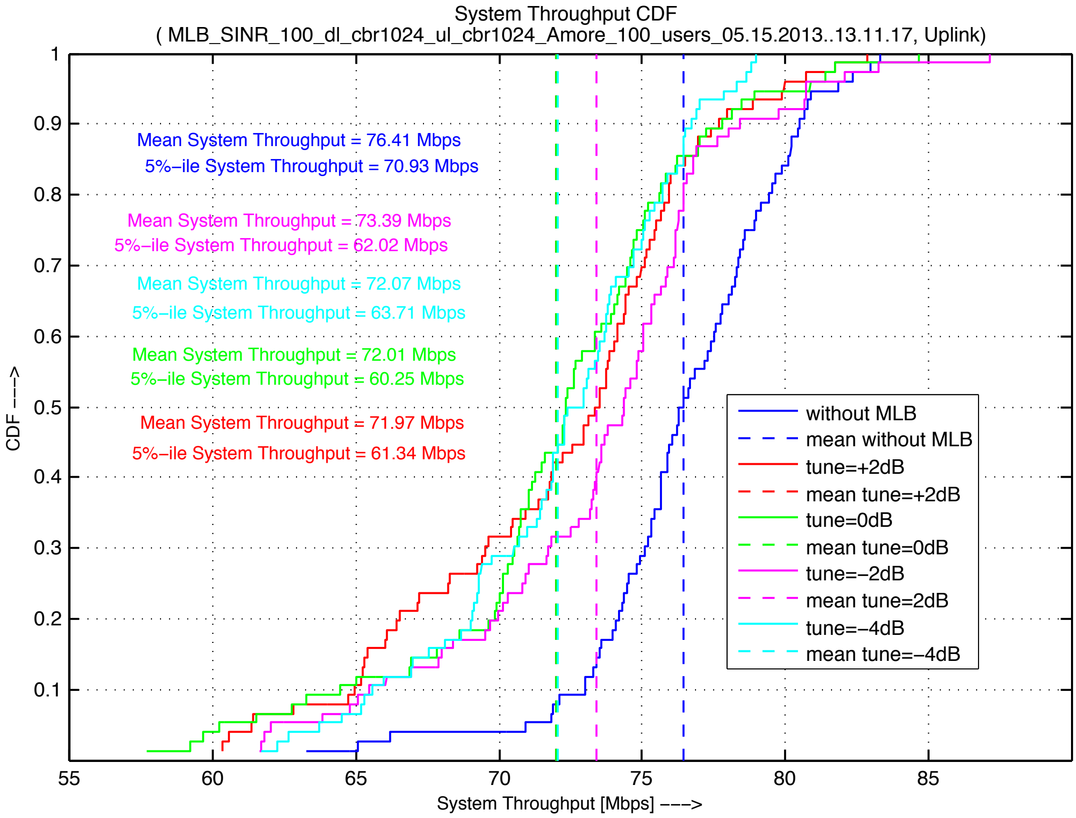


Figure 6.38: Effect of the SINR tuning in the UL System Throughput with 100 users -

7

Conclusion & Future Work

Up to this point we have study the performance of the Mobility Load Balancing algorithm based on the results obtained for the Radio Network simulator called AMoRE. We have seen that certain trade-offs should be taken into account when using the proposed solution in real scenarios.

In the downlink, the mean PDCP throughput and the 5%-ile PDCP Throughput in the scenarios proposed increase in the cases where overload occurs in the cell. This increase on the downlink comes with a decrease on the Uplink due to the increase in interference due that the MLB algorithm forces changes to the Handover Offset that produce the user close to the edge to handover to a neighbor eNodeB with worst conditions than normal producing more interference to the original cell.

We study the influence of the number of users to be considered by the MLB Algorithm in the handover offset change decision process. Whenever more users were available to offload, the MLB performed better since the number of possible solutions increased.

The different traffic types requested by the user also influence the behavior of the algorithm and the enhancements in the overall system performance. When the traffics are smaller the eNodeB finds easier a solution to the overload than when the service is bigger. In some case, where the service types are too big the MLB algorithm does not enhance the overall system performance.

Thus, when the number of user increases the maximum gain produced on the throughput shifts to lower service types.

While setting the MLB periods, certain trade-offs need to be made. If the periods are small the gain produced in the overall system increases but it leads to overhead and saturation of the control channels, due to the increasing amount of measurements, and reduce the battery life of the users.

The MLB Threshold should be set depending on the type of scenario taking place. This way the overall system performance can be increase if the parameter is chosen appropriately.

And finally, the SINR estimation taking place in the MLB algorithm can be fine tuned to produce better SINR estimations and therefore better performance in both, uplink and downlink.

7.1 Future Work

In this thesis, many of the parameters that can be adjust have been taken into consideration. Nevertheless, the algorithm could be enhanced in the future.

A further study of the MLB algorithm could lead to the introduction of Uplink Power Control in combination with the algorithm. This would help to control and optimize the interference produced in the uplink and allow for better perform in the uplink PDCP throughput.

Furthermore, it would be interesting to consider a recovery system that allowed the cell to return to its original and natural conditions. That is, adjusting the Handover Offset whenever the overload situation has been managed. In the implementation proposed, this solution has not been implemented and it might be the cause for loss of performance in the 5%-ile downlink PDCP throughput.

References

[1] ERIK DAHLMAN, STEFAN PARKVALL, AND JOHAN SKOLD. *LTE / LTE-Advanced for Mobile Broadband*. Academic Press, 2011. 5, 12, 13, 28

[2] STEFANIA SESIA, ISSAM TOUFIK, AND MATTHEW BAKER. *LTE - The UMTS Long Term Evolution: From Theory to Practice*. isbn: 978-0-470-66025-6. Wiley Publishing, 2nd edition, 2011. 6, 9, 13, 19, 20, 24, 28

[3] 3GPP. **(E-UTRAN); Overall description; Stage 2 (Release 9)**. TS 36.300, 3rd Generation Partnership Project (3GPP), 2011. 7, 14, 25, 50

[4] MARTIN SAUTER. *From GSM to LTE: An Introduction to Mobile Networks and Mobile Broadband*. isbn: 9780470667118. Wiley Publishing, 1st edition, 2011. 11

[5] C GESSNER. **UMTS Long Term Evolution (LTE) Technology Introduction**. Technical report, Rohde & Schwarz, 2008. 11

[6] PIERRE LESCUYER AND THIERRY LUCIDARME. *Evolved Packet System (EPS): The LTE and SAE Evolution of 3G UMTS*. Wiley Publishing, 1st edition, 2008. 12, 13

[7] SIMON CHANDLER. **4M Wireless LTE protocol stack**. Technical report, http://www.3g.co.uk, 2008 [cited May 2012]. 16, 19

[8] 3GPP. **(E-UTRAN); Radio Resource Control protocol specification (Release 9)**. TS 36.331, 3rd Generation Partnership Project (3GPP), 2011. 16, 25

[9] HARRI HOLMA AND ANTTI TOSKALA. *LTE for UMTS: Evolution to LTE-Advanced*. John Wiley & Sons, 2nd edition, 2011. 20, 28

[10] 3GPP. **(E-UTRA); X2 application protocol (X2AP) (Release 9)**. TS 36.423, 3rd Generation Partnership Project (3GPP), 2011. 23, 24, 59

[11] 3GPP. **(E-UTRAN); User Equipment (UE) procedures in idle mode (Release 9)**. TS 36.304, 3rd Generation Partnership Project (3GPP), 2011. 25

[12] 3GPP. **(E-UTRAN); Requirements for support of radio resource management (Release 9)**. TS 36.133, 3rd Generation Partnership Project (3GPP), 2011. 25

[13] SEPPO HÄMÄLÄINEN, HENNING SANNECK, AND CINZIA SARTORI. *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*. Wiley Publishing, 1st edition, 2011. 29, 31, 41, 47

[14] HONGLIN HU, JIAN ZHANG, XIAOYING ZHENG, YANG YANG, AND PING WU. **Self-configuration and self-optimization for LTE networks**. *Communications Magazine, IEEE*, **48**(2):94 –100, february 2010. 29, 30

[15] FRANK LEHSER. **NGMN Use Cases related to Self Organising Network, Overall Description**. Technical report, Next Generation Mobile Networks, 2008. 32, 33

[16] EIKO SEIDEL AND SUJUAN FENG. **Self-Organizing Networks (SON) in 3GPP Long Term Evolution , White Paper**. Technical report, Nomor Research GmbH, May 2008. 35, 36

[17] NSN. **Self Organizing Networks (SON), White Paper**. Technical report, Nokia Siemens Networks, 2009. 36, 37

[18] M. DOTTLING AND I. VIERING. **Challenges in mobile network operation: Towards self-optimizing networks**. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 3609–3612. Ieee, 2009. 37, 51

[19] JUAN RAMIRO AND KHALID HAMIED. *Self-Organizing Networks (SON): Self-Planning, Self-Optimization and Self-Healing for GSM, UMTS and LTE*. Wiley Publishing, 1st edition, 2012. 39

[20] A. LOBINGER, S. STEFANSKI, T. JANSEN, AND I. BALAN. **Load balancing in downlink LTE self-optimizing networks**. In *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pages 1–5. Ieee, 2010. 40, 46

[21] 3GPP. **E-UTRA; Layer 2 - Measurements (Release 9)**. TS 36.314, 3rd Generation Partnership Project (3GPP), 2011. 43

[22] 3GPP. **(E-UTRA); Physical layer; Measurements (Release 9)**. TS 36.214, 3rd Generation Partnership Project (3GPP), 2011. 43

[23] I. VIERING, M. DOTTLING, AND A. LOBINGER. **A mathematical perspective of self-optimizing wireless networks**. In *Communications, 2009. ICC'09. IEEE International Conference on*, pages 1–6. IEEE, 2009. 45

[24] 3GPP. **Self-configuring and self-optimizing network (SON) use cases and solutions (Release 9)**. TS 36.902, 3rd Generation Partnership Project (3GPP), 2011. 47, 48

[25] LC SCHMELZ, JL VAN DEN BERG, R. LITJENS, K. ZETTERBERG, M. AMIRIJOO, K. SPAEY, I. BALAN, N. SCULLY, AND S. STEFANSKI. **Self-organisation in wireless networks–use cases and their interrelation**. In *WWRF Meeting*, **22**, 2009. 47, 48

[26] A. AWADA, B. WEGMANN, I. VIERING, AND A. KLEIN. **A game-theoretic approach to load balancing in cellular radio networks**. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*, pages 1184–1189. IEEE, 2010. 56

Declaration

I herewith declare that I have produced this paper without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This paper has not previously been presented in identical or similar form to any other Spanish or foreign examination board.

The thesis work was conducted from November 2011 to July 2012 under the supervision of Abdallah Bou Saleh, PhD., Senior Researcher at Nomor Research GmbH.

Munich,