

MANAGING ENERGY EFFICIENCY IN BASE STATIONS
FOR GREEN CELLULAR NETWORKS

by

Weston Mwashita

47955864

submitted in accordance with the requirements for the
degree of

Master of Technology in Electrical Engineering

At the

UNIVERSITY OF SOUTH AFRICA

Supervisor: Prof. M.O. Ohanga

2016

“We are the first generation to feel the impact of climate change and the last generation that can do something about it.”

President Barack Obama, quoting one of America’s governors, when announcing the Clean Power Plan, on 3rd August 2015, at White House, East room.

DECLARATION

I, Weston Mwashita, student number 47955864, hereby declare that the dissertation entitled:

Managing Energy Efficiency in Base Stations for Green Cellular Networks

is the result of my own research and presents my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references. I further declare that I have not previously submitted this work, or part of it, for examination at UNISA for another qualification or at any other higher education institution.



.....
SIGNATURE

10th June 2016

.....
DATE

ABSTRACT

With the astronomical increase in cellular traffic, there is need to cut down on the power consumption so as to reduce CO₂ emissions and in the process lowering network operational expenditure (OPEX). In this thesis, one method that can be used to lower a Base Station energy consumption is proposed. Traditional cellular networks are designed to offer maximum coverage and connectivity for peak traffic. This is not energy efficient since a lot of energy will go to waste during the time cellular traffic is very low. The scheme that was developed, identifies Base Stations that have very low traffic loads and User Equipment that can all be transferred to neighbouring Base Stations and put the Base Stations to sleep for as long as necessary to save energy and to maintain Quality of Services (QoS) at an acceptable level. The Next Generation Networks (specifically 5G) will be heterogeneous networks as heterogeneous are a promising solution in increasing network performance especially in providing indoor and cell-edges coverage. The solution that was developed in this thesis was specifically designed to work with heterogeneous networks and its performance was also tested on heterogeneous networks. OMNeT++ V4.6 together with INET 2.3.0 and SimuLTE 9.1 were used for the validation of the proposed scheme. After extensive simulations were carried out, it was concluded that some Base Stations in a cellular network, can be put to sleep during the time that cellular traffic is low without compromising the Quality of Service. End-to-end delay, sum throughputs, queue length and Channel Quality Indicator were some of the performance metrics that were used to check whether the developed scheme did not reduce the QoS of a network.

ACKNOWLEDGEMENTS

This thesis could not have been produced were it not for my academic supervisor Prof. Marcel Odhiambo Ohanga whose advice and guidance were very invaluable throughout all the phases of this project. I am highly privileged to have worked under his supervision and I am totally convinced that he read and scrutinised every word in all the reports that I submitted to him. The timeous feedback allowed me to complete the project within the stipulated time. He could even engage me during weekends and holidays. From him, I got a number of documentations which were in line with the research work that I was engaged in and which I found so helpful. From the bottom of my heart, I thank him. I also wish to express my deepest appreciation to Giovanni Nardini, who is a member of the Computer Networking Group (the group that developed SimuLTE-an LTE User Plane Simulation model for INET and OMNeT++) at Pisa University in Italy, for his support on use of SimuLTE during the simulation phase of the project. I am also highly thankful to my family especially my wife Ndinaani, who gave me encouragement and moral support throughout the entire period that I was busy with the research work. Finally, I would like to thank the staff at Electrical Department and College of Graduate Studies at UNISA for all their support and according me the privilege to study at UNISA.

TABLE OF CONTENTS

DECLARATION.....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	iv
LIST OF TABLES.....	ix
LIST OF FIGURES.....	x
LIST OF ABBREVIATIONS AND ACRONYMS.....	xi
CHAPTER 1: INTRODUCTION.....	1
1.1 System Description and Problem Formulation.....	6
1.2 Research Aims and Objectives.....	7
1.2.1 Aim of the Research Project.....	8
1.2.2 Purpose of the Research Project.....	8
1.2.3 Objectives of the Research Work.....	8
1.4 Research Methodology.....	8
1.5 Contributions of the Research Work.....	9
1.6 Thesis Organisation.....	10
CHAPTER 2: REVIEW OF BASE STATION SLEEPING ALGORITHMS.....	11
2.1 Introduction.....	11
2.2 BS Sleeping Technologies.....	11
2.2.1 Distance-Aware Base Station Switching ON/OFF Scheme (DSO).....	12
2.2.2 Cell Zooming Technology.....	12
2.2.3 Cell Wilting and Blossoming.....	14
2.2.4 BS Sleep Control for Heterogeneously Powered BSs.....	14
2.2.5 Switching ON/OFF Based Energy Saving Algorithm.....	15
2.2.6 User-Aware Cell Switching ON/OFF Algorithm.....	15
2.3 Conclusion.....	16
CHAPTER 3: AN OVERVIEW OF HETEROGENEOUS NETWORKS.....	18
3.1 Introduction.....	18

3.2	Heterogeneous Networks.....	18
3.2.1	Femto Cells.....	20
3.2.2	Pico Cells.....	21
3.2.3	Micro Cells.....	21
3.2.4	Macro Cells.....	21
3.3	Conclusion.....	22
CHAPTER 4: ANALYSIS OF THE PROPOSED SOLUTION.....		23
4.1	Introduction.....	23
4.2	The Problem Statement.....	23
4.3	The Proposed Solution.....	23
4.4	The Breakdown of the Proposed Solution.....	24
4.5	The Switching ON/OFF Algorithm.....	25
4.6	The BS-UE Association Algorithm.....	27
4.6.1	Power Per Unit Area.....	27
4.6.2	Specific Energy, SE.....	27
4.7	Conclusion.....	30
CHAPTER 5: DESIGN OF THE PROPOSED SOLUTION.....		32
5.1	Introduction.....	32
5.2	The BS Sleeping Algorithm.....	32
5.2.1	List of Notations.....	35
5.2.2	The BS Sleeping Pseudo Code.....	35
5.3	The BS-UE Association Algorithm.....	36
5.3.1	The BS-UE Association Pseudo Code.....	38
5.4	Relay Nodes.....	38
5.4.1	Relay Nodes Switching ON/OFF Pseudo Code.....	39
5.5	The Handover Process.....	39
5.5.1	The Detailed Handover Procedure.....	40
5.6	Conclusion.....	41

CHAPTER 6: IMPLEMENTATION, TEST PROCEDURES AND TEST RESULTS....	42
6.1 Introduction.....	42
6.2 Simulation Environment.....	42
6.3 The Simulation Tests.....	46
6.3.1 End-to-end Delay.....	48
6.3.2 Queue Length.....	49
6.3.3 Channel Quality Indicator.....	50
6.3.4 UEs' Sum Throughputs.....	51
6.3.5 Energy Consumed.....	52
6.4 Conclusion.....	53
CHAPTER 7: ANALYSIS OF THE TEST RESULTS.....	54
7.1 Introduction.....	54
7.2 End-to-end Delay.....	54
7.3 Packet Loss.....	55
7.4 Queue Length.....	56
7.5 Channel Quality Indicator.....	56
7.6 Throughput.....	57
7.7 Power Consumption.....	58
7.8 Conclusion.....	58
CHAPTER 8: CONCLUSION.....	59
8.1 Introduction.....	59
8.2 Implications of the Research.....	59
8.3 Applications of the Research.....	60
8.4 Recommendations for Further Work.....	60
8.5 Conclusion.....	61
REFERENCES.....	62
Appendix A List of publications.....	65
Appendix B WindhoekCentral.ned.....	66
Appendix C The BaselineCase-omnetpp.ini.....	70

Appendix D	multi.xml.....	75
Appendix E	The BS sleeping C++ application.....	76
Appendix F	The handover C++ application.....	86

LIST OF TABLES

Table 1.1	CO ₂ emissions of various electricity generation technologies in Japan, Sweden and Finland.....	2
Table 2.1	LTE BS transceiver power consumption, 2012 State-of-the-art estimation for different BS types.....	20
Table 6.1	Simulation parameters.....	47
Table 6.2	Simulation tests and results obtained.....	48
Table 6.3	End-to-end delay in seconds before and after some BSs were switched Off....	48
Table 6.4	Queue length before and after some BSs were switched OFF.....	49
Table 6.5	CQIs of 17 UEs before and after switching OFF some BSs.....	51
Table 6.6	Sum throughput before and after switching OFF some BSs.....	52
Table 6.7	Total transmitted energy as BSs are switched OFF.....	52

LIST OF FIGURES

Figure 1.1	Mobile gadgets and their predicted growth	1
Figure 1.2	Different sources of energy.....	2
Figure 1.3	Energy efficiencies enhanced by Green Touch Innovations.....	3
Figure 1.4	Energy consumption compositions of a mobile operator.....	4
Figure 1.5	energy use- gas and electricity.....	5
Figure 1.6	System components of a Base Station.....	5
Figure 1.7	Variation of mobile operators' OPEX with time.....	9
Figure 2.1	Zooming in and out by reducing transmit powers and adjusting antenna tilt.....	13
Figure 3.1	A HetNet architecture.....	18
Figure 3.2	3GPP Release 12 work programme.....	19
Figure 3.3	Coverage provided by femtocells in a stadium.....	21
Figure 5.1	A 7-cell cluster.....	32
Figure 5.2	The flow diagram of the BS sleeping algorithm.....	34
Figure 5.3	The BS-UE association flow diagram.....	37
Figure 5.4	A donor cell with a relay node inside.....	38
Figure 5.5	The handover procedure.....	40
Figure 6.1	OMNeT++ module connections.....	42
Figure 6.2	Modules making up the UEs and BSs	43
Figure 6.3	The experimental set up.....	44
Figure 6.4	A screenshot showing BSs exchanging messages.....	45
Figure 6.5	A screenshot showing network elements exchanging messages.....	47
Figure 6.6	Mean VoIP packets end-to-end delay.....	49
Figure 6.7	Queue length at the main server.....	50
Figure 6.8	CQIs for 17 UEs.....	51
Figure 6.9	UEs' sum throughputs	52
Figure 6.10	Energy consumed versus number of BSs switched OFF.....	53
Figure 7.1	Teleconferencing using mobile devices.....	54
Figure 7.2	Cellular traffic variation during a 24-hour weekday.....	55

LIST OF ABBREVIATIONS AND ACRONYMS

3GPP	3 rd Generation Partnership Project
5G	Fifth Generation mobile networks
BS	Base station
BSs	Base stations
CAPEX	Capital Expenses
CO ₂	Carbon dioxide
CoMP	Coordinated Multi-Point
CQI	Channel Quality Indicator
CTSO	Clear to switch OFF
DeNB	Donor eNB
EARTH	Energy Aware Radio Network Technology
E ³ F	Energy Efficiency Evaluation Framework
GUI	Graphical User Interface
HetNet	Heterogeneous Network
HSDPA	High Speed Downlink Packet Access
ICT	Information and Communication Technology
IEA	International Energy Agency
LTE	Long Term Evolution
LTE-A	Long Term Evolution- Advanced
MIMO	Multiple Input Multiple Output
MSC	Mobile switching centre
NED	Network Description
OPEX	Operational Expenses
PA	Power amplifier
QoS	Quality of Service
RN	Relay node
RTSO	Request to switch OFF
SE	specific energy
S/N	Signal to noise ratio
SINR	Signal to Interference plus Noise Ratio

TTI	Transmission Time Interval
UE	User Equipment
VoIP	Voice over the Internet Protocol

CHAPTER ONE: INTRODUCTION

Of late, the telecommunications sector has witnessed an explosive growth [1] in mobile cellular systems as mobile data services are fast becoming necessities to many members of the society in peoples' day to day lives. Many people are drooling to own network devices with which they can use to connect to ubiquitous wireless and Internet services from the various mobile service providers. In [2], CISCO predicts that globally, mobile devices and connections will grow to 12 billion by year 2021. Authors in [3] predict that by 2020, over thirty billion things with over two hundred billion intermittent connections will be in existence. Many people now want to use gadgets like tablets and smart phones to access the Internet. Figure 1.1 shows this very high growth as predicted by CISCO.

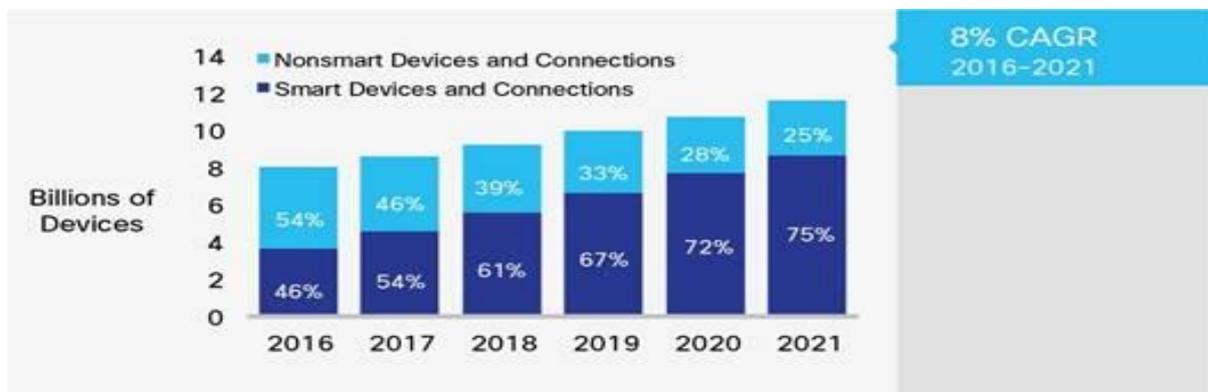


Figure1.1: Mobile gadgets and their predicted growth [2]

China Mobile, one network provider, has been doubling its number of Base Stations (BS) [4] so that they can provide better network coverage and capacity. This rapid growth is forcing network operators to deploy more and more BSs per unit area. This huge increase in the number of BSs has resulted in a corresponding increase in energy consumption and CO₂ emissions especially in remote areas that rely on diesel generators for their power requirements [5].

Some network operators have started using power sources like solar and wind generators. These sources are weather driven and as such are highly unpredictable, just like the weather, and this makes the amount of generated energy to be also unpredictable. The electrical grid is the most reliable source of power supply for BSs. The problem is that most of the electricity generation methods produce the unwanted CO₂ emissions. Only a minute number of the generation sources/methods does not produce the emissions.

Most people think that the nuclear generation method is clean but this method produces CO₂ emissions during the uranium enrichment process. Sims et al. in [5] estimate the emissions to be in the region of 650Kg CO₂ /MWh. Japan’s Central Research Institute of the Electric Power Industry carried out a research that resulted in a life cycle CO₂ emission figures for the different electricity generation methods and the figures are shown in Table 1.1.

Table 1.1: CO₂ emissions of various electricity generation technologies in Sweden, Finland and, Japan [5]

Electricity Generation Technology	Japan g/kWh CO2	Sweden g/kWh CO2	Finland g/kWh CO2
Coal	975	980	894
Gas Thermal	608	1170(peak load reserve)	-
Gas combined cycle	519	450	472
Solar Voltaic	53	50	95
Wind	29	5.5	14
Nuclear	22	6	10-26
Hydro	11	3	-

The British Energy’s Torness Nuclear research carried out in 2002 [5] puts the figure at 5.05g/KWh for emissions coming from the uranium enrichment process. From this data, it can be seen that some generation processes that do not seem to produce CO₂ emissions actually produce CO₂ emissions. CO₂ emissions are also produced in the manufacturing processes of solar panels. Table 1.1 shows that coal, gas thermal and gas combined cycle, produce the highest percentage of CO₂ emissions.

As of 2010, according to statistics that was made available by the International Energy Agency (IEA), slightly over 81% [6] of the world-wide energy consumption comes from oil, gas and coal and these are sources of energy that have the highest CO₂ emissions. Figure 1.2 was part of the statistics that was published by the International Energy Agency that shows world-wide energy consumption.

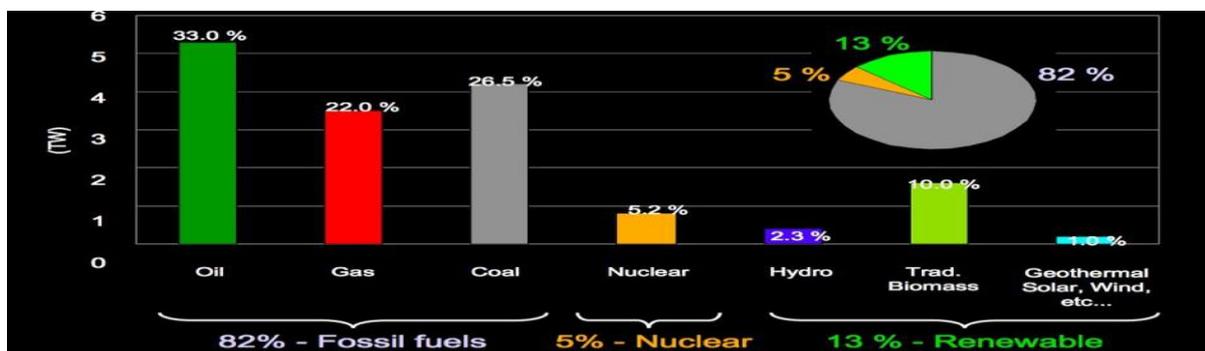


Figure 1.2: Different sources of energy [6]

It has been estimated by scientists that ICT will be responsible for 3% [7] of all global emissions by year 2020 and it can also be noted that of late, most ICT systems are going wireless. This 3% is the same as the percentage of CO₂ greenhouse gas emissions that is contributed by all airlines combined. This is quite substantial.

The unprecedented mobile cellular growth is not only poised to affect the environment in the form of CO₂ emissions, but operational expenditure, commonly known as OPEX, substantially increases. Authors in [4] reported that a total cellular network OPEX of USD22 billion was realised in 2013 alone. This has strongly motivated researchers from industry to join hands with academia to undertake research projects on what has been termed “Green Cellular Communication” [8]. Green Cellular Communication is the practice of making use of energy efficient systems in cellular networks. It involves minimising resources wherever and whenever possible and in the process reducing the amount of CO₂ greenhouse gas emissions. The EARTH (Energy Aware Radio network technology) [9] under the European Framework Program 7, was a big European research project which was composed of 15 of which 10 countries were tasked to focus on energy efficiency for the next generation mobile networks.

Green Touch [10] is another consortium of leading ICT industry, academic and non-governmental experts that is dedicated to reducing carbon footprint of ICT devices. Their intention was to deliver an architecture, specifications and technologies by year 2015. Figure 1.3 graphically outlines Green Touch’s vision.

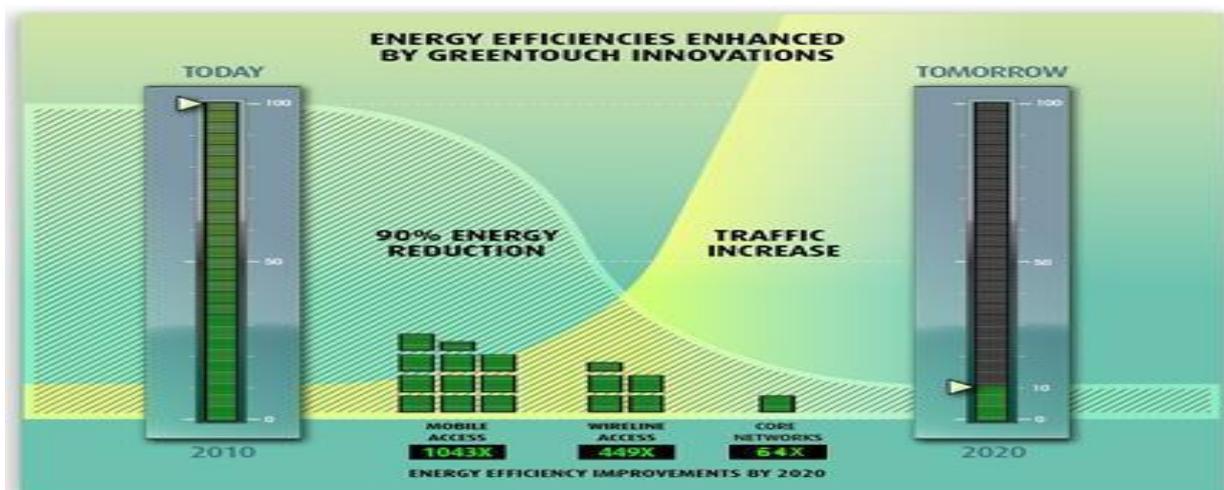


Figure 1.3: Energy efficiencies enhanced by Green Touch innovations [10]

Several other organisations and independent researchers are currently working on Green Cellular networks technologies in a bid to come up with technologies that greatly reduce energy consumed by cellular network elements. There has been quite a number of methods that have been proposed by both academia and industrialists that can be used in cellular networks to increase energy efficiency. Improvement of RF efficiency and reduction of antenna losses at the same time maximizing antenna input is one method that has been introduced [3]. A technology that is known as MIMO (Multiple Input and Multiple Output) was proposed in [11]. The technology uses BSs and user devices that have multiple transmitting and receiving antennas. MIMO systems have high spectral efficiency and an increased data rate. BSs that are powered by wind and solar energy [8] have also been designed. Base station sleeping is also one solution that has been advanced by several researchers.

The BS, as stated by Wu et al. in [12], is the most energy intensive component of a cellular network which accounts for a significant portion of the energy budget of a network. According to [12], [13], 60% to 80% of the total energy consumption in a cellular network, takes place in BSs. Blume et al. in [14], put the BS energy consumption close to 60% as shown in Figure 1.4.

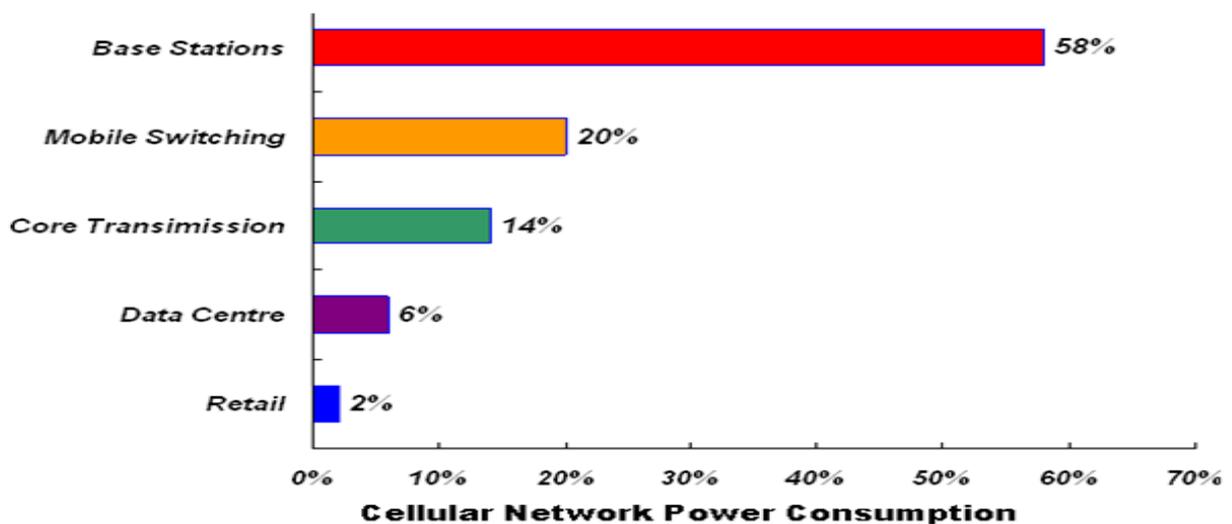


Figure 1.4: Energy consumption composition of a mobile operator [14]

Vodafone-UK in its 2008 corporate responsibility report produced the graph in Figure 1.5 that shows network energy usage in UK from 2004 to 2008 [15].

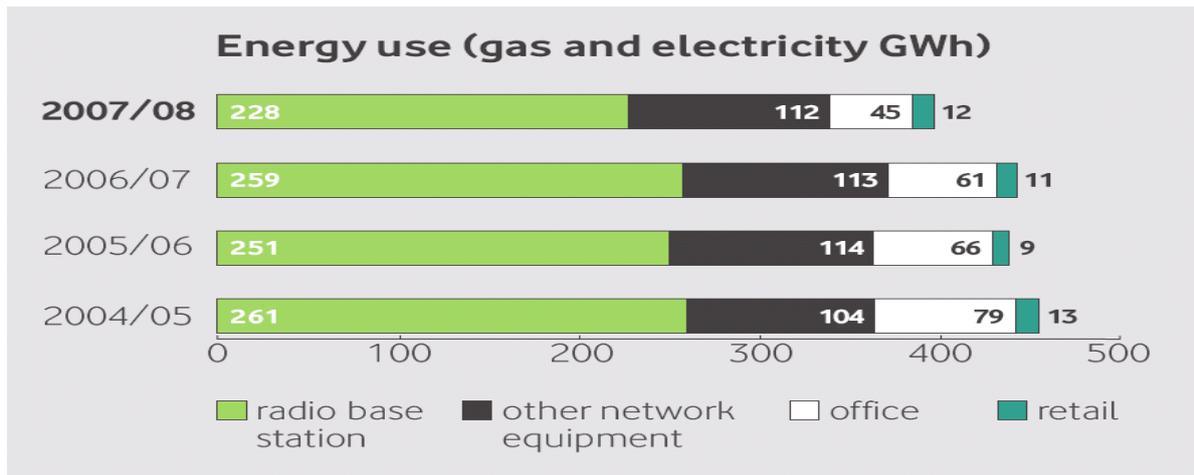


Figure: 1.5: Energy use-gas and electricity (GWh) [15]

It can be observed from Figure 1.5, that most of the power in a mobile cellular network is consumed by BSs. The main energy consumers within BSs are power amplifiers. Figure 1.6 shows the components within a BS that are the chief energy consumers.

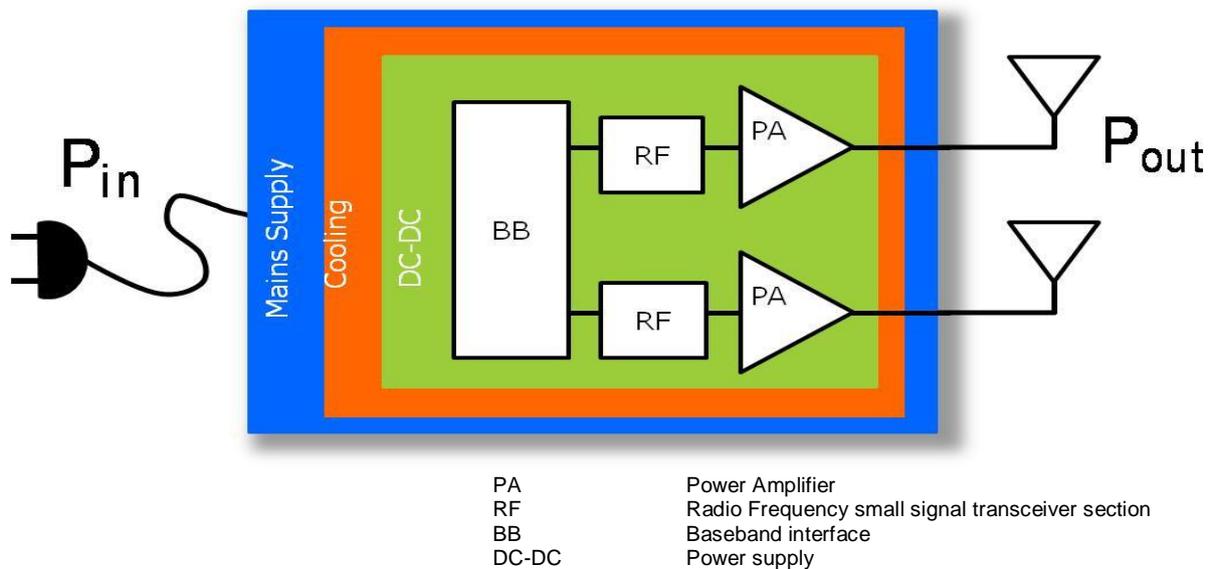


Figure 1.6: System components of a Base station [16]

Typically, a 3G BS makes use of 500W of input power to produce 40W of RF output power [17]. This translates to 50 GWh a year in a network with 12000 BSs which results in a tremendous amount of CO₂ emissions to the atmosphere.

The ever increasing mobile traffic means more BSs need to be deployed and an increase in deployed BSs means more energy needed. The high energy that is required by these many BSs means that there will be high amounts of CO₂ emissions into the atmosphere.

1.1 System Description and Problem Formulation

A wireless cellular network that has B BSs is considered in this research. It is assumed that all BSs are randomly distributed in the network. The power consumed by each BS, P_b is obtained using the power model taken from the Energy Efficiency Evaluation Framework (E³F) [18]. This framework is the most comprehensive solution that has been published by the EARTH [9] project that is used to evaluate and quantify the performance and energy consumption of BSs. According to E³F as given by authors in [19], the power consumed by a BS, P_b is given by:

$$P_b = \begin{cases} N_{TRX} \cdot (P_O + \Delta_P P_{out}), & 0 < P_{out} \leq P_{out(max)} \\ N_{TRX} \cdot P_{Sleep}, & P_{out} = 0 \end{cases} \quad (1.1)$$

Where:

- P_{Sleep} Is the power consumed by a BS while in sleep mode.
- $P_{out(max)}$ Is the maximum transmitter power of a BS at maximum load.
- P_O Is the power consumption calculated at the minimum possible output power, assumed to be 0.1% of $P_{out(max)}$.
- N_{TRX} Number of transceiver chains.

The following parameters were used in the formulation of the problem.

- B Is the maximum number of Base stations in a cellular network.
- U Is the maximum number of users in the network.
- $P_{out(max)}$ Is the maximum transmitter power of a BS.
- $x_{b,u}$ Represents the association relationship between user, u and BS, b .
- λ_b Is the BS sleep coefficient.
- P_b Is the power consumed by BS u .
- $P_{b,u}$ Is transmit power from BS b to user u .
- P_{total} Is the total power used by B BSs.
- P_{Sleep} Is the power consumed by a BS while in sleep mode.

For this research work, it was assumed that when a BS goes into sleep mode, it consumes negligibly small energy. This therefore means that the BS power consumption can be modelled with two values: $\{0, P_b\}$.

This means that the sleep coefficient that can be used to represent the sleep mode of any BS in the cellular network is given as:

$$\lambda_b = \begin{cases} 1, & \text{if } b \text{ is active} \\ 0, & \text{otherwise} \end{cases} \quad (1.2)$$

For a cellular network that has B BSs and U users, the total power, P_{total} , consumed by all BSs is given by:

$$P_{total} = \sum_{b=1}^B P_b \quad (1.3)$$

The optimization problem was to optimize the BS power consumption but maintaining the QoS at an acceptable level so that users of the network continue to receive satisfactory network services. The problem was then formulated as:

$$\text{Minimize } P_{total} = \sum_{b=1}^B P_b \quad (1.4)$$

$$\text{Subject to: } \sum_{b=1}^B x_{b,u} = \begin{cases} 1, & \text{if user } u \text{ is associated to basestation } b \\ 0, & \text{otherwise} \end{cases} \quad \forall_u = 1, \dots, U \quad (1.5)$$

$$\sum_{u=1}^U x_{b,u} \cdot P_{b,u} \leq P_{out(max)} \quad \forall_b = 1, \dots, B \quad (1.6)$$

$$x_{b,u} \leq \lambda_b \quad \forall_u = 1, \dots, U \quad \text{And} \quad \forall_u = 1, \dots, U \quad (1.7)$$

- The objective function (1.4) is to minimize the total power consumption of the BSs in the cellular network.
- The first constraint (1.5) limits each user u , to select only one active BS b .
- The second constraint (1.6) is the capacity constraint that prevents a BS from being overloaded. This ensures that a service process can guarantee a statistical QoS requirement.
- Constraint (1.7) states that users should only be served by active BSs.

1.2 Research Aims and Objectives

Having established that BSs are the major energy consumers in a cellular network, it was decided that a research project be carried out that could result in an increase in the energy efficiency of BSs. The next section outlines the aims and objectives of the research.

1.2.1 Aim of the Research Project

The aim of the research was to investigate methods/schemes that could be deployed to BSs such that as their number increases exponentially, CO₂ emissions associated with this rapid growth are curbed in addition to ensuring that an acceptable level of QoS is maintained.

1.2.2 Purpose of the Research Project

The purpose of the research project was to develop energy saving schemes that could be used in Green Cellular Networks, specifically for BSs in order to reduce CO₂ emissions in a bid to save the environment and to reduce OPEX.

1.2.3 Objectives of the Research Work

- (i) To critically assess the existing and emerging technologies on BS energy saving schemes in Green Cellular Networks.
- (ii) To develop an algorithm that could be used to put BSs to sleep during the time traffic load was very low.
- (iii) To develop an algorithm that could be used to re-associate mobile users in cells where BSs would have been made to go to sleep, to active neighbouring BS.
- (iv) To ensure that the developed algorithm maintains acceptable Quality of Service (QoS) in the cellular network.

1.3 Research Methodology

Two algorithms were developed as described in Chapter 4. Initially, flow diagrams were produced, followed by pseudo codes and finally the pseudo codes were developed into C++ applications which were later imbedded in modules in OMNeT++ [20] for the purposes of simulation. One algorithm was for BS sleeping and the other was for user association. The algorithms were validated through extensive computer simulations. Simulation was used as a substituted form of experimentation and intervention on the actual system since experimenting using actual BSs was going to be very costly and prototyping was going to be time consuming. According to [19], researchers compare different outputs that are obtained by use of different structures and models and then infer what might happen in a real situation.

For this research work, OMNeT++ 4.6 together with INET 2.3.0 [21] and SimuLTE 0.9.1 [22] were used. OMNeT++ is very popular with researchers as it provides a comprehensive framework for modelling wireless networks. With OMNeT++, each stage of the network simulation incorporates model design, simulation, data collection and analysis by use of discrete event driven simulation mechanism [20]. The simulation tool was used to simulate the proposed network with a set of predefined parameters. Results of computer simulations were analysed with regards to the performance of applications like VoIP, web and file transfers on an LTE-Advanced network to represent a real network. The performance parameters that were tested were throughput, packet loss, end-to-end delay and queue length. Total power consumption was also measured to check the extent to which the proposed scheme reduced the total power consumed by a BS.

1.4 Contributions of the Research Work

The research work may lead to practical solutions that save energy in future mobile networks whilst maintaining QoS at desired levels. There has been quite a number of contributions on BS sleeping strategies but very few have concentrated on energy saving that has an emphasis on QoS. Most of these have tended to put more emphasis on the actual energy savings at the expense of QoS. Most network users of today are very critical of the service that they get from their providers. If the service is not up to standard, they move on to the next provider. This means that these network providers can only accept strategies from researchers that do not compromise their customers' satisfaction. Network providers also stand to benefit from the proposed scheme in the form of a reduction on OPEX.

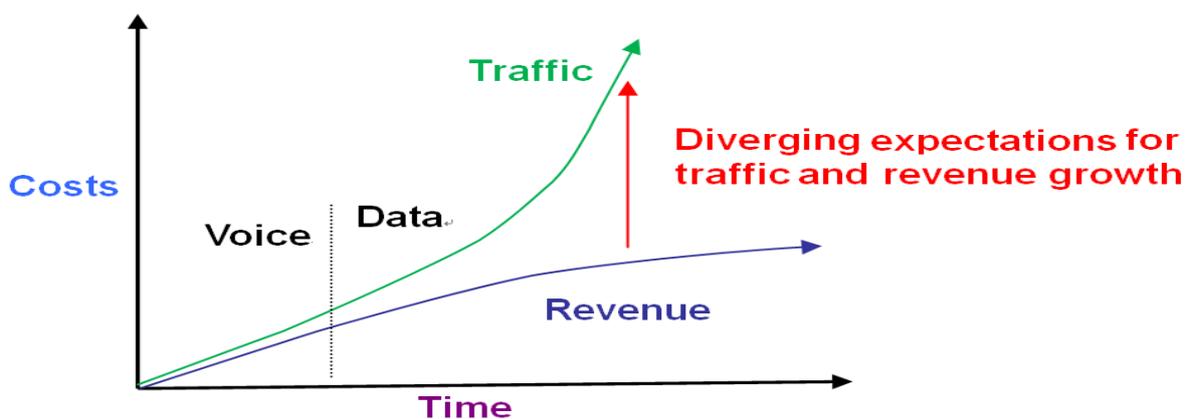


Figure 1.7: Variation of mobile operator's OPEX with time [6]

Figure 1.7 shows how a mobile operator's operating expenses vary with time. The revenue as shown in Figure 1.7 does not increase at the same rate as traffic. This means that with an increase in traffic, OPEX increases and this reduces the expected profits. This goes to show that mobile operators are likely to accept solutions that help reduce the operating costs by reducing energy consumption whilst providing an acceptable level of QoS which was the main goal of this research project. This research work also adds to existing literature on Green Cellular networks which researchers might find useful in future in the managing of energy in future networks. A saving in energy consumption in cellular networks results in the reduction of CO₂ emissions. Most corporate organisations like Vodafone, ECONET, MTN and MTC-Namibia, to mention but a few, are willing to improve their corporate responsibility by adopting strategies that help reduce their carbon footprint. The solution herein in this research, that was analysed and tested, might be what these corporate organisations are looking for in order to provide an acceptable service to the network user.

1.5 Thesis Organisation

Chapter 1, introduced the research topic covering issues relevant to green cellular networks and the rationale as to why the research project was undertaken. Chapter 2, provides an analysis of other researchers' work who researched on BS sleeping schemes and other energy saving schemes. A brief overview of heterogeneous networks is given in Chapter 3. This is necessary as it provides a good grounding for the proposed solution. Chapter 4 outlines the proposed solution. The solution is broken down into sub-solutions that are all analysed in this chapter. Chapter 5, presents the design of the proposed solution outlining all the sub-solutions' design. The implementation, test procedures and test results are presented in Chapter 6. Mechanisms that were used to implement the prototype together with detailed test procedures are given in this section. Chapter 7 presents the analysis of the test results with Chapter 8 concluding the thesis and providing suggestions for further work.

CHAPTER TWO: REVIEW OF BS SLEEPING TECHNOLOGIES

2.1 Introduction

There has been quite a number of proposed technologies that have tried to address the energy consumption issue but most have failed to adequately address the problem especially when it comes to the issue of QoS. Strategies that only seek to address energy consumption reduction, without tackling the issues of QoS, risk being rejected by mobile network operators. The cellular business has become so competitive that operators would not incorporate strategies that would result in a compromised QoS as that would scare away potential customers. Quality of service in cellular networks in [23] is defined as the ability of a cellular network provider to provide a satisfactory service. According to [24], [25] QoS of a cellular network is measured using:

- (i) Throughput- this refers to the rate at which packets move in a network.
- (ii) End-to-end delay- this refers to the time a packet travels from one point to the other.
- (iii) Packet loss rate –refers to the rate at which packets are being lost.
- (iv) Reliability –this refers to the availability of a connection.

QoS in cellular networks is of paramount importance, it ensures that communication cannot be disrupted and whatever information needs to be exchanged is exchanged without errors. Bank transfers containing some errors would result in litigations against the network providers just to mention one example.

2.2 BS Sleeping Technologies

Traditional cellular networks are designed such that they meet the QoS requirements of a network during the busy hour. This means that even when cellular traffic is at its lowest level, BSs still remain switched ON resulting in a huge amount of energy going to waste. Many researchers have, of late, come up with schemes that allow the number of BSs that are switched ON, to be dependent on the prevailing cellular traffic. The next section will present most of these BS sleeping technologies. The benefits and shortcomings of the schemes will also be discussed.

2.2.1 Distance-Aware Base Station Switching OFF Scheme (DSO)

By capitalising on the fact that a BS transmission power depends on the furthest UEs in the cell, Bousia et al. in [26], introduced a scheme that they called, “Distance-aware Base Station Switching OFF Scheme (DSO).” Under the proposed scheme, each BS has to estimate the distances of UEs that it is serving and also the distances of UEs under the control of neighbouring BSs. This information is shared by the BSs using the X2 interface. After acquiring this distance information, BSs go on to calculate the average distances. The BS with the maximum average distance is the one to be switched OFF first. The process is continued until a maximum number of BSs is switched OFF. The scheme is not centralised and as result active BSs have the responsibility of informing their neighbours to switch back ON when traffic levels are picking up. The proposed scheme does not seem to offer solutions to the association problem. If a BS is switched OFF, current users within the cell served by the BS that will be switching OFF need to be rechannelled to neighbouring BSs to guarantee an agreeable QoS. The BSs to which the current users must be redirected, might not be the ones closest to them. The acceptable QoS might come from other BSs serving the least traffic. This scheme seems to have been developed for homogeneous networks. Where the network is a heterogeneous network (HetNet), problems are likely to be encountered with this proposed solution as in a HetNet, BSs have different cell sizes. Ranking the BSs according to the maximum average distance values will always leave the macros on top of the list. The fact that there is no central controller might lead to coverage holes creeping into the system. Coverage holes degrade QoS. There has to be some level of central control that checks whether all areas that are supposed to be covered at any given point in time are covered. Leaving this responsibility to active BSs does not solve the issue. The authors evaluated the performance of their proposed scheme using LTE-Advanced Standard and considering an urban scenario. The authors produced results indicating a 29% energy saving on implementation of their proposed energy saving scheme. This is a huge energy saving but the issue of QoS was not thoroughly investigated. A proper association mechanism is very important to fully utilise the energy saving coming from traffic aware BS operations.

2.2.2 Cell Zooming Technology

Niu et al. in [27], proposed a cell zooming technology where cell sizes are dynamically changed to follow the variation of traffic.

The scheme makes use of a cell zooming server that is in charge of controlling the cell zooming procedure. The server is a virtual entity which can reside in a gateway or in BSs. The server continuously collects traffic statistics and decides when and where cell zooming needs to be applied. Where a cell has to zoom in or out, coordination with neighbouring BSs is supported by communication through the X2 interface. Several techniques can be applied for cell zooming:

- (i) Increasing or decreasing transmit power of BSs.
- (ii) Antenna height or tilt.

Figure 2.1 shows zooming in/out by use of angle of tilt. Several benefits can be derived from this scheme of cell zooming. It can be quite useful in load balancing and improvement of user experience in the form of increased throughput and battery life. The scheme can also result in minimal energy saving, minimal because as some BSs sleep, the remaining BSs have to increase their output power accordingly to try and reduce coverage holes.

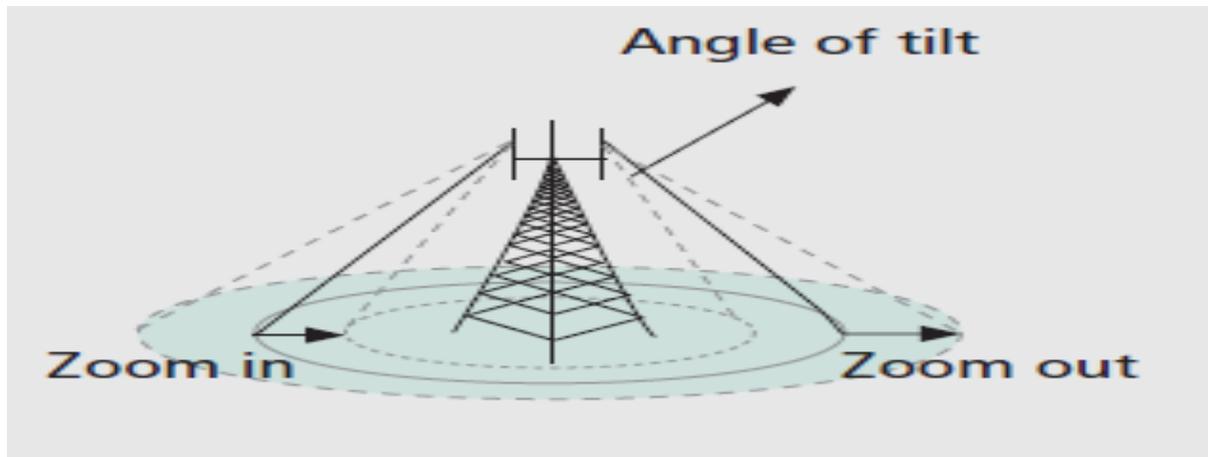


Figure 2.1: Zooming in and out by reducing transmit power and adjusting antenna tilt [27]

By the authors' own admission, the scheme can have challenges when there are significant spatial and temporal load fluctuations. No solutions are offered to inter-cell interference that emerges as cells zoom in and out. Cell zooming results in larger areas of overlapping cells that leads to an appreciable level of inter-cell interference. Again, the scheme is silent when it comes to the issue of HetNets. This means that the scheme could have been designed solely for traditional homogeneous networks and might need some readjustments if it has to be used in

next generation networks which are going to make use of HetNets. The proposed scheme does not fully address the problem of traffic imbalance.

2.2.3 Cell Wilting and Blossoming

A very similar scheme to the cell zooming technique was proposed in [1]. Unlike the cell zooming technique, this scheme was designed to work with HetNets. The authors called the scheme, cell wilting and blossoming. Under this scheme, BS activity is monitored and once low activity is identified, the BS is switched OFF by progressively reducing its transmit power at the same time neighbouring BSs' expand their coverage. If UEs are to experience some level of degradation in the signal quality as a BS is wilting, the wilting BS has to be notified to abandon the wilting process. The authors suggested that the switching ON/OFF process should be very gradual to allow for very smooth handovers. The gradual process is not necessary when the BS that has to switch OFF is a femto or micro cell since most of the time, UEs under these small cells can also hear from the other BSs. This means that the handovers will be quick and effective. This scheme does not however guarantee a continuous coverage or avoidance of coverage holes. Its main problem arises when a UE is switched ON in an area where a BS is in sleep mode. The BS will not be able to detect the UE. The UE has no means to communicate its problem since it's in an area that is not covered. The algorithm that was proposed by these researchers does not have a solution to coverage holes. QoS can be affected by abrupt changes in transmitted power levels. A UE that receives a strong signal from a nearby BS cannot hear other BSs' signals and if this BS were to switch OFF too quickly, this would result in the UE not managing to get a connection. The proposed algorithm cannot handle this sudden disconnection. The authors conceded that there should be a new generation of BS sleeping techniques that can address the coverage holes' issue. They suggested that blind wake up might help solve the issue in the next generation networks.

2.2.4 BS Sleep Control for Heterogeneously Powered BSs

Zhou et al. in [28], proposed a solution that includes BS sleeping technique at the same time using a combination of the conventional BSs where some are connected to the power grid and others get power from the sun and wind. The solar and wind powered BSs are forced into sleep mode if there is not enough harvested energy in their batteries. The BS connected to the grid on the other hand, goes into sleep mode when traffic is very low. The system needs a fairly

sophisticated mechanism that delicately coordinates both sleep modes such that energy outages do not take place.

There can be energy outages in instances where solar and wind powered BSs cannot take over UEs that would have been handed over to them by grid powered BSs that would have gone to sleep. Though the proposed scheme results in moderate energy savings, it requires a high capital expenditure in the beginning in setting up the infrastructure.

2.2.5 Switching ON/OFF Based Energy Saving Algorithm

Son et al. in [29], came up with a SWES (Switching ON/OFF based energy saving) algorithm that uses network impact as a metric to switch ON/OFF BSs. The authors defined network impact as, “how much the switching OFF of a BS will affect the network.” This decision criterion depends on traffic information of a BS and its neighbours. This means that a central controller is not required. Under the scheme, BSs share their traffic loads as well as UEs’ signal strengths. Using this information BSs individually decide whether they can stay switched ON or be switched OFF. Switching OFF a BS will have an impact on neighbouring BSs. Two conditions have to be met for a BS to switch OFF:

- (i) UEs that the BS is currently serving have to be in coverage areas of neighbouring BSs.
- (ii) The total traffic in an accepting BS after UEs are handed over should be less than a prescribed threshold.

Once a BS has decided that it has to switch OFF, it broadcasts a RTSO (Request to Switching OFF) message before going into sleep mode. This is done to prevent overloading neighbouring BSs when two or more BSs simultaneously switch OFF. A BS will switch OFF after receiving a CTSO (Clear to Switching OFF) message from all its neighbours. The scheme results in substantial energy saving but, since the switching ON and OFF does not have a central controller there is a strong possibility of having coverage holes in the cellular network. If it so happens that the RTSO and CTSO messages are exchanged at the same time, the proposed algorithm might fail to operate effectively leading to a compromised QoS. The algorithm was never tested on a HetNet which means its applicability with next generation networks is questionable.

2.2.6 User-Aware Cell Switch OFF Algorithm

Aydin et al. in [30], proposed a heuristic algorithm that they called User-Aware Cell Switch OFF algorithm.

For their algorithm, they did not switch OFF BSs using the least traffic load criterion but they used the total power of UEs

When a BS is switched OFF, UEs are handed over to neighbouring BSs with the highest received power. UEs that are handed over to neighbouring BSs, have to increase transmit powers to cater for the increased distance. BSs are sorted according to the total power required by UEs to connect to neighbouring BSs. The BS that offers the least total power is the one that is switched OFF first.

Additional signalling is required for the capturing of locations of UEs and BSs by a central station. Handover process is initiated once candidate BSs that are to be switched OFF have been identified followed by the switching OFF process.

From the simulations carried out, the authors managed to show that the scheme can perform better, compared to cell zooming techniques especially in terms of the maximum number of BSs that can be switched OFF at any given time. However, the results do not show how QoS is affected by the introduction of the scheme. There is no mention of coverage holes in their scheme though it is obvious that where BSs have to be switched OFF, coverage holes have to be addressed. It was suggested that the scheme can be used for 5G networks, but there is no mention of HetNets. Future generation networks will be HetNets which means that solutions that do not cater for such networks will not be very useful. Several other BS sleeping schemes are reviewed in the author's previous work [31], [32].

2.3 Conclusion

This chapter has presented a number of BS sleeping technologies that can lead to substantial energy savings. Most network operators have always designed their networks to cater for maximum traffic during the busy hour. This means that during the time traffic is low, a lot of energy is wasted. The schemes that have been presented in this Chapter will result in high energy efficient networks since the total network energy consumed follows the total network

traffic variation. Most of them have however, failed to address the issue of coverage holes. Where a BS has been switched OFF, coverage holes follow. The schemes should then suggest ways of how these holes have to be handled. Again, most of the schemes have also failed to convincingly address the issue of HetNets which will be predominant in the next generation networks.

The next chapter will present a brief overview of HetNets since the proposed solution should be deployable in HetNets. In almost all the schemes reviewed here, the main objective was to reduce the total power consumed by the network elements. This was mostly achieved, but it was achieved at the expense of QoS. QoS of a cellular network is very critical and should never be compromised when such energy saving schemes are deployed.

CHAPTER THREE: AN OVERVIEW OF HETEROGENEOUS NETWORKS

3.1 Introduction

Presently, most traditional cellular network operators are making use of homogeneous networks. In a homogeneous network, all BSs have similar transmitter power levels. BSs use the same antenna patterns. Receiver noise floors and backhaul connectivity are all similar. For such a network, to cater for an increase in traffic load, it has been the tradition by network operators to reduce the cell sizes and setting up BSs in the new areas formed. This is a complex and expensive procedure. It is also time consuming. Traditional networks use macro BSs throughout their entire networks. Macro BSs consume very high power and they also radiate high power. As a result of the high power transmissions, large areas are covered. They serve a large number of users within their jurisdiction. A heterogeneous network (HetNet) on the other hand is a network that has different types of BSs that operate together such that the desired network parameters like coverage, capacity and energy efficiency can easily be met. They are an inexpensive alternative to increase network capacity. Vandris et al. in [33], suggest that to meet future network capacity, the following approaches can be taken:

- (i) Improving the macro layer.
- (ii) Densifying the macro layer.
- (iii) Adding micro cells.

3.2 Heterogeneous Networks

HetNets enable the deployment of a cocktail of BSs such as macros, picos, micros, femtos and relay BSs. An example of a HetNet is given in Figure 3.1.



Figure 3.1: A HetNet architecture [33]

HetNets have been proposed as a technology of the future by prominent groups such as 3GPP. 3GPP is a consortium of standardisation bodies in mobile communications. Some of 3GPP's highlights [34] which include small cells are shown in Figure 3.2.

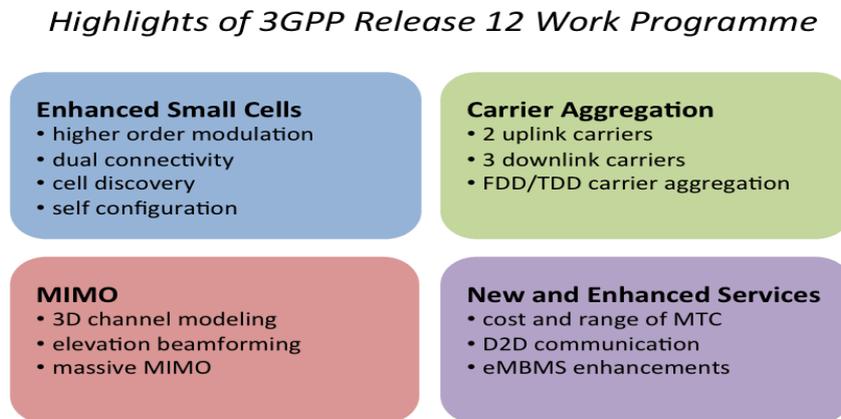


Figure 3.2 3GPP Release 12 work programme [34]

In next generation networks, as the demand for services increases, small cells should be added to certain parts of the network to form HetNets. Benefits of small cells according to [33] are:

- (i) Close proximity of a mobile device to a BS leads to high S/N which in turn leads to the usage of higher order modulation schemes like 256 QAM for the downlink. The use of high modulation schemes substantially increases data throughput.
- (ii) Easy to install.
- (iii) Low power, high performance and low cost deployment.
- (iv) Dual connectivity, mobile devices can be connected to a macro cell and a micro cell at the same time simplifying handovers in the process.
- (v) Efficient schemes are being designed for mobile devices to discover the small cells around them without draining their batteries excessively.
- (vi) Performance improves.

There is an increasing demand for high QoS and this makes the deployment of micro BSs a priority. Although micro BSs cover reasonably small geographical areas, their S/N is much higher compared to that of macro BSs due to the shorter propagation delays leading to lower path losses. HetNets improve the broadband user experience in a ubiquitous and cost effective manner.

Generally, it is believed, as Vandris et al. in [33], state that deploying many micro BSs which are very small and have very low power, results in considerable energy savings compared to the deployment of fewer macros that consume higher power. Table 2.1 shows the different BSs and their power consumption.

Table 2.1: LTE BS transceiver power consumption, 2012 State-of-the-art estimation for different BS types [36].

	Parameters	Macro	Micro	Pico	Femto
PA	Max. Tx rms power [W]	39	6.3	0.13	0.1
TRX	Max. Tx rms power [W]	5.7	5.7	2.9	0.2
BB	Radio [inner rx-tx] [W]	5.4	4.6	0.6	0.5
DC-DC	σ_{DC} %	6.0	6.4	8.4	8.0
Cooling	σ_{COOL} %	9.0	0.0	0.0	0.0
Mains supply	σ_{MS} %	7.0	7.2	10	10
Total power for 1 Radio	[W]	160.8	47.0	4.5	3.1
#Sectors	#	3.0	1.0	1.0	1.0
# PAs/Antennas	#	2.0	2.0	2.0	2.0
#Carriers	#	1.0	1.0	1.0	1.0
Total power for n radios	[W]	964.9	94.0	9.0	6.2
Typical cell radius		>1km	250m~1km	100m~300m	10~50m

3.2.1 Femto Cells

A femto cell is the smallest cell. The capacity in a building or stadium can best be served by femto cells as they have very small footprints. Compared to pico cells, their radii are about 10-30% and this leads to an increase in indoor capacity by a factor of between 10 to 100 [35]. Femto-eNBs have lower power than pico-eNBs and their processing power is much lower. These are some of the factors that make them more suitable for high-capacity indoor applications compared with the other BSs. Most indoor mobile users are either stationary or moving, moving slowly and this means that the femto-eNB's low processing speed will suffice. Femto cells have full plug-and-play features and they also have sniffing and auto-configuration capabilities making it easier, cheaper and faster for service provider to deploy them. Figure 3.3 shows femtocells BS deployment to cover a stadium.

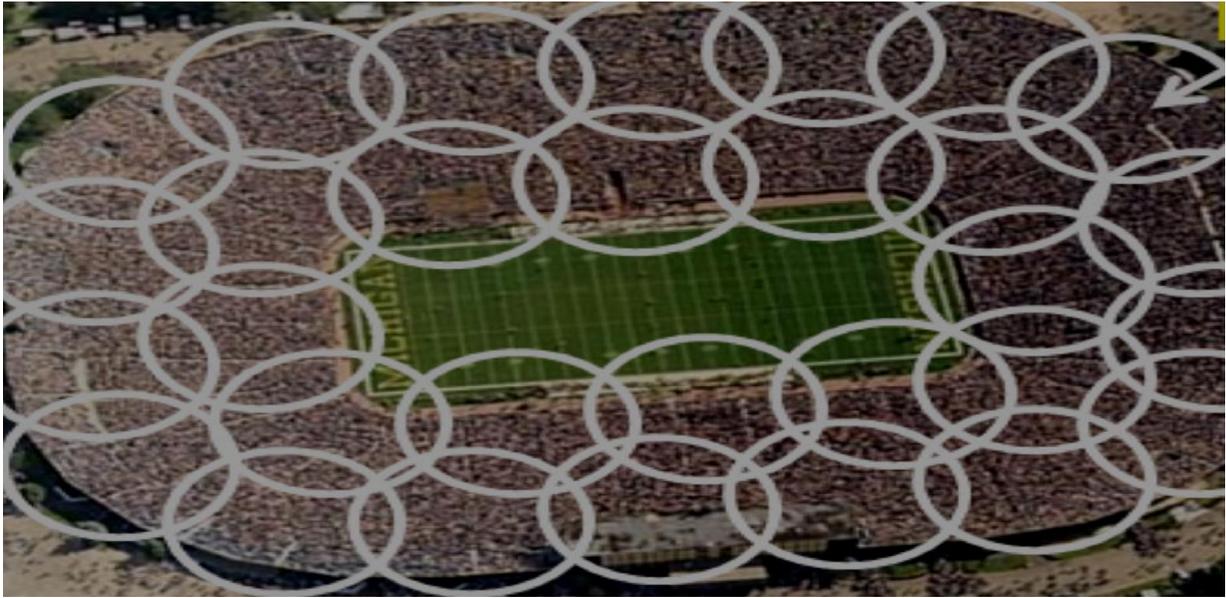


Figure 3.3: Coverage provided by femtocells in a stadium [35]

3.2.2 Pico Cells

Pico cells are slightly bigger than femto cells as shown in Table 2.1. They have higher PA power which makes them a natural choice where large coverage footprints are needed. Their processing speed is high which makes them better placed to serve outdoor hotspots that have mobile users who might be on the move. Their high processing speed can easily handle the fast hand-offs that come with mobile users that are on the move.

3.2.3 Micro Cells

Micro cells are special BSs that are used to connect the operators' core network and they usually provide additional coverage and capacity. Their coverage areas are slightly bigger than those of pico cells. Micro cells are simple to deploy, they can be wall mounted, mounted on a light pole or can be mounted as a simple standalone BS. They are usually mounted to serve dense clusters.

3.2.4 Macro Cells

A macro cell provides the largest radio coverage in a cellular network. Macros have high power outputs of typically tens of watts. They are the BSs of choice if the area to be covered is a highway or a rural area that is sparsely populated. They are usually mounted at a height that provides a clear view over the surrounding buildings and terrain. Macros are mainly used to provide basic coverage.

For this research work, all the different types of small cells were collectively considered as micro cells. Micro cells are estimated to be close to 100 million by year 2020 [36]. If these micro cells are always on, overprovisioned to be able to meet a specified QoS during the busy hour, 5TWh of energy will be consumed in 2020 alone. This is quite a huge amount of energy which is equivalent to an annual electricity consumption of 1.4 million households in South Africa [37].

3.3 Conclusion

This chapter has presented an overview of HetNets. The benefits of using HetNets have been given. The section has also outlined the different types of small cells that exist and the best scenarios where these different types can be deployed. The next chapter will outline the proposed solution. All the components of the solution will be presented and analysed.

CHAPTER FOUR: ANALYSIS OF THE PROPOSED SOLUTION

4.1 Introduction

This chapter outlines the proposed solution. Firstly, the problem statement is presented. A solution to the problem was identified and decomposed into several components. These components were then analysed leading to the choice of performance metrics that were used in the project. The solution that was proposed was one that was specifically meant to work with heterogeneous networks. Heterogeneous networks are networks that are intricately designed to take care of the ever increasing cellular traffic demand. HetNets were introduced in LTE Advanced and are taken as an integral network feature in 5G networks. This means that the proposed solution is very well positioned to work with next generation networks.

4.2 The Problem Statement

Most of the existing cellular networks are designed such that BSs are always kept switched ON to provide coverage anytime even in cells where there might not be a single active user [17]. This is not energy efficient. BSs, which are the most energy hungry network equipment, should be put to sleep during periods of low traffic to save energy. Most BS sleeping schemes that have been proposed do not adequately address the issue of QoS. With the exponential growth of mobile systems, a solution that results in energy efficient BSs would go a long way in alleviating the problem of greenhouse gas emissions, specifically CO₂ emissions. This would make the world a better place to live in.

4.3 The Proposed Solution

The proposed solution is a totally new algorithm developed to specifically cater for heterogeneous networks. It should also be stated here that the proposed solution is an improvement of the solution proposed by the author of this thesis in [32]. Most of the BS sleeping algorithms [26], [29], [38], [39], [40], [41] that have been advanced by researchers focus on homogeneous networks. The current trend is that most cellular network providers are shifting from homogeneous networks to heterogeneous networks. The few solutions for heterogeneous networks that have been suggested do not adequately address coverage holes and to some extent do not result in substantial energy savings. They do not put an emphasis on QoS whilst trying to address the energy consumption problem.

The proposed solution could be used to put to sleep micro BSs that are lightly loaded in HetNets whilst maintaining an acceptable QoS level.

4.4 Breakdown of the Proposed Solution

The proposed solution was first broken down into two major components, namely:

- (i) An algorithm that is used to switch OFF lightly loaded micro BSs in heterogeneous networks.
- (ii) An algorithm that is used to re-associate UEs to still active neighbouring BSs.

These two major components were then further broken down into subtasks. This top-down design strategy was quite effective in designing the two algorithms. The task of designing the algorithm that was used in the switching ON/OFF of BSs in heterogeneous networks was decomposed into the following subtasks:

- (i) Designing a sub algorithm that is used to collect the traffic load from all BSs within a specified area.
- (ii) Designing a sub algorithm that is used to arrange BSs in ascending order according to their traffic load.
- (iii) Designing a sub algorithm that is used to switch OFF BSs whose traffic loads are below a specified threshold.
- (iv) Designing a sub algorithm that is used to check for any deterioration of QoS.

The other major component of the solution; the BS-UE association algorithm, was decomposed into the following subtasks:

- (i) Designing a sub algorithm that is used to collect Specific Energy (SE) values from neighbouring BSs. Specific energy is the amount of energy a network consumes to the total number of bits correctly delivered in the network within an observation period.
- (ii) Designing a sub algorithm that is used to arrange the SEs in descending order.
- (iii) Designing a sub algorithm that that is used to hand over UEs according to the BSs traffic load/ SEs pairings.

This adopted divide and conquer design strategy ensured that the complex task of switching OFF some BSs and re-assigning UEs to neighbouring BSs was broken down into a series of smaller tasks that were more manageable.

This made the resultant program easier to understand and modify. Testing and debugging of the program also became easier. The components could be tested and debugged separately. The next section discusses the first major component of the solution; the switching ON/OFF of BSs in heterogeneous networks and the subsequent sub tasks associated with this main task.

4.5 The Switching ON/OFF Algorithm

Switching OFF some BSs manually when there is low traffic in a bid to save energy is not a viable and recommended option. A practical approach would be to make use of a centralised controller that continuously monitors network information within a specific reasonably sized area within the coverage area. The controller collects network information on traffic load, user requirements as well as channel information. If it so happens that the central controller goes down, the network should simply revert back to a non-energy saving mode where the BSs are always ON regardless of the number of users. A traffic load threshold should be set such that if a micro BS reaches this value, it is switched OFF with the few users being re-assigned to neighbouring BSs. There is no need for BSs to completely shut down as was proposed by Son et al. in [29]. If BSs are to completely shut down, coverage holes will emerge in the network. If it happens that a UE is switched ON in an area where a BS is OFF, network problems will be encountered. To avoid coverage holes, a BS in sleep mode is switched ON using control signals coming through a dedicated backhaul link to participate in a handover process of the UE to a neighbouring active BS. With the proposed scheme, BSs should have a limit of traffic load that they are allowed to handle.

If there is a micro BS in sleep mode in the near vicinity, it is awoken up and the excess traffic is handed over to an accommodating BS or other neighbours handling a reasonable amount of traffic load. In addition, if the number of UEs within a cell where the BS is sleeping reaches a certain prescribed value, that sleeping micro BS should be awoken. Micro BSs should be awoken through a backhaul interface connecting BSs to the core network. In LTE A, B and C, this interface is called the X2 interface. The decision to switch ON/OFF the micro BSs comes

from a central controller through the effective and efficient operation of the energy saving scheme. The remaining active BSs need not increase their power to cover cells where BSs would have gone into sleep mode. This is not necessary as in the proposed solution; the micro BSs are not completely switched OFF.

Since the micro BSs are partially switched OFF, it means if a UE arrives in the cell, the receiver section of the micro BS would pick up the signal and in conjunction with the core network, the micro BS would be awoken up to carry the arriving traffic. It wakes up to participate in the process of handing over that UE to an active neighbouring BS. This is how micro BSs on cell edges should handle the BS sleeping scheme. For the ones inside areas covered by macros, once they go sleeping, their traffic is handed over to macros. They do not leave behind coverage holes. This is one aspect in which the proposed solution differs with the majority of the schemes that have been advanced to solve the problem. With schemes where BSs have to zoom in power to cover neighbouring BSs where BSs might be in sleep mode [1], [27], [29], [42], coverage holes are always a problem. This is because BSs have a maximum transmission power and if it so happens that there are UEs outside a BS that is transmitting at maximum power, then those UEs are not covered. Raising the power of the BSs that remain active reduces the intended energy savings. The zooming in by BSs also increase network interference as overlapping areas increase. The maximum and minimum thresholds for BSs should not be made the same for the entire network. The proposed solution was designed to be able to handle heterogeneous networks. BSs meant to serve areas like shopping malls should be those that have very high capacities and cannot have the same maximum values as BSs that serve residential areas with low traffic. The thresholds used in the algorithm were set from historical daily traffic variation analysis. The proposed solution is very similar to the one proposed in the author of this thesis's previous work. [32]. The main difference is that the one proposed in this thesis is meant to deal with HetNets.

The proposed micro BS switch ON/OFF algorithm made use of the following steps:

- Step 1: Collection of traffic load from all BSs.
- Step 2: Arranging micro BSs in ascending order according to the collected traffic load.
- Step 3: If a micro BS's traffic load is below an agreed minimum threshold, the micro BS is switched OFF provided there are some neighbouring BSs that can accommodate

the few UEs the micro BS was handling. If the neighbours cannot accept any additional load, then switch ON/OFF algorithm is stopped.

If there are accommodating neighbours, the controller needs to broadcast intention to switch OFF the micro BS.

This process then triggers the BS-user association algorithm. If the neighbours are not in a position to accommodate extra traffic, then the algorithm is halted.

Step 4: The QoS of BSs is then checked. If the switching OFF procedure compromises QoS, then the algorithm is stopped.

4.6 The BS-UE Association Algorithm

Users in cells in which BSs are to be put to sleep have to be transferred to neighbouring BSs that can best serve them. This is done by considering the traffic load the users might be currently processing and also the traffic loads the neighbouring BSs will be processing. Re-association of users is carried out after analysing the performance of network elements. The most commonly used quality performance metrics that are used to analyse energy consumption in a network according to E³F in [9] are power per unit area and energy per bit. The next section analyses these two performance metrics leading to the justification of the usage of energy per bit as the metric that was used in this research. The choice of a performance metric is very important in Green Cellular Networks as it gives an indication of energy savings that stand to be achieved.

4.6.1 Power Per Unit Area

Power per unit area is the ratio of average power (p) that is used in a network to network coverage area (A), expressed mathematically as:

$$ECI_{P/A} = \frac{P}{A} [\text{W/m}^2] \quad (4.1)$$

Where:

ECI represents the Energy Consumption Index.

The ECI gives an overview of the total power consumed by the network. This performance metric may also be used to give an indication of CO₂ emissions. However, ECI is not so sensitive to traffic load because it increases very slowly with total system throughput. It could therefore not be used in this project because of its insensitivity to throughput.

4.6.2 Specific Energy, SE

This metric is very sensitive to the traffic load a BS might be handling. This sensitivity to system throughput made it the natural choice as a performance metric for use in this project for the BS-UE association algorithm.

Specific Energy is the amount of energy the network consumes to the total number of bits that are correctly delivered in a specific network within a given time, T.

Mathematically:

$$SE = \frac{E}{B} = \frac{P}{R} \quad (4.2)$$

Where:

B gives the total number of bits.

P is the average network power.

R gives the data rate.

E is the total amount of energy that is consumed by a network.

SE is the specific energy.

SE is an indicator of network delivery efficiency. It gives the energy that is required to move one bit of data from one point to the other in a network. It is called Energy Consumption Ratio (ECR) by Dini et al. in [34]. Equation 4.2 shows that the traffic load in a given cell is inversely proportional to specific energy.

At maximum traffic load, the power that is consumed [36] by a BS is given by:

$$P_{in} = N_{TRX} \frac{\frac{P_{out}}{\eta_{PA}(1-\sigma_{feed})} + P_{RF} + P_{BB}}{(1-\sigma_{DC})(1-\sigma_{MS})(1-\sigma_{cool})} \quad (4.3)$$

Where:

N_{TRX} Gives the number of transceiver chains.

σ Is the loss factor and $\sigma = 1 - \eta$ where η is the efficiency given by P_{out}/P_{in} .

σ_{feed} Represents feeder losses.

σ_{cool} Represents losses due to cooling system.

η_{PA} Represents efficiency of the power amplifier.

σ_{MS} Losses in the mains supply.

σ_{DC} Losses in the DC section of the power supply.

According to [1] the overall data rate for one user is given by:

$$R_n = \sum_{k=1}^K r_n^k \quad (4.4)$$

Where k is the total number of sub channels. In LTE, Sub channels can be taken as resource blocks assigned to one user.

For a total of n users:

$$R_T = \sum_{n=1}^N R_n \quad (4.5)$$

The total data rate of all users, i , being handled by one BS taking into account interference plus noise [13] is given by:

$$R_{T,i} = \eta_i \sum_{n=1}^N N_{RB,n} \text{Log}_2 \left(1 + \frac{P_{tx} - ALF_n}{IN_n} \right) \quad (4.6)$$

Where:

$$\eta_i = \eta_{BW} \cdot \eta_{BW_{RB}}$$

[η_{BW} Is system bandwidth and $\eta_{BW_{RB}}$ is resource block Bandwidth efficiency]

IN_n Is the interference plus noise.

$N_{RB,n}$ Is the number of resource blocks allocated to user n .

P_{tx} Is the transmitted power.

ALF_n Represents the average signal attenuation factor due to path loss of n th user.

The average signal attenuation factor is normally obtained by choosing a suitable path loss model.

In full, SE for one base station serving n users is computed as, combining equations (4.3) and (4.6):

$$SE = \left(N_{TRX} \frac{\frac{P_{out}}{\eta_{PA}(1-\sigma_{feed})} + P_{RF} + P_{BB}}{(1-\sigma_{DC})(1-\sigma_{MS})(1-\sigma_{cool})}} \right) / \left(\eta_i \sum_{n=1}^N N_{RB,n} \text{Log}_2 \left(1 + \frac{P_{tx} - ALF_n}{IN_n} \right) \right) \quad (4.7)$$

It should be noted that SE roughly indicates how busy a particular BS is. A high SE means the traffic load is low. High traffic load is characterised by a low SE. Where users have to be re-associated with accommodating BSs, BSs with high SE are the ones that are chosen.

This makes it possible to transfer users to neighbouring BSs in an energy efficient manner, and the method is also good for load balancing. Overloading a specific BS reduces QoS that is experienced by users connected to that particular BS. UEs that need to be transferred to accommodating BSs are themselves arranged in descending order according to the traffic load they are handling. The one with the highest load is the one that is transferred to a BS with the highest SE. This is done so that UEs with high traffic loads are handed over to neighbouring BSs with the lightest traffic load. This makes it possible to meet the QoS desired in a network. The BS-user association is a step by step process, with the UE with the highest load being associated with BS with the lightest traffic load using the SE performance metric, followed by the user with the second heaviest traffic load being transferred to the BS with the second heaviest traffic load. The process continues until all UEs served by ‘about-to-sleep micro BS’ are all re-associated with neighbouring BSs. This is a mechanism that ensures that QoS of a network is not degraded. The BS switching ON/OFF algorithm has another mechanism to ensure a high QoS is retained during the time lowly loaded BSs are being switched OFF. The process of re-association starts when the controller broadcasts intention to switch OFF a micro BS.

In short, the steps that are taken are as follows:

- Step 1: BSs that are neighbours to the micro BS that has to be switched OFF calculate their SEs and send the SE information to the controller.
- Step 2: The SEs are then arranged in descending order.
- Step 3: The users that need to be handed over to neighbouring BSs are sorted in descending order according to the number of resource blocks that they would have requested from the controller, which means the micro BS with the lightest load will be on top of the list paired with the UE that would have requested the highest number of resource blocks.

4.7 Conclusion

In this chapter, a BS sleeping technique has been presented and analysed; a technique that can be used to switch ON/OFF micro BSs within a cellular network so that the total energy consumed by network is able to follow traffic variation with time.

For it to work holistically with HetNets, it was decided to propose a solution that could be designed from scratch rather than to improve existing ones that were designed to work with homogeneous networks. The BS sleeping technique was broken down into two major components, namely:

- (i) The BS ON/OFF algorithm: BSs are not randomly switched OFF. The steps that need to be followed in putting the BSs to sleep have been outlined.
- (ii) BS - user association algorithm: the method that is used to re-associate UEs within a cell where a BS is switching OFF to neighbouring BSs that are still active has been outlined in this chapter.

The metrics, particularly the specific energy, that are used in the re-association process have also been discussed together with the steps involved in the re-association process. The next chapter discusses the design of the solution. The design of all the components making the solution are also outlined.

CHAPTER FIVE: DESIGN OF THE PROPOSED SOLUTION

5.1 Introduction

This chapter outlines the design of the proposed solution. The proposed solution was broken down into three major modules and the three modules were further broken down into sub modules for easier handling. First part of the chapter describes the first module, which is the BS sleeping algorithm together with the subtasks. Second part details the BS-user association algorithm with the two subtasks included. The third part presents how relay nodes are handled.

UEs within cells where BSs have to be switched OFF should be handed over to neighbouring BSs. The first two algorithms were developed to the level of flow diagrams. Pseudo codes were then developed from the flow diagrams. A top-down approach commonly known as a divide and conquer approach [43] was adopted for this project. The whole cellular network was divided into 7-cell clusters. The seven cells are macros. Inside the macros would be small cells to cater for the hotspots and in some areas, relay nodes. The system would be very complex if the BS switching ON/OFF was done centrally for all BSs within a mobile network. The complexity and slowness of the system could render the system dysfunctional and chaotic.

5.2 The BS Sleeping Algorithm

In this research, a controller is responsible for a 7-cell cluster as shown in Figure 5.1.

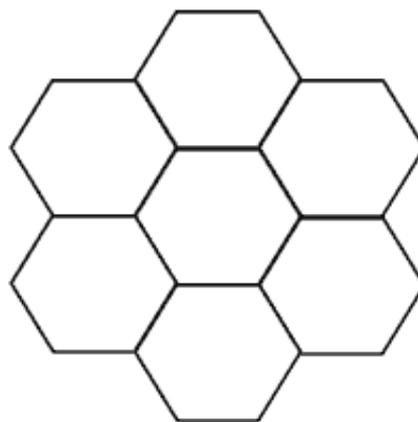


Figure 5.1: A 7-cell cluster

The seven cells represent macro cells. Femto cells, micro-cells and pico-cells are embedded inside the macro cells depending on the location of the hotspots. In some sections of the cells, depending on topology, relay nodes are installed.

Considering the fact that most future networks will be heterogeneous networks, the proposed solution was designed to work with these types of networks. For one cluster, the following steps are taken:

- Step 1: The traffic load for all BSs within a cluster is obtained.
- Step 2: The micro BSs are then arranged in ascending order according to their traffic loads.
- Step 3: If a micro BS's traffic load is below a specified minimum threshold, α_{th} , it is switched OFF provided there are some neighbouring BSs that can accept the few mobiles the BS would be handling. If there is no neighbour that can accept any additional load, then switch OFF algorithm is stopped. If there are accommodating neighbours, then the controller needs to broadcast the intention to switch OFF the micro BS. This process triggers the BS-UE association algorithm. If all micro BSs' traffic load is above a prescribed minimum threshold, then the algorithm is halted to be restarted after thirty minutes.
- Step 4: The QoS of BSs is then checked. If the switching OFF procedure results in degradation of QoS, then the algorithm is halted.

The above steps that have been mentioned are illustrated in the flow diagram in Figure 5.2.

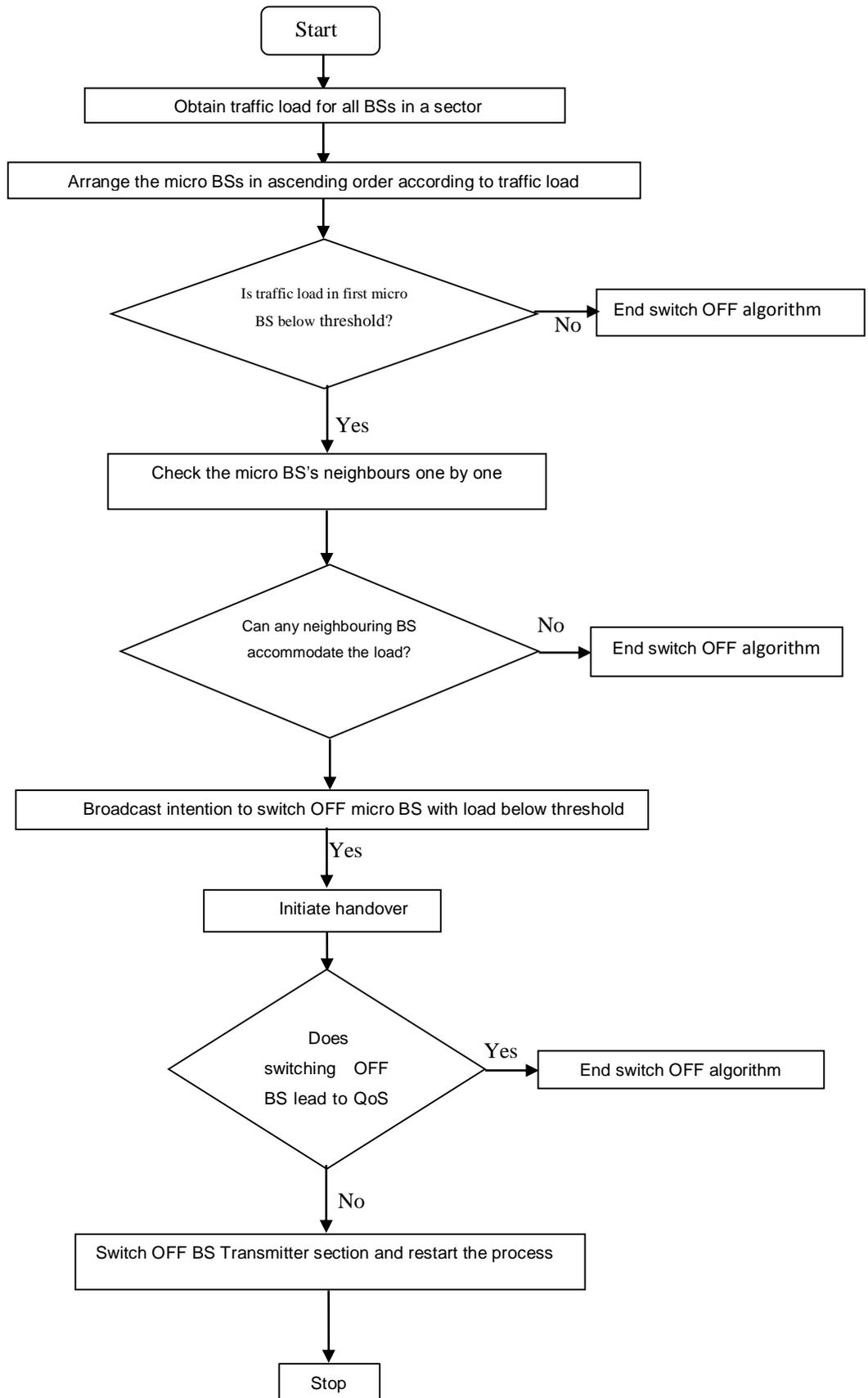


Figure 5.2: The flow diagram of the BS sleeping algorithm

Since flow charts as design tools do not normally reflect some aspects of algorithm development, a pseudo code, which is a newer design tool was developed and presented. The following notations were used in the pseudo code for BS sleeping, pseudo codes for BE-UE association and for switching OFF relay nodes.

5.2.1 List of Notations

Symbol	Meaning
α_{th}	Normalised traffic threshold.
α_{max}	Maximum normalised traffic load to achieve a certain demanded QoS.
α_{load}	Traffic load.
β_{on}	Active BSs.
β_{sleep}	BSs in sleep mode.
β	A set of BSs in a sector.
UE	User equipment or mobile station.
β_{macro}	Macro BSs.
α_{th-RN}	Relay node threshold.
$\alpha_{DeNB-max}$	DeNB maximum normalised traffic load to achieve a specified QoS.
RN_{on}	Active relay nodes.
RN_{sleep} ,	Relay nodes in sleep mode.

5.2.2 The BS Sleeping Pseudo Code

- 1: Inputs: α_{th} , α_{max} , β , β_{macro}
- 2: Outputs: β_{on} , β_{sleep}
- 3: Initialize $\beta_{on} \in \beta$
- 4: **while** $\beta_{on} \neq 0$ **and** $\beta_{on} \not\subset \beta_{macro}$ **do**
- 5: Obtain normalised traffic load for all micro BSs in a sector
- 6: **sort** micro BSs
 - 6.1: Find the micro BS with the smallest traffic load and **swap** it with the first traffic load in the array
 - 6.2: Find the second smallest and **swap** it with the second position
 - 6.3: Find the third smallest and **swap** it with the third position

```

6.4:  repeat finding the next smallest traffic load and swapping it into the
      correct position until the whole array is sorted.
      /* several methods can be used to sort the BSs but the selection sort
      method used here gives the shortest time for ± 50 BSs. */
7:    for each micro BS, starting with the first on the list, do
8:      while  $\alpha_{load} < \alpha_{th}$  do
9:        if neighbouring BSs  $\alpha_{load} < \alpha_{max}$  then
10:       Broadcast intention to switch OFF all BSs with traffic load
        below threshold
11:      else terminate process
12:      end if
13:    end while
14:  end for
15:  Obtain responses to handover requests
16:  if switching OFF a BS does not lead into QoS degradation /* this is the
      case if a specific BS has accepted a handover process*/ then
17:    switch OFF BS
18:  else terminate process
19:  end if
20:  end while

```

5.3 The BS-UE Association Algorithm

The moment the controller broadcasts the intention to switch OFF a particular micro BS, the neighbouring BSs to the micro BS that has to be switched OFF, react by calculating their SEs and sending these to the controller. They calculate their SEs using the following formula:

$$SE = (N_{TRX} \frac{\frac{P_{out}}{\eta_{PA}(1-\sigma_{feed})} + P_{RF} + P_{BB}}{(1-\sigma_{DC})(1-\sigma_{MS})(1-\sigma_{cool})}) / (\eta_i \sum_{n=1}^N N_{RB,n} \text{Log}_2 \left(1 + \frac{P_{tx} - ALF_n}{IN_n} \right)) \quad (4.7)$$

Where:

$$\eta_i = \eta_{BW} \cdot \eta_{BWRB}$$

η_{BW} Is system bandwidth.

η_{BWRB} Is resource block Bandwidth efficiency.

IN_n Represents interference plus noise.

- $N_{RB,n}$ Is the number of resource blocks allocated to the nth user.
- P_{tx} Represents the transmitted power.
- ALF_n Represents the aggregate signal attenuation factor due to path loss of the nth user.

The process of BS-UE association takes the following steps:

- Step 1: BSs that are neighbours to the micro BS that has to be made to sleep calculate their SEs and send this information to the controller.
- Step 2: The controller arranges the SEs from the neighbouring BSs in descending order
- Step 3: The UEs have to be transferred neighbouring BSs are arranged in descending order according to the load they are handling, which means the UE with highest load will be on top of the list. The UE on the top of the list is paired with the BS on the top.

If a BS can serve a UE according to the pairings, then the handover process is accepted otherwise the micro BS is prevented from switching OFF. It might happen that a user with high traffic load, if handed over to a BS, causes the taking over BS to be driven into an overload condition; this has to be avoided at all costs as it compromises the QoS. The BS-user association is summarised in the flow diagram given in Figure 5.3.

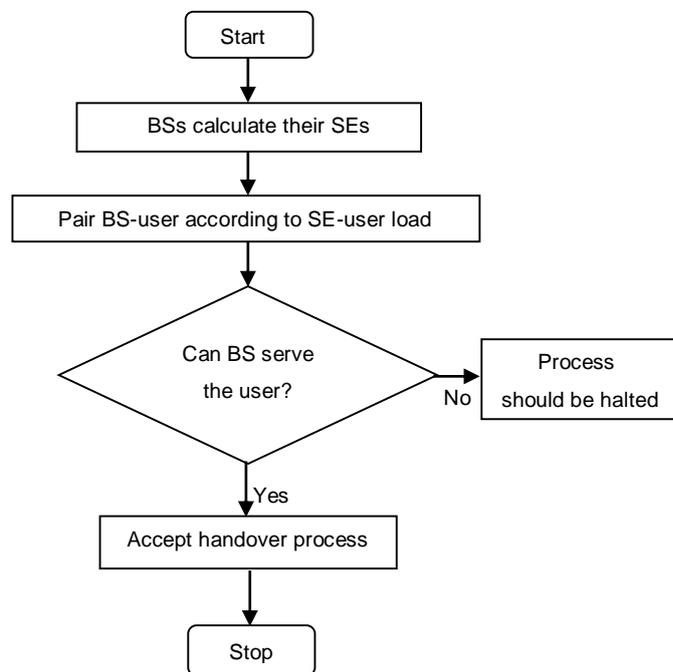


Figure 5.3: The BS-UE association flow diagram

5.3.1 The BS-UE Association Pseudo Code

```

1:   Inputs: micro BSs' SEs,  $\alpha_{UE}$ ,  $\alpha_{max}$ , Outputs:  $BS_n - UE_n$ 
2:   BSs calculate their SEs using equation (4.7)
3:   Obtain SEs from BSs
4:   Pair BS-UE according to SE-UE traffic load
5:   for all BS-UE pairings do
6:       if BS can serve UE then           // BS can accept handover if its  $\alpha_{load} < \alpha_{max}$ 
7:           Accept handover                 // handover is accepted if QoS is not
                                           // compromised upon accepting additional
                                           // traffic
8:       else terminate process
9:       end if
10:  end for

```

5.4 Relay Nodes

In LTE-Advanced, according to 3GPP specifications in the latest release, Release 13 of 2015, HetNets utilise relay nodes (RNs) [45] to provide enhanced coverage to boost capacity at all edges including hotspots. RNs are connected via the radio interface to the donor eNB (DeNB) as shown in Figure 5.4. Relays make it possible to cover specific areas at very low cost. Heavily shadowed areas in a cell can also be catered for by using relays.

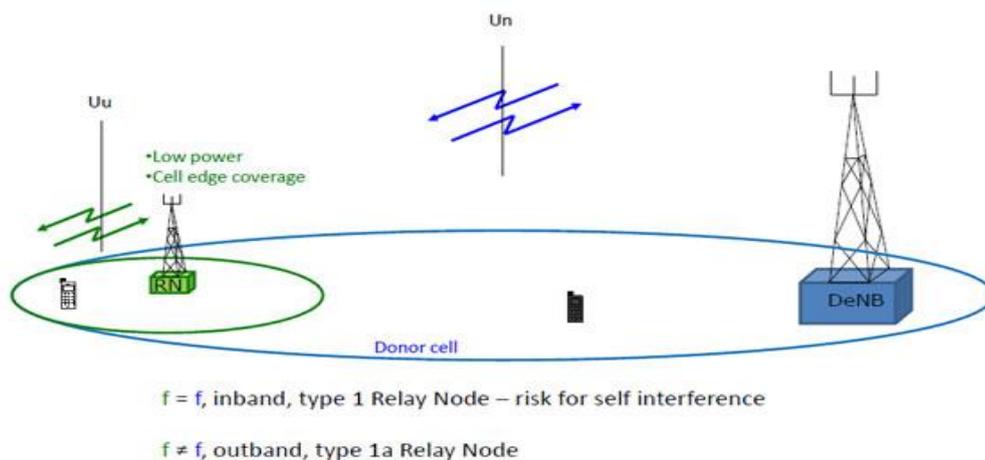


Figure 5.4: A donor cell with a Relay Node inside [45]

The hotspot areas might be shopping malls or stadia that normally have very little traffic during certain times of the day. Future networks utilising relay nodes should also be catered for by the proposed BS sleeping algorithm.

At times around the hotspots, the number of users might go below a certain level such that the few remaining UEs after the RN is switched OFF, have to be handed to over to the DeNB.

Soft handovers take place if the Un and Uu Interfaces (Un and Uu being the air interfaces shown in Figure 5.4) are using the same frequency. If the frequencies are different, then hard handovers take place. The following pseudo code to be used to switch OFF these relays nodes was thus developed.

5.4.1 Relay Nodes Switching OFF Pseudo Code

```

1:   Inputs:  $\alpha_{th-RN}$  ,  $\alpha_{DeNB-max}$ ,  $\alpha_{load-RN}$ , Outputs:  $RN_{sleep}$ ,  $RN_{on}$ 
2:   Initialize  $RN_{on} \in RN$ , DeNB  $\in \beta$ 
3:   while  $RN_{on} \neq 0$  do
4:       if RN  $\alpha_{load-RN} < \alpha_{th-RN}$  and  $\alpha_{load-DeNB} < \alpha_{DeNB-max}$  then
5:           Initiate handover process
6:       else terminate process
7:       end if
8:   end while

```

5.5 The Handover Process

The current handover procedure available in the LTE-Advanced standard [46] involves the UEs deciding when the handover process should take place. For this research, it was proposed that the handover process should proceed as shown in Figure 5.5. This handover process should be used only when certain BSs or relay nodes have to be made to go to sleep. For the normal handovers, the LTE standard has to be followed.

Where BS A has to go into sleep mode because its traffic load is less than the specified threshold, the controller within the BS at the centre of a 7 cell cluster, will send a-handover-required message to micro BS A, which responds by sending a-handover-required

acknowledgement packet. The controller then has to send a-handover-request message to BS B that has to take over. The process continues as outlined in Figure 5.5 until UE 1 has been handed over from micro BS A that has to switch OFF to BS B that is still active. The detailed handover procedure is as follows.

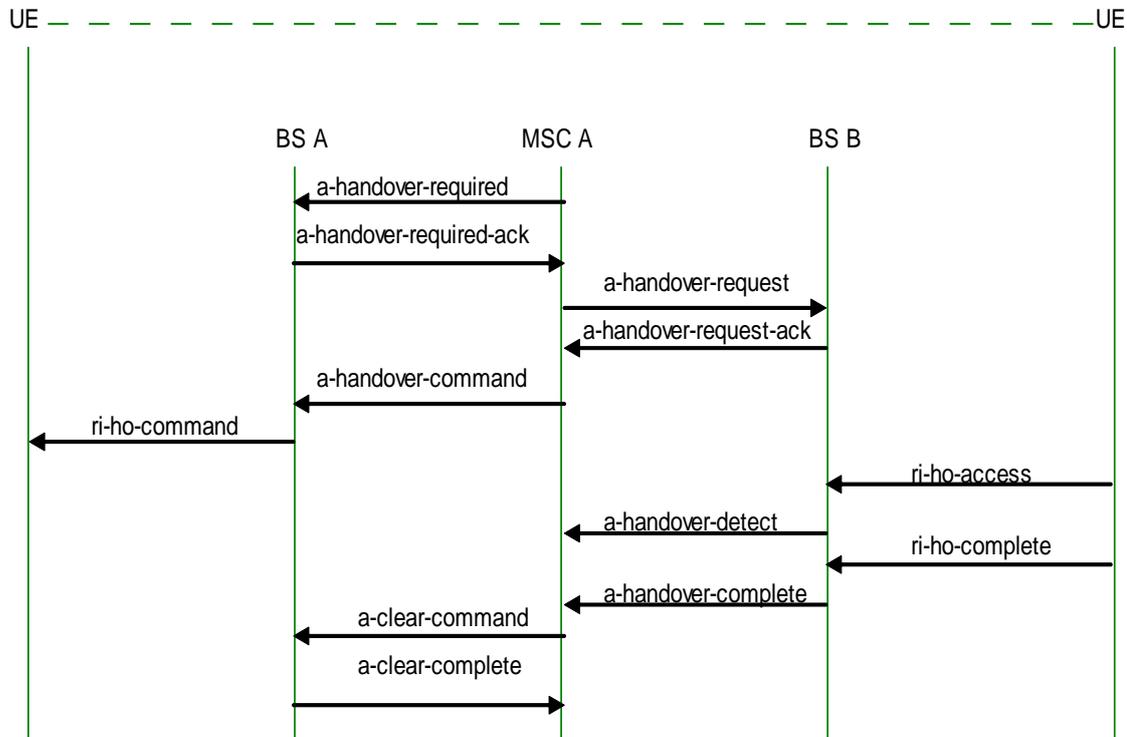


Figure 5.5: The handover procedure

5.5.1 The Detailed Handover Procedure

- 1: MSC (or control element) -A Sends A-HANDOVER-REQUIRED message to BS-A
- 2: BS A replies by sending A-HANDOVER-REQUIRED-ACK message
*//acknowledging that it has received A-HANDOVER-REQUIRED
//message.*
- 3: MSC-A then sends A-HANDOVER-REQUEST message to BS B
- 4: BS B replies by sending A-HANDOVER-REQUEST-ACK message
*// acknowledging receipt of A-HANDOVER-REQUEST
//message.*
- 5: MSC A Sends A-HANDOVER-COMMAND message to BS A
- 6: BS B then sends RI-HO-ACCESS message to UE *// this radio interface
// handover message contains a handover Reference Number
// for the UE to access the new radio resource*

- 7: UE sends RI-HO-ACCESS message to BSB // the UE uses the Handover
//Reference number to access the new radio resource
- 8: **if** UE is the expected and correct number, **then** BS B sends A-HANDOVER-DETECT message to MSC A
- 9: **else** BS B sends A-HANDOVER-HANDOVER-RESTART message to MSC A
- 10: **if** UE is communicating successfully with BE B, **then**
UE sends a RI-HO-COMPLETE message to BS B
- 11: **else** UE sends HANDOVER-FAILED to BSA
- 12: BS B then sends A-HANDOVER-COMPLETE message to MSC A
- 13: MSC responds by sending A-CLEAR-COMMAND to BS A // MSC A releases
//resources allocated to BS A using this clear command message
- 14: BS A then sends A-CLEAR-COMPLETE message to acknowledge release

5.6 Conclusion

In this chapter, the details of the design of the two algorithms, one for the actual switching OFF of the micro BSs and the other one for the re-association of UEs to neighbouring BSs have been presented. Flow diagrams followed by the respective pseudo codes for the algorithms have been given. The next chapter will provide the implementation details, test procedures and tests results.

CHAPTER SIX: IMPLEMENTATION, TEST PROCEDURES AND TEST RESULTS

6.1 Introduction

This chapter presents the implementation of the prototype, test procedures and simulation test results. The chapter also outlines the reasons why most of the tests were carried out. A brief description of the simulation tools that were used is also provided.

6.2 Simulation Environment

For the validation of the proposed algorithms, the proposed solution needed to be simulated and various tests done and results checked against research objectives. OMNeT++ version 4.6 [20] was chosen ahead of a number of network simulators. OMNeT++ is an object oriented modular discrete event network simulation framework that provides infrastructure and tools for writing simulations. In OMNeT++, modules are used to assemble simulation models. Modules can be simple or compound. Compound modules are formed when simple modules are connected together as shown in Figure 6.1. Connections link the modules' gates.

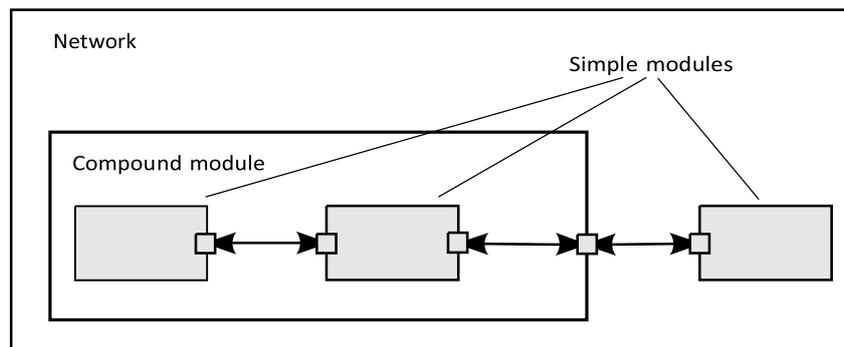


Figure 6.1: OMNeT++ module connections [20]

Gates are the interfaces that are found on the modules. Modules pass messages via well defined connections and are able to read and react to messages. The behaviour of modules comes from C++ application programs that reside inside the modules. For a network to be formed, several modules have to be connected together. A network in OMNeT++ is described using files that are written in a language called Network Description shortened to NED. OMNeT++ allows researchers to edit NED files graphically through a Graphical User Interface (GUI) or using a text editor. The GUI makes usage and debugging much easier compared to other simulators. Parameter values of a network are written in special files called initialisation files (.ini files).

An experiment that needs to be carried out goes into the .ini files. For this project, INET 2.3.0 [21], as a communications network simulation package for OMNeT++, was used.

INET framework provides models for both wired and wireless networking protocols which build upon one another creating simulation models in the process. INET makes it possible for researchers to use communication networks without having to write very complex protocols. All that the researchers are required to do is to link up ready-made modules together to form communication networks and then study the networks' behaviour. A simulation model starts with simple modules that are programmed using C++ to define their behavior and declared using the NED language. As simulations run, relevant statistics are collected in the form of output vectors and output scalars. Information that is recorded during a simulation goes into vectors and information that is recorded at the end of the simulation goes into scalars. SimuLTE 0.9.1 [22] was used to build the two models; the first one where no BS was switched OFF and the second one where some micro BSs were switched OFF.

SimuLTE is a simulation tool that can be used to carry out system level performance testing of LTE and LTE Advanced networks. It is a simulation tool that is written in C++ and it is fully customizable and allows researchers to simulate LTE networks in a bid to understand in detail the behaviour of the networks. SimuLTE builds on top of OMNeT++ and INET frameworks. It can be used to implement BSs and UEs connected together with other network elements like routers and servers to form networks. UEs and BSs are themselves made up of the following modules shown in Figure 6.2 [22]:

- (i) TCP and UDP applications
- (ii) TCP and UDP transport layers
- (iii) IP layer
- (iv) LTE NIC

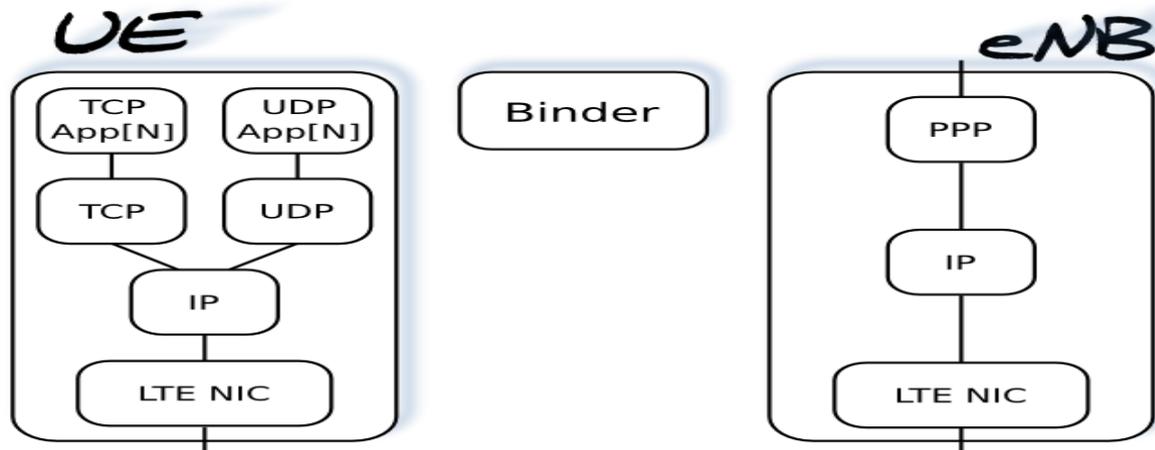


Figure 6.2: Modules making up the UEs and BSs [22]

The binder in Figure 6.2 is a special module that exists in SimuLTE that is used to store all information about all the other nodes in a network. It can also be used in the computation of inter-cell interference by UEs. These network elements are taken as compound elements.

For this project, a network that had 7 macro BSs, 7 micro BSs, 4 routers and a main server was built. This network is displayed in Figure 6.3 with all the network elements and their connections.

Communication amongst the BSs is established by enabling X2Enabled flag in the .ini file as follows; `**X2Enabled = ${X2 = true}`

Communication by BSs is exploited by having each BS have one X2 application for each peer that a BS has to be connected to. As shown in Figure 6.3, a star configuration was used to interconnect the BSs. For a star configuration, each BS has only one X2 interface connected to an X2 router. The X2 router forms the star centre. In the experimental network shown in Figure 6.3, BS iMacroBSC is made the master by appending the following code in the .ini file:

```
*.iMacroBSC.nic.BSsleeping_Manager.nodeType="BSSLEEPING_MASTER"
*.iMacroBSC.nic.BSsleeping_Manager.slavelist = "5 3"
```

With this configuration, iMacroBSC is the master and all the other BSs become slaves. BSs can then start exchanging messages. The slave BSs send their traffic load and their SEs to the master BS. The algorithms for BS sleeping and BS UE association reside in the master BS.

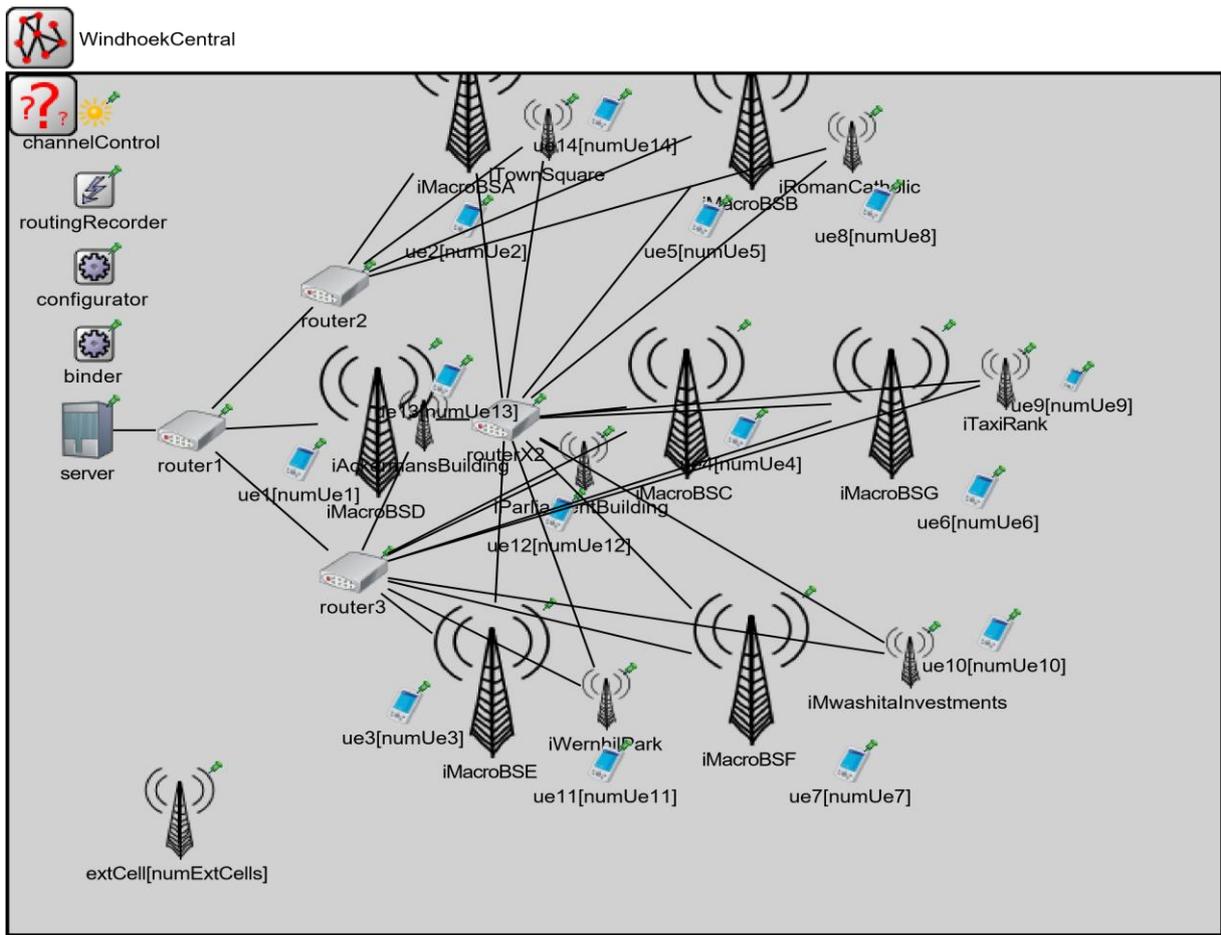


Figure 6.3: The experimental set up

The master BS uses these algorithms to decide which BS to switch OFF and which UE to reconnect to which BS. Figure 6.4 is a screenshot of the BSs exchanging messages. The screenshot shows seven macro BSs; iMacroBSA, iMacroBSB, iMacroBSC, iMacroBSD, iMacroBSE, iMacroBSF and iMacroBSGs in Central Windhoek, Namibia. These macro BSs provide adequate network coverage to the area in question. Co-located within the same coverage areas are seven micro BSs. The micro BSs improve capacity in hotspots in the coverage area. As BSs share their traffic load information and their SE values, they also share interference levels. HetNets are characterised by high interference levels. In the screenshot in Figure 6.4, iWernhillPark micro BS can be seen dispatching a message to the X2 router for onward transmission to iMacroBSC, the master BS.

Where a number of BSs are sharing the same frequency spectrum as is the case with HetNets, the impact of interference should be handled properly. High interference affects SINR.

6.3 The Simulation Tests

To test the viability of the proposed BS sleeping technology, much emphasis was placed on checking the effect of the technology on the QoS of a network. The performance metrics that were used were:

- (i) End-to-end delay.
- (ii) Channel quality indicator (CQI).
- (iii) Packet loss.
- (iv) Queue length.
- (v) Sum throughput.

In addition, the total power consumed was also measured. This was necessary so that it could be ascertained whether the proposed solution resulted in energy saving since the main aim of the research project was to design a scheme that would result in energy savings without compromising the QoS of the network. Table 6.1 shows the main simulation parameters that were used for the experimental set up. For the baseline case, the system was allowed to run for 0.04s simulation time. Output vectors captured the behaviour of the network over time and output scalars contained the following summary statistics:

- (i) Number of packets sent.
- (ii) Number of packets dropped.
- (iii) Average end-to-end delay of received packets.
- (iv) Peak throughput.
- (v) Queueing time.
- (vi) Queueing length.
- (vii) End-to-end-delay.

Table 6.1: Simulation parameters

Parameter	Value
1 UE distribution	Random
2 Max. UE Tx. Power	250mW
3 Macro BS Tx. Power	40000mW
4 Micro BS Tx. Power	20000mW
5 No. of macro sites	7
6 No. of micro sites	7
7 No. of UEs	53
8 Traffic type	VoIP

9. Network area	5000m by 5000m
10. Pass loss model	ITU-R, urban
11. Noise figure	5dB
13. Simulation time	0.04s

Figure 6.5 is a screenshot of the UEs sending and receiving messages to and from the main server through the BSs that the UEs are connected to. The experimental area shown in Figure 6.5 is populated with network elements that are exchanging messages. A message can be observed at iMacroBSD.

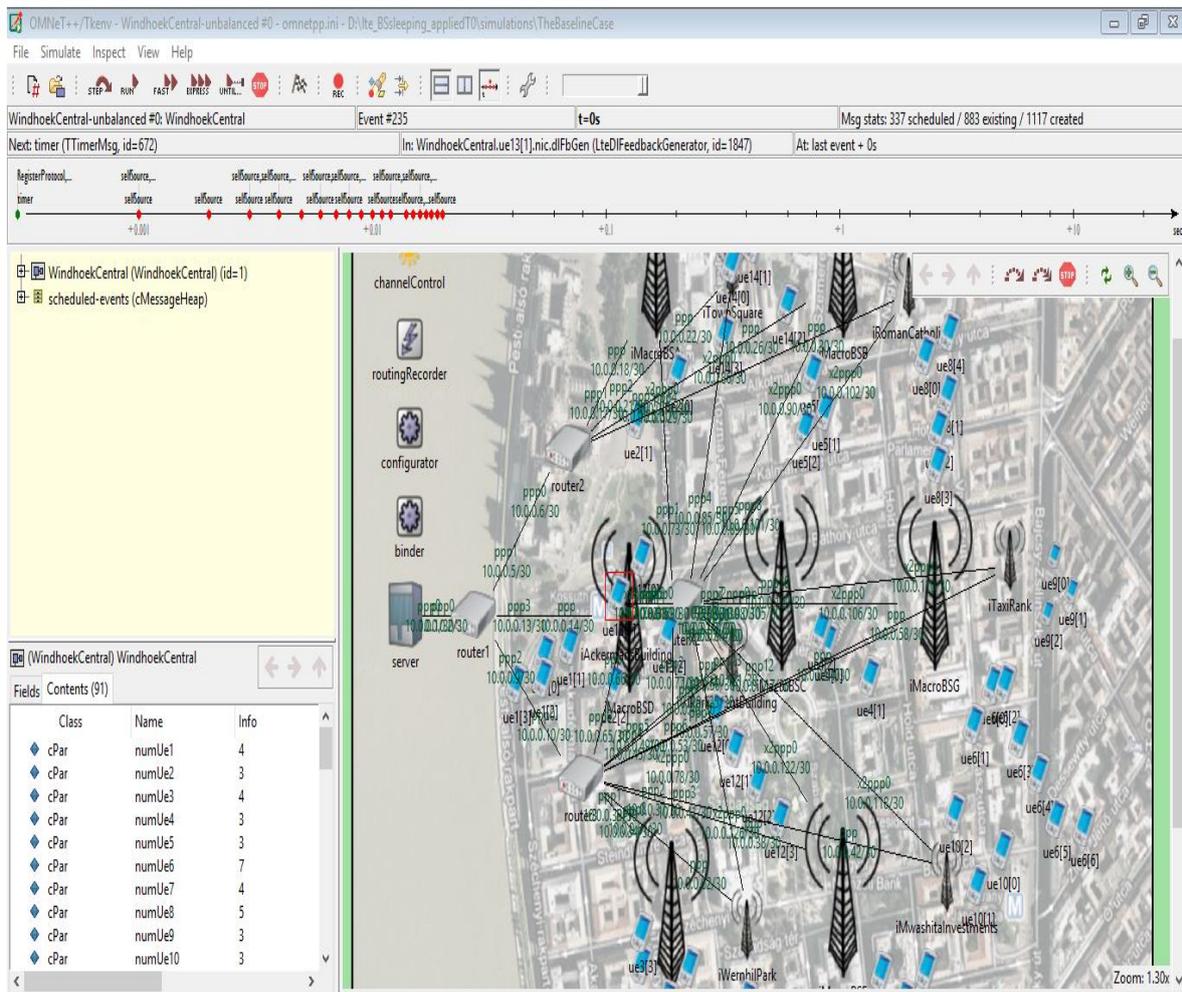


Figure 6.5: A screenshot of network elements exchanging messages

Table 6.2 gives the overview of the tests that were carried out and the reasons the tests were carried out.

Table 6.2: Experimental tests and results obtained

Item	Test type	Description of test	Results
1	Test 1	Measuring end-to-end delay of packets from UEs to the main server before and after implementation of scheme	Scheme did not affect the end-to-end delay of packets
2	Test 2	Checking whether the scheme affects queue length of packets at the main server	Packets queue length did not change after scheme was introduced
3	Test 3	Deterioration of channel quality was measured before and after application of scheme	No deterioration in quality of channels was observed
4	Test 4	Number of packets lost in the network was checked before and after implementation of energy saving scheme	No packets were lost before and after the BS sleeping algorithm was implemented
5	Test 5	The sum throughputs of 17 UEs were measured before and after scheme was introduced	No major change in the UEs' sum throughputs was observed
6	Test 6	Total network power transmitted by BSs was measured before and after introduction of the scheme to check whether the scheme improves energy efficiency	The scheme resulted in meaningful energy savings

6.3.1 End-to-end Delay

Some applications fail to perform well if delay is too large compared to some prescribed value. Many real-time applications like VoIP and video conferencing are not supported if the delay is too high. In this simulation, end-to-end delay by packets from the UEs to the main server was measured for 17 UEs before some micro BSs were switched OFF and measured again after certain micro BSs were switched OFF. The results are shown in Table 6.3 and plotted in Figure 6.6.

Table 6.3: End-to-end delay in seconds before and after some BSs were switched OFF

UE	End-to-end delay before some BSs were switched OFF	End-to-end delay after some BSs were switched OFF
1	0.00488	0.004977
2	0.00482	0.004819
3	0.00482	0.004819
4	0.00476	0.004797
5	0.0047	0.00488
6	0.00464	0.004977
7	0.00464	0.00488
8	0.0057	0.005819
9	0.00476	0.00476
10	0.00476	0.0048
11	0.004593	0.004598
12	0.004626	0.004638
13	0.00483	0.004836
14	0.0048	0.0048
15	0.00491	0.00499
16	0.004816	0.004829
17	0.004793	0.00482

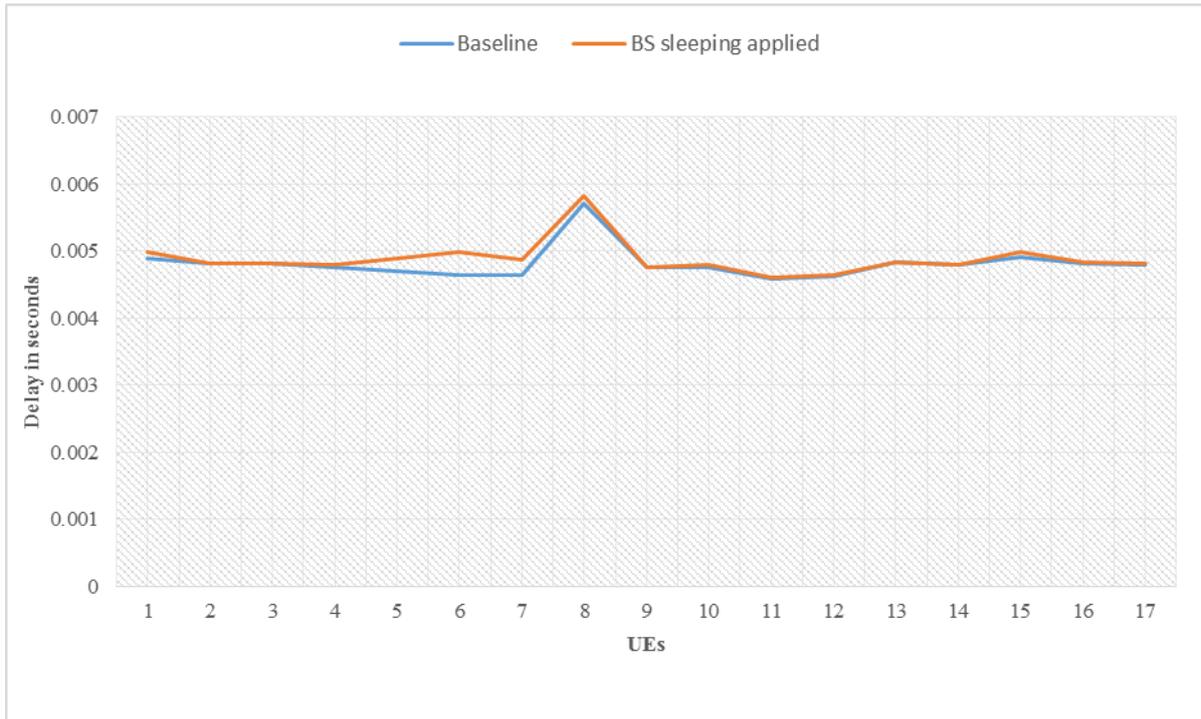


Figure 6.6: Mean VoIP packets end-to-end delay

6.3.2 Queue Length

A queueing system involves packets arriving and waiting for a service at a particular node. Queue length is one metric that can be used to measure QoS of a system. In this simulation, UEs generated VoIP traffic and sent this to the main server. The queue length was then measured before some micro BSs were switched OFF and measured again after some micro BSs were switched OFF. The results are shown in Table 6.4 and plotted in Figure 6.7.

Table: 6.4 Queue length before and after switching OFF some BSs

Simulation time in sec	Queue length before switching OFF some BSs	Queue length after switching OFF some BSs
0.00005	9	10
0.00011	7	8
0.00017	4	5
0.00023	3	3
0.00029	3	3

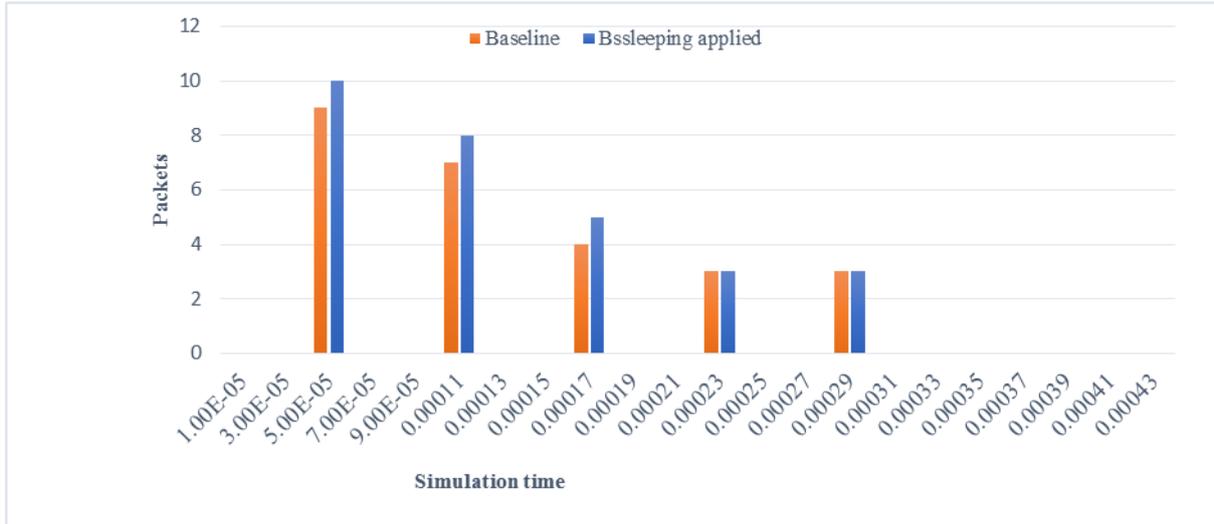


Figure 6.7: Queue length at the main server

6.3.3 Channel Quality Indicator

Channel quality indicator(CQI) is the information that the UEs send to the network. This information is some values given to how the UEs perceive the available channels. For LTE and LTE-A, CQI ranges between 0 and 15 with 15 indicating the best channel quality. The UE estimates the current link quality by use of signal to interference plus noise ratio (SINR) that is measured at the UE. For LTE and LTE-A, estimation of CQI by UE is achieved by use of a complicated formula [47] embedded in a chipset in the UE. For High Speed Downlink Packet Access (HSDPA), the UEs use the following formula (6.1) [48] to calculate CQI.

$$CQI = \begin{cases} 0 & \gamma \leq -3.96dB \\ \left(\frac{\gamma}{1.02} + 4.81 \right) & -3.96dB < \gamma \leq 26.04dB \\ 30 & \gamma > 26.04dB \end{cases} \quad (6.1)$$

Where:

γ is the downlink SINR on the link between the i th BS and the j th UE. For HSDPA, the CQI ranges from 0 to 30, with 30 giving the best channel quality.

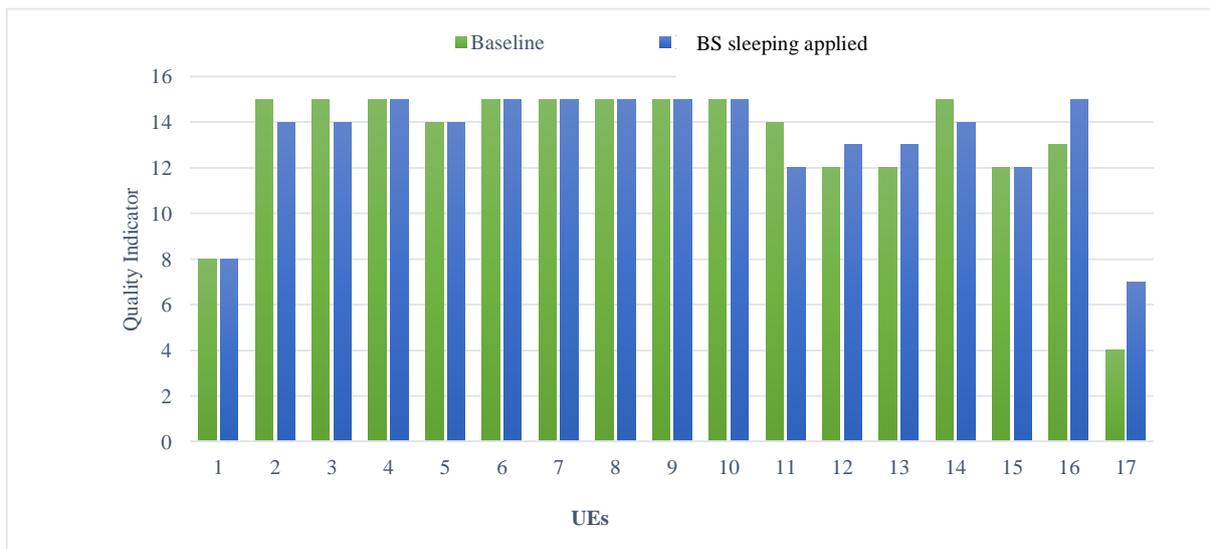
A CQI is used by a BS to adapt its scheduling and the modulation scheme to be used for a channel connecting to a UE. A periodic reporting was chosen for this simulation. This reporting has the advantage of producing CQI values that are current all the time. For the experiment, CQI values from 17 UEs were collected before some micro BSs that they were connected to were switched OFF.

The CQI values from the same 17 UEs were collected again after some micro BSs were switched OFF and handed over to active BSs. The results are shown in Table 6.5 and plotted in Figure 6.8.

Table 6.5: CQIs of 17 UEs before and after some BSs were switched OFF

UE	CQIs of 17 UEs before some BSs were switched OFF	CQIs of 17 UEs after some BSs were switched OFF
1	8	8
2	15	14
3	15	14
4	15	15
5	14	14
6	15	15
7	15	15
8	15	15
9	15	15
10	15	15
11	14	12
12	12	13
13	12	13
14	15	14
15	12	12
16	13	15
17	4	7

Figure 6.8: CQIs for 17 UEs



6.3.4 UEs' Sum Throughputs

To further evaluate the performance of the scheme, sum throughputs for 17 UEs were taken before some micro BSs were switched OFF and retaken after some micro BSs were put to sleep. The results that were obtained are shown in in Table 6.6 and plotted in Figure 6.9

Table 6.6: Sum throughput before and after switching OFF some BSs

UE	Sum throughput before some BSs were switched OFF	Sum throughput after some BSs were switched OFF
1	40	39
2	40	40
3	38	38
4	37	37
5	40	39
6	40	40
7	36	36
8	38	37
9	38	38
10	40	40
11	36	36
12	39	40
13	38	38
14	37	37
15	36	36
16	40	40
17	40	40

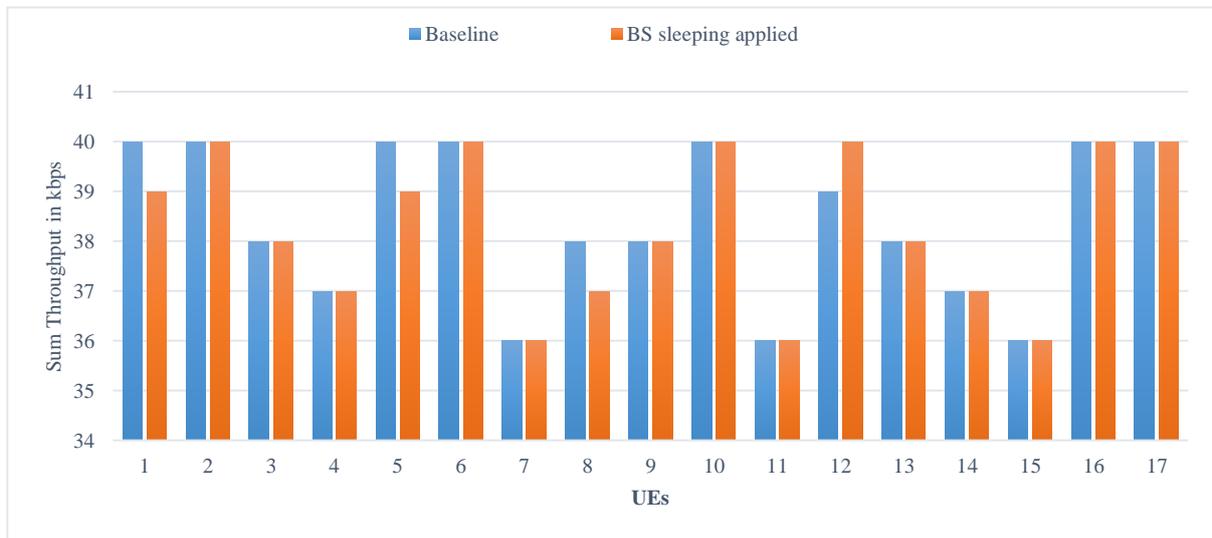


Figure 6.9: UEs' sum throughput

6.3.5 Energy Consumed

Table 6.7 and Figure 6.10 show the energy savings that were achieved by switching OFF more and more micro BSs. It can be seen that energy that is saved increase as the number of micro BSs being switched OFF increases.

Table: 6.7 Total transmitted energy as BSs are switched OFF

Number of Micro BSs switched OFF	Total transmitted energy by all BSs in a sector (in Joules)
1	403
2	386
3	369
4	352
5	335
6	318
7	301

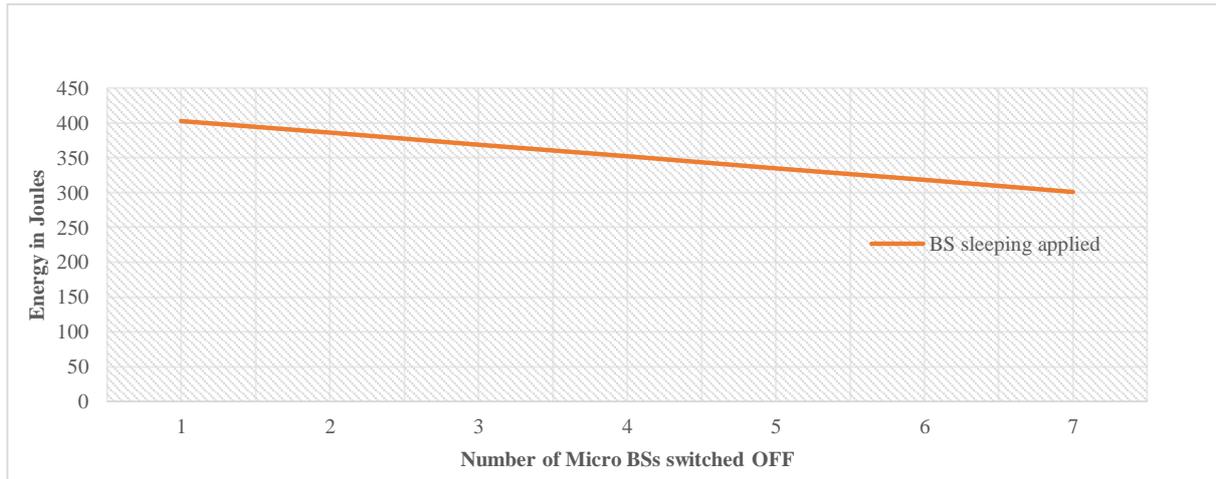


Figure 6.10: Energy consumed versus number of BSs switched OFF

6.4 Conclusion

In this chapter, the implementation and test procedures have been outlined. The results that were obtained after the simulations were carried out have been presented. In the next chapter, these results are analysed.

CHAPTER SEVEN: ANALYSIS OF THE TEST RESULTS

7.1 Introduction

This chapter presents the analysis of the results obtained. The results obtained from the simulations tests carried out are analysed in a bid to determine whether the proposed solution performed as per design.

7.2 End-to-end Delay

End-to-end delay is very important in the monitoring of the performance of mobile networks. It refers to the delay a packet encounters in the network from the time a packet is generated by a source node to the time an acknowledgement or feedback packet is received by the source node. This includes the queuing delays at intermediate network elements. End-to end packet delay is a very important QoS metric. VoIP and teleconferencing, shown in Figure 7.1, are very good examples of delay-sensitive applications.



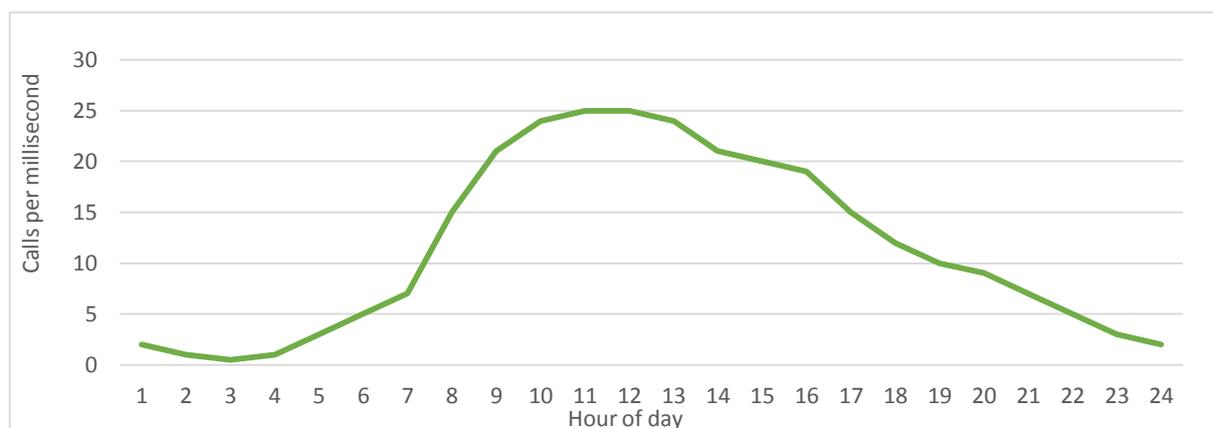
Figure 7.1: Teleconferencing using mobile devices [49]

For next generation mobile networks, technologies are being worked out that will further improve quality of teleconferencing. Any technology that improves the energy efficiency of a network at the expense of end-to-end delay of packets will not be tolerated by network operators. The solution that was designed involves the switching OFF of some micro BSs. When some micro BSs are switched OFF in a network, the network reconfigures and this might affect the end-to-end delay of packets.

Figure 6.6, shows that the end-to-end delay that was experienced before application of the BS sleeping algorithm, did not change very much compared to the end-to-end delay that was experienced after the application of the BS sleeping algorithm. This means that quality delivery of delay sensitive applications like voice and video is not affected by switching OFF some heavily underutilised micro BSs using the proposed solution. The results obtained show that QoS of the network was not degraded by these small changes in the end-to-end delay simply because the changes were very small.

7.3 Packet Loss

Packet loss in a cellular network is usually caused by congestion. Packets are dropped by network elements such as routers and gateways when they arrive at a network element at a rate that is higher than the rate at which packets will be leaving the network element. Audio and video applications normally fail to perform correctly if the network is leaky and some packets are occasionally being lost. To maintain an acceptable level of QoS, the number of packets lost in a network should be kept to a very minimum number. The minimum number of packets that can be lost without affecting the QoS of the network depends on the type of data being transported. For this experiment, no packets were lost before and after some micro BSs were switched OFF. This was something that was expected because the total traffic in the area in question was very low during the time that the experiment was conducted. Low traffic was generated by the model. This was a deliberate action since the model was supposed to mimic a real mobile network during a period of low traffic. Figure 7.2 shows a typical variation of traffic



on a weekday in an urban area.

Figure 7.2: Cellular traffic load variation during a 24 hr weekday [23]

Since the proposed technique starts switching OFF BSs during the time when traffic is very minimal, not many packets can be expected to be lost, hence QoS cannot be expected to be compromised because of packet loss. Even though some micro BSs would have been switched OFF, congestion cannot be expected because of the low traffic that exists during the time the BS sleeping scheme comes into effect.

7.4 Queue Length

Queue length in a mobile network is designed to accommodate temporary congestion. Data that cannot instantly be delivered to the destination is temporarily stored in buffers until bandwidth becomes available. Queueing takes place on outbound interfaces. A server can queue packets that are coming out going to UEs. Monitoring the behaviour of the queues can give an insight into the performance of the network. If the queues are getting too long resulting in some packets being lost, this indicates that all is not well in the network. This might mean that traffic is being generated at a far higher rate than the rate that the network elements are prepared to process. From Figure 6.7, it can be observed that the queue lengths were slightly higher at the beginning but this was for both the baseline case and after the BS sleeping algorithm was applied. This means that switching OFF some BSs when traffic is low, does not affect the queue length. Queue length can be used as a performance metric in a mobile network. As can be seen from the results, it can safely be said that since queue length is not being affected by the BS sleeping scheme, it means the proposed scheme does not affect the QoS of the network.

7.5 Channel Quality Indicator

For each Transmission Time Interval, TTI, a BS scheduler, allocates resource blocks and modulation schemes to UEs based on the CQI information that the BSs get from the UEs. CQI values roughly translate to the maximum possible data rate a UE is able to receive after taking into account the prevailing signal to noise ratio. CQI values also depend on the prevailing traffic and the interference levels. CQI values give an indication of how the radio conditions would have changed in the eyes of the UE. It can be taken as the summary of the channel condition. It should be pointed out that this CQI information that the UE sends to the BS only serves as

general information to be used by the BS in the decision making process. The BS might decide not to use it.

However, this CQI information is very important in the simulation that was undertaken. It is very useful as it gives an indication of the quality of connection between a BS and a UE. On switching OFF some micro BSs, UEs are allocated new channels. It was therefore important to check the effect the BS sleeping scheme had on the general condition of the newly allocated channels as reported by the UEs.

From Figure 6.8, it can be observed that the CQI values that were reported by 17 UEs before some micro BSs were switched OFF remained almost unchanged after some micro BSs were switched OFF. This can be attributed to the fact that when some micro BSs are switched OFF in a cellular network, interference levels decrease. This explains why for some of the UEs, there was actually an increase in CQI values. Close to 50% of the UEs' CQI values did not change at all after some micro BSs were switched OFF. The results that were obtained indicate that the scheme did not affect the quality of the channels as perceived by the UEs. If the quality of the channels was not affected by the BS sleeping scheme, it can be deduced that the BS sleeping scheme did not compromise the QoS of the network by way of degrading the quality of the channels.

7.6 Throughput

With network operators offering ubiquitous services seamlessly, users expect to access the Internet anywhere anytime at tremendously high speeds. This means that when it comes to issues surrounding throughput, it's only expected to go up and never down otherwise the network QoS would be impacted negatively. QoS can quantitatively be measured by measuring the system throughput. The UEs' sum throughputs refer to the packets that are delivered successfully to the individual UEs in a given amount of time. From Figure 6.9, the sum throughputs for the 17 UEs did not change very much when some micro BSs were switched OFF. Over 75% of the UEs' throughputs did not change at all. The sum throughputs for most of the UEs were not affected most likely because they were handed to BSs that were not so heavily loaded. This is a very probable scenario since the BS sleeping algorithm springs into action when traffic in a network is very low. It can also be observed on Figure 6.9 that for one

UE, there was actually an increase in its sum throughput. This could have happened because of reduced interference that is experienced when some micro BSs are switched OFF.

The UE could also have been re-associated with a BS handling less traffic load than the BS to which the UE was previously connected. The results show that throughput of the network was not affected by the BS sleeping scheme. Since the throughput was not affected by the BS sleeping algorithm, it can be deduced that the BS sleeping scheme that was designed did not degrade the QoS of the network.

7.7 Power Consumption

Figure 6.10 shows that as more and more micro BSs are switched OFF, more and more energy is saved. The scheme does not produce a very high energy saving compared to similar schemes, but this one is better in that energy is saved without compromising the QoS of a network unlike with other schemes. Not many network operators will sacrifice QoS at the expense of energy saving. It is better to have a system that is accepted by industry but has modest energy saving than to have a system that results in huge energy savings but degrades the QoS of the network.

7.8 Conclusion

In this chapter, the test results obtained using the simulator were analysed in detail and viability of the designed solution discussed. The next chapter concludes the dissertation by highlighting the strengths and weaknesses of the designed solution and proposes further work.

CHAPTER EIGHT: CONCLUSION

8.1 Introduction

This chapter concludes the dissertation and explains what was achieved, what was not and why it was not achieved. A very brief summary of the prototype and its performances is also presented. An energy saving scheme, deployable in HetNets, was proposed, designed, simulated and its performance analysed. The scheme springs into action each time cellular traffic goes beyond a certain threshold in a cellular network. Micro BSs with traffic load below the threshold are systematically switched OFF provided the action does not lead into a degradation of QoS of the network. Before a micro BS is switched OFF, UEs have to be re-associated with accommodating neighbouring BSs in an energy efficient manner. Sleeping micro BSs are only re-awoken when the cellular traffic reaches a prescribed threshold.

8.2 Implications of the Research

The main aim of the research was to investigate methods/schemes that could be deployed in Base stations such that as their number increases exponentially, CO₂ emissions associated with this rapid growth are curbed in addition to ensuring that acceptable level of QoS is maintained. This was achieved and the research showed that it is possible to switch OFF certain micro BSs in a HetNet that are lightly loaded without affecting the QoS of the network. Extensive simulations were carried out and it was observed that the following performance metrics were not negatively affected.

- (i) Throughput.
- (ii) End-to-end delay.
- (iii) Channel quality indicator.
- (iv) Queue length.
- (v) Packet loss.

Since these fundamental performance metrics were not negatively affected, it was concluded that the proposed BS sleeping scheme did not degrade the QoS of a cellular network.

The results that were obtained in this research can be of interest to a number of groups. Since the results showed that energy can be saved without compromising the QoS of the network, network operators are likely to take the technology on board so that their OPEX can be reduced.

The network operators are likely to spruce up their corporate image by announcing that they are going Green with the use of energy efficient technologies. The BS sleeping scheme is also likely to be of interest to climate activists. Since the scheme saves energy, CO₂ emissions reduction will follow and this is sweet news to climate activists who are then likely to nudge the network operators into fully embracing the BS sleeping technology. The findings of this research can also be of interest to standards making organisations like the 3GPP to consider co-opting the technology when formulating standards for the next generation mobile networks.

8.3 Applications of the Research

The research study managed to put together literature on Green Cellular Networks and this can be useful to researchers when designing new technologies for next generation networks. The extensive literature review that was presented in this dissertation can also help researchers in formulating feasible BS sleeping technologies for future networks.

The BS sleeping technology can be used by network operators when they are upgrading their networks. Traditionally, cellular networks have been upgraded by reducing sizes of macro sites thereby increasing the number of macro BSs in the process. This should now be a thing of the past. Network operators should just leave macro cells intact, identify hotspots and deploy micro BSs in these areas. They should then use the BS sleeping technology to switch OFF the micro BSs when not needed. Micro BSs are easy to mount, cheaper and finding locations to deploy them is not as difficult as finding a site to deploy a macro BS. Micro BSs can be mounted inside buildings.

8.4 Recommendations for Further Work

SimuLTE 0.9.1 was used together with INET 2.3.0 and OMNeT++ v4.6 for the simulation of models used to test the proposed BS sleeping scheme. By using SimuLTE, it was only possible to simulate LTE-A networks. LTE-A is a bridge between 4G and 5G. This means the results that were posted by the BS sleeping technology when simulated using LTE-Advanced, might slightly change if it's used on true 5G. 5G will be handling Internet of Things traffic and it is strongly recommended that the technology be simulated using a simulator that can generate true 5G traffic. Outside cells were used in this project to generate interference but this can be different to the interference that is experienced in 5G networks.

8.5 Conclusion

This chapter concluded the thesis by providing a summary of what was achieved, implications of the research, applications of the research and recommendations for further work. All the four objectives of the research were met:

- (i) Existing and emerging technologies on BS energy saving schemes in Green Cellular Networks were critically assessed.
- (ii) An algorithm that could be used to put BSs to sleep during the time traffic load was very low was successfully developed.
- (iii) An algorithm that could be used to re-associate mobile users in cells where BSs would have been made to go to sleep, to active neighbouring BSs was developed.
- (v) The algorithms that were developed did not degrade the QoS of a cellular network.

It can be concluded that the objectives of the research to develop a BS sleeping algorithm which does not degrade network QoS was achieved though, more research needs to be done to simulate it in 5G network.

REFERENCES

- [1] A. Conte, A. Feki, L. Chiaraviglio, D. Ciullo, M. Meo & M. A. Masan, "Cell wilting and blossoming for energy efficiency," *Wireless Communications, IEEE*, vol. 18, Issue 5, pp. 50-57, Oct. 2013.
- [2] Cisco. (2016, Feb). Cisco visual networking index: Global mobile data traffic forecast update, 2015-2020. [Online]. Available: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>
- [3] O. Vermesan & P. Firess, Eds., "Internet of Things—Converging technologies for smart environments and integrated ecosystems," New York: CRC Press, 2013, pp. 8-35.
- [4] J. Wu, S. Rangan & H. Zhang, Eds., "Green communications: Theoretical Fundamentals, algorithms and applications," New York: CRC Press, 2013, pp. 280-304.
- [5] R. E. H Sims, H.H Roger & K. Gregory, "Carbon emission and mitigation cost comparison between fossil fuel, nuclear and renewable resources for electricity generation," *Energy Policy* 31, (2013), pp. 1315-1326.
- [6] Y. Fan, X. Wang and C. H. J Peter, "Green Cellular.... Towards supportable networks," [Online]. Available: http://www.mobile.ecei.tohoku.ac.jp/COE/seminar_2010_06_2/Green.pdf
- [7] M.H Asharif, R. Nordin & M. Ishmail. "A survey of green radio communication networks: Techniques and recent advances," *Journal of computer networks and communications*, 2013, vol. 2013, Article 45389, 13 pages, Hindawi Publishing Corporation, Nov. 2013.
- [8] E. Hossain, V.K Bhargava, G.P Fettweis (Ed), "Green Radio Communication Networks," Cambridge: Cambridge University Press, 2012, pp.1-23.
- [9] EARTH. [Online]. Available: <https://www.ict-earth.eu/>
- [10] Green Touch. [Online]. Available: <http://www.greentouch.org/>
- [11] An overview of MIMO systems in wireless communication systems. [Online]. Available : <http://www.iet.ntnu.no/projects/beats/documents/mimo.pdf>
- [12] J. Wu, S. Rangan & H. Zhang, "Green Communications : Theoretical Fundamentals, Algorithms and Applications," CRC Press, Oct 2013, pp.405-433.
- [13] R. Murthy, & C. Kavitha. "A survey of Green base stations in cellular networks," *IRACST International Journal of Computer Networks and Wireless Communications, (IJCNWC)*, vol.2, Issue 2, pp. 232-236. Jun. 2012.
- [14] O. Blume, H. Eckhardt, S. Klein, E. Kuehn & W.M. Wajda. "Energy savings in mobile networks based on adaptation to traffic statistics," *Bell Labs Technical Journal* vol. 15, Issue 5, pp. 77-94, Dec 2010.
- [15] A review of corporate responsibility from Vodafone UK. [Online]. Available : http://www.vodafone.co.uk/cs/groups/configfiles/documents/contentdocuments/cr_report_2008_09.pdf
- [16] R. Li, Z. Zhao, Y. Wei & H. Zhang, "GMPAD : a grid based energy saving scheme with predicted traffic load guidance for cellular networks," in *proceedings of IEEE ICC 2012*, Ottawa 2012, pp. 1160-1164.
- [17] Z. Niu, "TANGO: traffic-aware network planning and green operation," *IEEE Wireless Communications*. vol.18, no. 5, 2011, pp. 25-29, Oct. 2011.

- [18] G. Auer, V. Giannini, I. Godor, M. Olsson, M.A Imran, D.Sabella, M.J. Gonzalez, O. Blume, A. Fehske, J. A Rubio, P. Frenger & C. Derser. How much energy is needed to run a wireless network? [Online]. Available : <http://epubs.surrey.ac.uk/735354/1/EARTH%20Book%20Chapter.pdf>
- [19] K. Dooley, "Simulation research methods," *Companion to organisations*, J. Baum (Ed) London: Blackwell, p.829-848, Dec. 2002.
- [20] OMNeT++, An object oriented modular discrete event network simulation framework, 2016. [Online]. Available: <https://omnetpp.org>
- [21] INET Framework. An open source model suite for wired and wireless mobile networks, Available: <https://inet.omnetpp.org>
- [22] SimuLTE. LTE User Plane Simulation Model for INET & OMNeT++, [Online]. Available: www.simulte.com
- [23] W. S Soh & H. S Kim, "QoS provisioning in cellular networks based on mobility prediction techniques," *IEEE Communications Magazine*, vol. 41 Issue 1, Jan. 2003.
- [24] S. Adibi, R. Jain & S. Parekh, "Quality of service architectures for wireless networks: Performance metrics and management," Information Science Reference, Hershey, USA, 2010.
- [25] Quality of Service (QoS) and policy management in mobile data networks: Validating service quality to ensure subscriber Quality of Experience (QoE). [Online]. Available: http://www.ixiacom.com/sites/default/files/resourcers/whitepaper/policy_management.pdf
- [26] A. Bousia, A. Antonopoulos, L. Alonso & C. Verikoukis, "'Green' Distance-aware base station sleeping algorithm in LTE- advanced," in *IEEE International Conference on Communications*, Jun. 2012, pp.1347-1351.
- [27] Z. Niu, Y. Wu, J. Gong & Z. Yang, "Cell zooming for cost-efficient Green Cellular Networks," *IEEE Communications Magazine*, vol. 48, Issue 11, pp. 74-79, Nov. 2013.
- [28] S. Zhou, J. Gong & Z. Niu, "Sleep control for base stations powered by heterogeneous energy sources," in *International Conference on ICT Convergence (ICTC)*, Oct. 2013, pp. 666-670.
- [29] K. Son & B. Krishnamachari, "Dynamic base station switching-on/OFF strategies for green cellular networks," *IEEE transactions on wireless communications*, vol. 12, Issue 5, pp. 2126-2136, May 2013.
- [30] I. Aydin, H. Yanikomeroglu & U. Aygolu, "User-aware switch-OFF algorithms," in *11th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug. 2015, pp. 112-117.
- [31] W. Mwashita and M.O Odhiambo, "A survey on Base station sleeping technologies for Green Cellular Networks," *International Conference on Wireless Networks*, Las Vegas, Jul 2015, pp. 227-233.
- [32] W. Mwashita and M.O Odhiambo, "Designing and analysing a technique to switch ON/OFF Base Stations for Green Cellular networks," *International Journal of Research in Engineering and Technology*, vol. 4, Issue 07, Jul 2015, pp. 322-328.
- [33] S. Vandriss & V. Killiches. Power optimization in wireless heterogeneous networks. [Online]. Available: <http://www.lsi.com>
- [34] P. Dini, M. Miozzo, N. Bui & N. Baldo, "A model to analyze the energy savings of base station sleeping in LTE HetNets," in *IEEE International Conference on Green computing and communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, Aug. 2013, pp. 1375-1380.

- [35] High capacity indoor wireless solutions: Pico cell or femto cell? [Online]. Available: <https://www.fujitsu.com/us/Images/High-Capacity-Indoor-Wireless.pdf>
- [36] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermark, M. Olsson, M. A. Imran & D. Sabella, O. Blume, W. Wajda, Y. Jading & E. Katranaras, "D2. 3: Energy efficiency analysis of the reference systems, areas of improvements and target breakdown," *INFOSOCT-247733 EARTH (Energy Aware Radio and Technologies)*, No. 2010.
- [37] South Africa, 2012, Ministry of Energy, "A survey of energy-related behaviour and perceptions in South Africa: The residential sector," 2012.
- [38] E. Oh & B. Krishnamachari, "Energy savings through dynamic base station switching in cellular wireless access networks," *Global Telecommunications Conference (GLOBECOM 2010)*, 2010 IEEE, vol. 2, Issue 1, pp. 1-5, Dec. 2010.
- [39] J. Gong, S. Zhou, Z. Niu & P. Yang, "Traffic-aware base station sleeping in dense cellular networks," *18th International Workshop on Quality of Service (IWQoS)*, vol. 3, Issue 3 pp. 1-2, Jun. 2010.
- [40] G. Lee, H. Kim, Y. T. Kim & B. H. Kim, "Delaunay triangulation based green base station for self-organising networks," *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things, Cyber Physical and Social Computing*. Sept. 2013.
- [41] A. S. Alam, L. S. Dooley, A. S. Poulton & Y. Ji, "Energy savings using an adaptive base station relay station switching paradigm," in *International Conference on Wireless Communications and Signal Processing (WCSP'12)*, 25-27 Oct. 2012.
- [42] M. A. Marsan, L. Chiaraviglio, D. Ciulo & M. Meo, "Optimal energy savings in cellular access networks," *Communications Workshops, 2009, ICC Workshops 2009. IEEE International Conference on*, pp. 1-5, Jun. 2009.
- [43] W. Goddard, "Introduction to algorithms; Part 1: Divide and Conquer- Sorting and searching," Clemson University, 2004.
- [44] J. Wannstrom, "The mobile broadband standard," [Online]. Available: <http://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced>.
- [45] P. J. Chung, "Green Radio-The case of more efficient cellular base stations," University of Edinburgh, UK. http://www.mobilevce.com/dloads-publ/GR_Globecom2010.pdf
- [46] ETSI TS 123009 V12.0.0 (2014-10), "Digital telecommunications system (Phase 2+) Universal Mobile Telecommunications System (UMTS), Handover procedures," 3GPP TS 23.009 version 12.0.0 Release 12.
- [47] LTE Quick Reference. [Online], Available: www.sharetechnote.com/html/Handbook_LTE_CQI.html
- [48] S. Morosi, P. Piunti and E. D. Re, "Sleep mode management in cellular networks: a traffic based technique enabling telecommunications technologies," *Transactions on emerging telecommunications technologies*, doi: 10.1002/ett.2621, Feb, 2013.
- [49] Multiparty calling. [Online], Available: <https://lifesize.com/en/products/video-conferencing-infrastructure/software-based-bridge>

APPENDIX A: LIST OF PUBLICATIONS

The following is a list of publications that are related to this thesis.

- [1] Weston Mwashita, Marcel O Odhiambo. A Survey of Base Station Sleeping Technologies for Green Cellular Networks. International Conference on Wireless Networks, (ICWN 2015), The 2015 World Congress in Computer Science, Computer Engineering, and Applied Computing WORLDCOMP'15, July 27th – 30th, 2015, Las Vegas, USA. pp227-233. ISBN: 1-60132-424-3/©CSREA Press, USA 2015.
<http://worldcomp-proceedings.com/proc/p2015/ICW6059.pdf>
- [2] Weston Mwashita, Marcel O Odhiambo. Designing and Analysing a Technique to Switch ON/OFF Base Stations for Green Cellular Networks. *IJRET-International Journal of Research in Engineering and Technology (IJRET)*, Volume: 04 Issue: 07, July-2015, pp322-328, e-ISSN: 2319-1163, p-ISSN: 2321-7308, pp322 – 328,
<http://esatjournals.net/ijret/2015v04/i07/IJRET20150407053.pdf>

Appendix B: WindhoekCentral.ned

```
package lte.simulations.Networks;

import inet.networklayer.autorouting.ipv4.IPv4NetworkConfigurator;
import inet.networklayer.ipv4.RoutingTableRecorder;
import inet.nodes.ethernet.Eth10M;
import inet.nodes.inet.Router;
import inet.nodes.inet.StandardHost;
import inet.world.radio.ChannelControl;
import lte.corenetwork.binder.LteBinder;
import lte.corenetwork.nodes.eNodeB;
import lte.corenetwork.nodes.Ue;
import lte.corenetwork.nodes.ExtCell;

network WindhoekCentral
{
    parameters:
        @display("i=block/network2;bgb=710,558;bgi=background/windhoek;is=n");
        int numUe1 = default(0);
        int numUe2 = default(0);
        int numUe3 = default(0);
        int numUe4 = default(0);
        int numUe5 = default(0);
        int numUe6 = default(0);
        int numUe7 = default(0);
        int numUe8 = default(0);
        int numUe9 = default(0);
        int numUe10 = default(0);
        int numUe11 = default(0);
        int numUe12 = default(0);
        int numUe13 = default(0);
        int numUe14 = default(0);

        int numExtCells = default(0);

    submodules:
        channelControl: ChannelControl {
            @display("p=50,25;is=s");
        }
        routingRecorder: RoutingTableRecorder {
            @display("p=50,75;is=s");
        }
        configurator: IPv4NetworkConfigurator {
            @display("p=50,125");
            config = xmlDoc("multi.xml");
        }
        binder: LteBinder {
            @display("p=50,175;is=s");
        }
        server: StandardHost {
            @display("p=46,231;is=n;i=device/server");
        }
        router1: Router {
            @display("p=107,231;i=device/smallrouter");
        }
}
```

```

router2: Router {
    @display("p=191,137;i=device/smallrouter");
}
router3: Router {
    @display("p=202,323;i=device/smallrouter");
}
routerX2: Router {
    @display("p=291,224;i=device/smallrouter");
}
iMacroBSD: eNodeB {
    @display("p=216,225;is=v1");
    txPower = 40000mw;
}
iMacroBSA: eNodeB {
    @display("p=269,14;is=v1");
    txPower = 40000mw;
}
iMacroBSE: eNodeB {
    @display("p=283,394;is=v1");
    txPower = 40000mw;
}
ue1[numUe1]: Ue {
    @display("p=171,252;is=s");
}
ue2[numUe2]: Ue {
    @display("p=269,94;is=s");
}
ue3[numUe3]: Ue {
    @display("p=232,410;is=s");
}
extCell[numExtCells]: ExtCell {
    @display("p=100,480;is=l");
}
iMacroBSC: eNodeB {
    @display("p=397,213;is=v1");
    txPower = 40000mw;
}
iMacroBSB: eNodeB {
    @display("p=435,25;is=v1");
    txPower = 40000mw;
}
iMacroBSG: eNodeB {
    @display("p=517,213;is=v1");
    txPower = 40000mw;
}
iMacroBSF: eNodeB {
    @display("p=435,386;is=v1");
    txPower = 40000mw;
}
ue4[numUe4]: Ue {
    @display("p=430,232;is=s");
}
ue5[numUe5]: Ue {
    @display("p=409,94;is=s");
}
ue6[numUe6]: Ue {
    @display("p=569,270;is=s");
}

```

```

}
ue7[numUe7]: Ue {
    @display("p=494,448;is=s");
}
iMwashitaInvestments: eNodeB {
    @display("p=528,379;is=n");
    txPower = 20000mw;
}
iTownSquare: eNodeB {
    @display("p=316,36;is=n");
    txPower = 20000mw;
}
iRomanCatholic: eNodeB {
    @display("p=494,44;is=n");
    txPower = 20000mw;
}
iTaxiRank: eNodeB {
    @display("p=584,198;is=n");
    txPower = 20000mw;
}
iAckermansBuilding: eNodeB {
    @display("p=243,225;is=n");
    txPower = 20000mw;
}
iWernhilPark: eNodeB {
    @display("p=350,406;is=n");
    txPower = 20000mw;
}
iParliamentBuilding: eNodeB {
    @display("p=337,252;is=n");
    txPower = 20000mw;
}
ue8[numUe8]: Ue {
    @display("p=509,84;is=s");
}
ue9[numUe9]: Ue {
    @display("p=624,198;is=vs");
}
ue10[numUe10]: Ue {
    @display("p=578,363;is=s");
}
ue11[numUe11]: Ue {
    @display("p=350,448;is=s");
}
ue12[numUe12]: Ue {
    @display("p=323,285;is=s");
}
ue13[numUe13]: Ue {
    @display("p=257,198;is=s");
}
ue14[numUe14]: Ue {
    @display("p=350,25;is=s");
}
}

```

connections:

```

server.pppg++ <--> Eth10M <--> router1.pppg++;
router1.pppg++ <--> Eth10M <--> router2.pppg++;

```

```

router1.pppg++ <--> Eth10M <--> router3.pppg++;
router1.pppg++ <--> Eth10M <--> iMacroBSD.ppp;
router2.pppg++ <--> Eth10M <--> iMacroBSA.ppp;
router3.pppg++ <--> Eth10M <--> iMacroBSE.ppp;
router3.pppg++ <--> Eth10M <--> iMacroBSF.ppp;
router3.pppg++ <--> Eth10M <--> iMwashitaInvestments.ppp;
router3.pppg++ <--> Eth10M <--> iParliamentBuilding.ppp;
router3.pppg++ <--> Eth10M <--> iMacroBSC.ppp;
router3.pppg++ <--> Eth10M <--> iMacroBSG.ppp;
router3.pppg++ <--> Eth10M <--> iTaxiRank.ppp;
router2.pppg++ <--> Eth10M <--> iTownSquare.ppp;
router2.pppg++ <--> Eth10M <--> iMacroBSB.ppp;
router2.pppg++ <--> Eth10M <--> iRomanCatholic.ppp;
router3.pppg++ <--> Eth10M <--> iWernhilPark.ppp;
router3.pppg++ <--> Eth10M <--> iAckermansBuilding.ppp;

```

*///
X2 connections. These connections make it possible for eNodeBs to communicate amongst themselves*

```

iMacroBSD.x2++ <--> Eth10M <--> routerX2.pppg++;
iMacroBSA.x2++ <--> Eth10M <--> routerX2.pppg++;
iMacroBSE.x2++ <--> Eth10M <--> routerX2.pppg++;
iAckermansBuilding.x2++ <--> Eth10M <--> routerX2.pppg++;
routerX2.pppg++ <--> Eth10M <--> iTownSquare.x2++;
routerX2.pppg++ <--> Eth10M <--> iMacroBSB.x2++;
routerX2.pppg++ <--> Eth10M <--> iParliamentBuilding.x2++;
routerX2.pppg++ <--> Eth10M <--> iMacroBSC.x2++;
routerX2.pppg++ <--> Eth10M <--> iRomanCatholic.x2++;
routerX2.pppg++ <--> Eth10M <--> iMacroBSG.x2++;
routerX2.pppg++ <--> Eth10M <--> iTaxiRank.x2++;
routerX2.pppg++ <--> Eth10M <--> iParliamentBuilding.x2++;
routerX2.pppg++ <--> Eth10M <--> iMwashitaInvestments.x2++;
routerX2.pppg++ <--> Eth10M <--> iMacroBSF.x2++;
routerX2.pppg++ <--> Eth10M <--> iWernhilPark.x2++;

```

```

}
```

Appendix C: The BaselineCase-omnetpp.ini

```
[General]
network = lte.simulations.networks.WindhoekCentral
tkenv-image-path=../../images
tkenv-plugin-path = ../../../../inet/etc/plugins
output-scalar-file-append = true
sim-time-limit = 200s
repeat = 5
**.vector-recording = true

##### Simulation parameters #####
tkenv-default-config =

##### Channel parameters #####
**.channelControl.pMax = 10W
**.channelControl.alpha = 1.0
**.channelControl.carrierFrequency = 2100e+6Hz

##### PhyLayer parameters #####
**.nic.phy.channelModel=xmldoc("config_channel.xml")

##### Feedback parameters #####
**.feedbackComputation = xmldoc("config_channel.xml")

##### Mobility parameters #####
**.mobility.constraintAreaMinZ = 0m
**.mobility.constraintAreaMaxZ = 0m

##### Deployer parameters #####
# UEs attached to eNB
**.fbDelay = 1

##### AMC MODULE PARAMETERS #####
**.rbAllocationType = "localized"
**.deployer.numRbDl = 10
**.deployer.numRbUl = 10
**.numBands = 10

##### SCTP configuration #####
**.sctp.**.scalar-recording = false # remove annoying statistics from SCTP
**.sctp.**.vector-recording = false # remove annoying statistics from SCTP
**.sctp.nagleEnabled = false # if true, transmission of small packets will
be delayed on the X2
**.sctp.enableHeartbeats = false

##### Scheduling policy #####
**.mac.schedulingDisciplineDl = ${sched="MAXCI", "MAXCI_COMP"}

##### eNodeBs configuration #####
# 7 Macro eNodeBs and 7 Micro eNodeBs Located in a 7 cell segment
# The hexagon is centered in 1000,1000
*.i*.mobility.initFromDisplayString = false
*.iMacroBSD.mobility.initialX = 245m
*.iMacroBSD.mobility.initialY = 225m
*.iMacroBSA.mobility.initialX = 269m
```

```

*.iMacroBSA.mobility.initialY = 25m
*.iMacroBSE.mobility.initialX = 283m
*.iMacroBSE.mobility.initialY = 394m
*.iMacroBSC.mobility.initialX = 381m
*.iMacroBSC.mobility.initialY = 213m
*.iMacroBSB.mobility.initialX = 435m
*.iMacroBSB.mobility.initialY = 25m
*.iMacroBSG.mobility.initialX = 517m
*.iMacroBSG.mobility.initialY = 213m
*.iMacroBSF.mobility.initialX = 435m
*.iMacroBSF.mobility.initialY = 386m
*.iMwashitaInvestments.mobility.initialX = 528m
*.iMwashitaInvestments.mobility.initialY = 379m
*.iTownSquare.mobility.initialX = 337m
*.iTownSquare.mobility.initialY = 32m
*.iRomanCatholic.mobility.initialX = 494m
*.iRomanCatholic.mobility.initialY = 44m
*.iTaxiRank.mobility.initialX = 584m
*.iTaxiRank.mobility.initialY = 198m
*.iAckermansBuilding.mobility.initialX = 243m
*.iAckermansBuilding.mobility.initialY = 225m
*.iWernhilPark.mobility.initialX = 350m
*.iWernhilPark.mobility.initialY = 406m
*.iParliamentBuilding.mobility.initialX = 337m
*.iParliamentBuilding.mobility.initialY = 252m

*.ue[*].mobility.initialZ = 0
*.ue[*].mobility.speed = 2mps
*.ue[*].mobilityType = "LinearMobility"

*.i*.nic.phy.txDirection = "ISOTROPIC"
*.iMacroBSD.nic.phy.txAngle = 360
*.iMacroBSA.nic.phy.txAngle = 360
*.iMacroBSE.nic.phy.txAngle = 360
*.iMacroBSC.nic.phy.txAngle = 360
*.iMacroBSB.nic.phy.txAngle = 360
*.iMacroBSG.nic.phy.txAngle = 360
*.iMacroBSF.nic.phy.txAngle = 360
*.iMwashitaInvestments.nic.phy.txAngle = 360
*.iTownSquare.nic.phy.txAngle = 360
*.iRomanCatholic.nic.phy.txAngle = 360
*.iTaxiRank.nic.phy.txAngle = 360
*.iAckermansBuilding.nic.phy.txAngle = 360
*.iWernhilPark.nic.phy.txAngle = 360
*.iParliamentBuilding.nic.phy.txAngle = 360

##### General UEs and server configuration #####

*.ue*[*].numUdpApps = 1
*.ue*[*].udpApp[*].typename = "VoIPReceiver"
*.ue*[*].mobility.initFromDisplayString = false
*.server.udpApp[*].typename = "VoIPSender"
*.server.udpApp[*].localPort = 3088+ancestorIndex(0)
*.server.udpApp[*].startTime = uniform(0s,0.02s)
#-----#

```

```

#*.ue*.udpApp[*].typename = "VoIPReceiver"
#*.ue*.udpApp[0].LocalPort = 3000
#*.ue*.udpApp[0].serverAddress = "server"
#*.server.udpApp[*].destAddresses = ""
#*.server.udpApp[*].destPort = 3000
#*.server.udpApp[*].LocalPort = 3088+ancestorIndex(0)
#*.server.udpApp[*].typename = "VoIPSender"
#*.server.udpApp[*].startTime = uniform(0s,0.02s)
#-----#
*.server.udpApp[0].destAddress = "ue1"
*.server.udpApp[1].destAddress = "ue2"
*.server.udpApp[2].destAddress = "ue3"
*.server.udpApp[3].destAddress = "ue4"
*.server.udpApp[4].destAddress = "ue5"
*.server.udpApp[5].destAddress = "ue6"
*.server.udpApp[6].destAddress = "ue7"
*.server.udpApp[7].destAddress = "ue8"
*.server.udpApp[8].destAddress = "ue9"
*.server.udpApp[9].destAddress = "ue10"
*.server.udpApp[10].destAddress = "ue11"
*.server.udpApp[11].destAddress = "ue12"
*.server.udpApp[12].destAddress = "ue13"
*.server.udpApp[13].destAddress = "ue14"

```

```

# connect each UE to its serving eNB

```

```

**.ue1[*].macCellId = 1
**.ue1[*].masterId = 1
**.ue2[*].macCellId = 2
**.ue2[*].masterId = 2
**.ue3[*].macCellId = 3
**.ue3[*].masterId = 3
**.ue4[*].macCellId = 4
**.ue4[*].masterId = 4
**.ue5[*].macCellId = 5
**.ue5[*].masterId = 5
**.ue6[*].macCellId = 6
**.ue6[*].masterId = 6
**.ue7[*].macCellId = 7
**.ue7[*].masterId = 7
**.ue8[*].macCellId = 10
**.ue8[*].masterId = 10
**.ue9[*].macCellId = 11
**.ue9[*].masterId = 11
**.ue10[*].macCellId = 8
**.ue10[*].masterId = 8
**.ue11[*].macCellId = 13
**.ue11[*].masterId = 13
**.ue12[*].macCellId = 14
**.ue12[*].masterId = 14
**.ue13[*].macCellId = 12
**.ue13[*].masterId = 12
**.ue14[*].macCellId = 9
**.ue14[*].masterId = 9

```

```

##### X2 configuration #####

```

```

**.x2Enabled = ${x2=false,true}
*.i*.numX2Apps = 2 # one x2App per peering eNodeB
*.i*.x2App[*].server.localPort = 5000 + ancestorIndex(1) # Server ports
(x2App[0]=5000, x2App[1]=5001, ...)

##### CoMP configuration #####
*.i*.nic.compEnabled = ${comp=false,true}
# Master configuration
*.iMacroBSC.nic.compManager.nodeType = "COMP_MASTER"
*.iMacroBSC.nic.compManager.slaveList = "5 3"
# Slaves configuration
*.i*.nic.compManager.masterId = 1

##### Constraint #####
# CoMP can be enabled only if X2 is enabled, too
constraint= ($comp==false && $x2==false && $sched=="MAXCI") || ($comp==true &&
$x2==true && $sched=="MAXCI_COMP")

##### Statistics #####
output-scalar-file = ${resultdir}/${configname}/comp=${comp}-${repetition}.sca
output-vector-file = ${resultdir}/${configname}/comp=${comp}-${repetition}.vec
seed-set = ${repetition}

#=====
# - Star topology: eNBs are connected to a router #
#=====
[Config X2-StarTopology]

##### Peering configuration #####
#####
# eNodeBs have only one x2ppp interface, thus peers connect to each other using
x2ppp0
#####
#####
*.iMacroBSD.x2App[0].client.connectAddress = "iMacroBSA%x2ppp0"
*.iMacroBSD.x2App[1].client.connectAddress = "iParliamentBuilding%x2ppp0"
*.iMacroBSA.x2App[0].client.connectAddress = "iMacroBSD%x2ppp0"
*.iMacroBSA.x2App[1].client.connectAddress = "iMacroBSE%x2ppp0"
*.iMacroBSE.x2App[0].client.connectAddress = "iMacroBSC%x2ppp0"
*.iMacroBSE.x2App[1].client.connectAddress = "iMacroBSA%x2ppp0"
*.iMacroBSC.x2App[0].client.connectAddress = "iMacroBSB%x2ppp0"
*.iMacroBSC.x2App[1].client.connectAddress = "iMacroBSE%x2ppp0"
*.iMacroBSB.x2App[0].client.connectAddress = "iMacroBSG%x2ppp0"
*.iMacroBSB.x2App[1].client.connectAddress = "iMacroBSC%x2ppp0"
*.iMacroBSG.x2App[0].client.connectAddress = "iMacroBSF%x2ppp0"
*.iMacroBSG.x2App[1].client.connectAddress = "iMacroBSB%x2ppp0"
*.iMacroBSF.x2App[0].client.connectAddress = "iMwashitaInvestments%x2ppp0"
*.iMacroBSF.x2App[1].client.connectAddress = "iMacroBSG%x2ppp0"
*.iMwashitaInvestments.x2App[0].client.connectAddress = "iTownSquare%x2ppp0"
*.iMwashitaInvestments.x2App[1].client.connectAddress = "iMacroBSF%x2ppp0"
*.iTownSquare.x2App[0].client.connectAddress = "iMacroBSD0%x2ppp0"
*.iTownSquare.x2App[1].client.connectAddress = "iMwashitaInvestments%x2ppp0"
*.iRomanCatholic.x2App[0].client.connectAddress = "iTaxiRank%x2ppp0"
*.iRomanCatholic.x2App[1].client.connectAddress = "iTownSquare%x2ppp0"
*.iTaxiRank.x2App[0].client.connectAddress = "iAckermansBuilding%x2ppp0"
*.iTaxiRank.x2App[1].client.connectAddress = "iRomanCatholic%x2ppp0"
*.iWernhilPark.x2App[0].client.connectAddress = "iWernhilPark%x2ppp0"

```

```

*.iWernhilPark.x2App[1].client.connectAddress = "iParliamentBuilding%x2ppp0"
*.iParliamentBuilding.x2App[0].client.connectAddress = "iMacroBSD%x2ppp0"
*.iParliamentBuilding.x2App[1].client.connectAddress = "iWernhilPark%x2ppp0"

```

```

#####
# UEs have been randomly deployed over the 7 cell segment area served #
# by the 14 eNodeBs. Positions are defined in one ini #
# file and included in the following configuration #
#####

```

```
[Config WindhoekCentral-unbalanced]
```

```
extends=X2-StarTopology
```

```
**vector-recording = true
```

```
**numUe1 = 4
```

```
**numUe2 = 3
```

```
**numUe3 = 4
```

```
**numUe4 = 3
```

```
**numUe5 = 3
```

```
**numUe6 = 7
```

```
**numUe7 = 4
```

```
**numUe8 = 5
```

```
**numUe9 = 3
```

```
**numUe10 = 3
```

```
**numUe11 = 3
```

```
**numUe12 = 4
```

```
**numUe13 = 3
```

```
**numUe14 = 4
```

```
*.server.numUdpApps = 53
```

```
*.server.udpApp[0..3].destAddress="ue1["+string(ancestorIndex(0)-0)+"]"
```

```
*.server.udpApp[4..6].destAddress="ue2["+string(ancestorIndex(0)-4)+"]"
```

```
*.server.udpApp[7..10].destAddress="ue3["+string(ancestorIndex(0)-7)+"]"
```

```
*.server.udpApp[11..13].destAddress="ue4["+string(ancestorIndex(0)-11)+"]"
```

```
*.server.udpApp[14..16].destAddress="ue5["+string(ancestorIndex(0)-14)+"]"
```

```
*.server.udpApp[17..23].destAddress="ue6["+string(ancestorIndex(0)-17)+"]"
```

```
*.server.udpApp[24..27].destAddress="ue7["+string(ancestorIndex(0)-24)+"]"
```

```
*.server.udpApp[28..32].destAddress="ue8["+string(ancestorIndex(0)-28)+"]"
```

```
*.server.udpApp[33..35].destAddress="ue9["+string(ancestorIndex(0)-33)+"]"
```

```
*.server.udpApp[36..38].destAddress="ue10["+string(ancestorIndex(0)-36)+"]"
```

```
*.server.udpApp[39..41].destAddress="ue11["+string(ancestorIndex(0)-39)+"]"
```

```
*.server.udpApp[42..45].destAddress="ue12["+string(ancestorIndex(0)-42)+"]"
```

```
*.server.udpApp[46..48].destAddress="ue13["+string(ancestorIndex(0)-46)+"]"
```

```
*.server.udpApp[49..52].destAddress="ue14["+string(ancestorIndex(0)-49)+"]"
```

```
include unbalancedScenario.ini
```

Appendix D: multi.xml

```
<config>
  <interface hosts="*" address="10.x.x.x" netmask="255.x.x.x"/>

  <!-- for each eNodeB, define a different wlan -->
  <wireless hosts="iMacroBSD ue1[*]"/>
  <wireless hosts="iMacroBSA ue2[*]"/>
  <wireless hosts="iMacroBSE ue3[*]"/>
  <wireless hosts="iMacroBSC ue4[*]"/>
  <wireless hosts="iMacroBSB ue5[*]"/>
  <wireless hosts="iMacroBSG ue6[*]"/>
  <wireless hosts="iMacroBSF ue7[*]"/>
  <wireless hosts="iMwashitaInvestments ue10[*]"/>
  <wireless hosts="iTownSquare ue14[*]"/>
  <wireless hosts="iRomanCatholic ue8[*]"/>
  <wireless hosts="iTaxiRank ue9[*]"/>
  <wireless hosts="iAckermansBuilding ue13[*]"/>
  <wireless hosts="iWernhilPark ue11[*]"/>
  <wireless hosts="iParlimentBuilding ue12[*]"/>

</config>
```

APPENDIX E: The BS sleeping C++ application

```
1  #ifndef BSSORTING_H
2  #define BSSORTING_H

3  #include <iostream>
4  #include <string>
5  #include <fstream>
6  include <cstdlib>
7  #include <vector>
8  #include <sstream>
9  #include <iomanip>

10 using namespace std;
11 void displayinfo() ;
12 void selectionsort();
13
14 class BSsorting
15 {
16     private:
17         string strBSname;
18         double douBSa_th;
19
20     public:
21         BSsorting(); //default constructor
22
23         BSsorting(string, double);
24
25         //setting values
26         void setBSname(string);
27         void setBSa_th(double);
28         //getting values
29         string getBSname() const;
30     #endif
31
32     using namespace std;
33
34     BSsorting::BSsorting()
35     {
36         //default empty constructor
```

```

37         }
38
39     BSSorting::BSSorting(string BSname, double BSa_th)
40     {
41
42         strBSname = BSname;
43         douBSa_th = BSa_th;
44     }
45
46
47     string BSSorting::getBSname() const
48     {
49         return strBSname;
50     }
51
52     double BSSorting::getBSa_th()
53     {
54         return douBSa_th;
55     }
56
57     //The following setters perform a very basic input validation,
58     //if the name of a Base Station(BSname) has at least one character
59     // it is considered a valid input
60
61     void BSSorting::setBSname( string BSname)
62     {
63         if(BSname.length() > 0)
64             strBSname = BSname;
65         else
66         {
67             cout << "Invalid name of Base Station\n";
68             exit(1);
69         }
70
71     }
72
73     //if the normalised traffic is equal to zero or greater than zero,
74     // it is considered a valid input
75
76     void BSSorting::setBSa_th(double BSa_th)
77     {
78         if (douBSa_th >= 0)
79             douBSa_th = BSa_th;

```

```

78         }
79         else
80         {
81             cout << "Invalid input for normalised traffic\n";
82             exit(1);
83         }
84     ostream& operator <<(ostream& outputFile, const BSsorting arframes)
85     {
86         return outputFile;
87     }

88     using namespace std;
89
90     int main()
91     {
92
93         //open file with normalised traffic from
94         //Base Stations for read only
95         ifstream inputFile;// File to be used
96         //for the puposes of testing
97         inputFile.open("normalisedtraffic.txt");
98         //verify if file opened
99         if (inputFile.fail())
100            {
101                cout <<" Unable to open or process MicroBS normalised traffic file.\n";
102                exit(1);
103            }
104         else
105            {
106                string line;
107                vector<BSsorting> arframes;
108                int counter = 0; //counts the number of elements in vector
109                while (getline (inputFile,line))
110                    {
111                        //splitting the string using spaces
112                        //there is one space per line so one

```

```

113         //index is declared
114         int spacePos[2];
115         int posIndex = 0;
116         for (int i = 0; i < line.length(); i++)
117             {
118                 if (line[i] == ' ')
119                     {
120                         spacePos[posIndex] = i;
121                         posIndex++;
122                     }
123             }
124         //The positions of the spaces have been obtained
125         // the line can be delimited into 2 words
126
127         string BSname = line.substr(0, spacePos[0]);
128         // string containing the BS name
129
130         string aBSa_th = line.substr(spacePos[1]);
131         //string containing the normalised traffic
132         //The following code converts the string into
133         //a number of type double
134         double BSa_th ;
135         istringstream convert(aBSa_th);
136         if (!(convert >> BSa_th))
137             BSa_th = 0;
138
139         arroframes.push_back(BSsorting(BSname, BSa_th));
140         //obtained information is loaded into a vector
141     }
142     inputFile.close();
143
144     //for the purposes of checking, the following commented script
145     //has been introduced. Uncommenting here will
146     //show the obtained normalised traffic from BS
147     /*for (int i = 0; i < arroframes.size(); i++)
148
149         {
150             cout<<arroframes[i];

```

```

151         cout<< i + 1; cout<<" ";
152         cout<< arroframes[i].getBSname(); cout<<" ";
153         cout<<arroframes[i].getBSa_th() <<endl;
154     }
155     */
156     //The next section deals with selection sorting the BSs so
157     //that they are arranged in ascending order
158     {
159         int index;
160         int smallestIndex;
161         int location;
162         BSSorting temp;
163
164         for (index = 0; index < arroframes.size() - 1; index++)
165             {
166                 //This section sorts the Base stations in
167                 //ascending order basing on normalised
168                 //traffic
169                 smallestIndex = index;
170                 for (location = index + 1; location < arroframes.size(); location++)
171                     if (arroframes[location].getBSa_th() < 1
172                         arroframes[smallestIndex].getBSa_th())
173                         smallestIndex = location;
174                 temp =arroframes[smallestIndex];
175                 arroframes[smallestIndex] = arroframes[index];
176                 arroframes[index] = temp;
177                 //Uncommenting the following section will show the
178                 sorted BSs
179
180                 //This order is very important particurly when it comes to
181                 handovers
182             }/*for (int i = 0; i < arroframes.size(); i++)
183             {
184                 cout<<arroframes[i];
185                 cout<< i + 1; cout<<" ";
186                 cout<< arroframes[i].getBSname(); cout<<" ";
187                 cout<<arroframes[i].getBSa_th() <<endl;*/
188             }
189
190         double traffic_threshold = 0.5;//A provision can be

```

```

189         //include to modify this threshold
190         double max_traffic = 0.92;// A provision can also be
191         included here
192         //to change this figure depending on the QoS desired
193         //inputFile;// File to be used
194         //for the puposes of testing
195         ifstream instream;
196         instream.open("macroBSnormalisedtraffic.txt");
197
198         //verify if file opened
199         if (instream.fail())
200             {
201                 cout <<" Unable to open or process MacroBS
202                 normalised traffic file.\n";
203                 exit(1);
204             }
205         else
206
207             {
208                 string line, BS_name;
209                 double macroBSA,macroBSB, macroBSC,
210                 macroBSD, macroBSE, macroBSF, macroBSG;
211                 BS_name = "MwashitaInvestments", "TownSquare",
212                 "WernhillPark", "AckermansBuilding", "RomanCatholic",
213                 "ParliamenBuilding", "TaxiRank";
214                 string macroBSname;
215                 macroBSname ="MacroBSA", "MacroBSB",
216                 "MacroBSC", "MacroBSD", "MacroBSE", "MacroBSF",
217                 "MacroBSG";
218                 vector<BSsorting> macroBSs;
219                 while (getline (instream,line))
220                     {
221                         //splitting the string using spaces
222                         //there is one space per line so one
223                         //index is declared
224                         int spacePos[2];

```

```

224             int posIndex = 0;
225
226             for (int i = 0; i < line.length(); i++)
227                 {
228                     if (line[i] == ' ')
229                         {
230                             spacePos[posIndex] = i;
231                             posIndex++;
232                         }
233                 }
224             //The positions of the spaces have been obtained
225             // the line can be delimited into 2 words representing
226
227             // the BS name and the corresponding normalised
228             //traffic
229
230             string BSname = line.substr(0, spacePos[0]);
231             // string containing the Base Station name
232
233             string aBSa_th = line.substr(spacePos[1]);
234             //string containing the normalised traffic
235             //The following code converts the string into
236             //a number of type double
237             double BSa_th ;
238             istringstream convert(aBSa_th);
239             if (!(convert >> BSa_th))
240                 BSa_th = 0;
241
242             macroBSs.push_back(BSsorting(BSname, BSa_th));
243             //obtained information is loaded into a vector
244             /* The following section when uncommented, will show information
245             relating to normalised traffic of neighboring BSs being retrieved
246
247             for (int i = 0; i < macroBSs.size(); i++)
248                 {
249                     cout<<arroframes[i];
250                     cout<< i + 1; cout<<" ";
251                     cout<< macroBSs[i].getBSname(); cout<<" ";
252                     cout<<macroBSs[i].getBSa_th() <<endl;

```

```

252         instream.close();*/
253         //Base stations that are candidates to be switched off
254         //are shown here

255         cout<<"The following Base Stations may be switched
256         off,\n";
257         cout<<endl;
258         for (int i = 0; i < arroframes.size(); i++)
259             {
260                 if((arroframes[i].getBSa_th())<=(traffic_threshold))
261                     {
262                         cout<<arroframes[i].getBSname();cout<<"
263                         ";cout<<arroframes[i].getBSa_th();cout<<endl;
264                     }
265             }
266         //Before switching off Base Stations QoS of accepting BSs
267         //need to be considered
268         //The following segment analyses this for all the BSs to be
269         //switched off.
270         for (int j = 0; j < arroframes.size(); j++)
271             {if(((arroframes[j].getBSname())=="MwashitaInvestments"))
272             && ((arroframes[j].getBSa_th())<=(traffic_threshold)))
273                 {
274                 if (((arroframes[j].getBSa_th()) +
275                 macroBSs[5].getBSa_th())<max_traffic)
276                     {
277                         cout<<endl;
278                         cout<<"MwashitaInvestments micro Base Station
279                         should be switched off\n";
280                         cout<<"This will not compromise QoS\n";
281                     }
282             }

283         if(((arroframes[j].getBSname())=="TownSquare")) &&
284         ((arroframes[j].getBSa_th())<=(traffic_threshold)))
285             {

```

```

286         if (((arroframes[j].getBSa_th() +
287         (macroBSs[0].getBSa_th()))<max_traffic)
288         {
289             cout<<endl;
290             cout<<"TownSquare micro Base Station should
291             be switched off.\n";
292             cout<<"This will not reduce QoS\n";
293         }
294     }if(((arroframes[j].getBSname()=="ParliamentBuilding")) &&
295         ((arroframes[j].getBSa_th())<=(traffic_threshold)))
296     {
297         if (((arroframes[j].getBSa_th() +
298         (macroBSs[2].getBSa_th()))<max_traffic)
299         {
300             cout<<endl;
301             cout<<"ParliamentBuilding Base Station should be
302             switched off.\n";
303             cout<<"This will not reduce QoS\n";
304         }
305     }
306
307     if(((arroframes[j].getBSname()=="WernhillPark")) &&
308         ((arroframes[j].getBSa_th())<=(traffic_threshold)))
309     {
310         if (((arroframes[j].getBSa_th() +
311         (macroBSs[4].getBSa_th()))<max_traffic)
312
313         {
314             cout<<endl;
315             cout<<"WernhillPark Base Station should be
316             switched off.\n";
317             cout<<"This will not reduce QoS\n";
318         }
319     }
320     if(((arroframes[j].getBSname()=="RomanCatholic")) &&
321         ((arroframes[j].getBSa_th())<=(traffic_threshold)))
322     {
323         if (((arroframes[j].getBSa_th() +

```

```

323                                     (macroBSs[1].getBSa_th())<max_traffic)
324     {
325         cout<<endl;
326         cout<<"RomanCatholic Base Station should be switched
327             off.\n";
328         cout<<"This will not reduce QoS\n";
329     }
330 }
331 if(((arroframes[j].getBSname()=="TaxiRank")) &&
332    ((arroframes[j].getBSa_th())<=(traffic_threshold)))
333     {
334         if (((arroframes[j].getBSa_th() +
335             (macroBSs[6].getBSa_th())<max_traffic)
336             {
337                 cout<<endl;
338                 cout<<"TaxiRank Base Station should be
339                     switched off.\n";
340                 cout<<"This will not reduce QoS\n";
341             }
342         }
343 if(((arroframes[j].getBSname()=="AckermansBuilding")) &&
344    ((arroframes[j].getBSa_th())<=(traffic_threshold)))
345     {
346         if (((arroframes[j].getBSa_th() +
347             (macroBSs[3].getBSa_th())<max_traffic)
348             {
349                 cout<<endl;
350                 cout<<"AckermansBuilding Base Station
351                     should be switched off.\n";
352                 cout<<"This will not reduce QoS\n";
353             }
354         }
355     }
356 }
357 }
358 return 0;
359 }

```

Appendix F: The handover C++ application

```
#ifndef BSUEASSOCIATION_H
#define BSUEASSOCIATION_H

#include <iostream>
#include <string>
#include <fstream>
#include <cstdlib>
#include <vector>
#include <sstream>

using namespace std;

class BSUEassociation
{
    private:
        string strBasestationname;
        double douSE;

    public:
        BSUEassociation(); //default constructor

        BSUEassociation(string, double);

        //setting values
        void setBasestationname(string);
        void setBSSE(double);

        //getting values
        string getBasestationname() const;
        double getBSSE();
};
```

```
#endif
```

```
using namespace std;
```

```
BSUEassociation::BSUEassociation()
```

```
{  
    //default empty constructor  
}
```

```
BSUEassociation::BSUEassociation(string Basestationname, double BSSE)
```

```
{  
    strBasestationname = Basestationname;  
    douSE = BSSE;  
}
```

```
string BSUEassociation::getBasestationname() const
```

```
{  
    return strBasestationname;  
}
```

```
double BSUEassociation::getBSSE()
```

```
{  
    return douSE;  
}
```

```
//The following setters perform a very basic input validation,
```

```
//if the name of a Base Station(BSname) has at least one character
```

```
// it is considered a valid input
```

```
void BSUEassociation::setBasestationname( string Basestationname)
```

```
{  
    if(Basestationname.length() > 0)  
        strBasestationname = Basestationname;  
    else
```

```

    {
        cout << "Invalid name of Base Station\n";
        exit(1);
    }
}

```

//if the normalised traffic is equal to zero or greater than zero,

// it is considered a valid input

void BSUEassociation::setBSSE(double BSSE)

```

{
    if (douSE >= 0)
        douSE = BSSE;
    else
    {
        cout << "Invalid input for normalised traffic\n";
        exit(1);
    }
}

```

#ifndef UETRAFFIC_H

#define UETRAFFIC_H

class Uetraffic

```

{
    private:
        string strUENAME;
        double douTraffic;

    public:
        Uetraffic(); //default constructor

        Uetraffic(string, double);

```

```

        //setting values
        void setUENAME(string);
        void setTraffic(double);

        //getting values
        string getUENAME() const;
        double getTraffic();
};
#endif

using namespace std;

UEtraffic::UEtraffic()
{
    //default empty constructor
}

UEtraffic::UEtraffic(string UENAME, double Traffic)
{
    strUENAME = UENAME;
    douTraffic = Traffic;
}

string UEtraffic::getUENAME() const
{
    return strUENAME;
}

double UEtraffic::getTraffic()
{
    return douTraffic;
}

```

```
//The following setters perform a very basic input validation,  
//if the name of a Base Station(BSname) has at least one character  
// it is considered a valid input
```

```
void UETraffic::setUENAME( string UENAME)
```

```
{  
    if(UENAME.length() > 0)  
        strUENAME = UENAME;  
    else  
    {  
        cout << "Invalid name of UE\n";  
        exit(1);  
    }  
}
```

```
//if the normalised traffic is equal to zero or greater than zero,  
// it is considered a valid input
```

```
void UETraffic::setTraffic(double Traffic)
```

```
{  
    if (douTraffic >= 0)  
        douTraffic = Traffic;  
    else  
    {  
        cout << "Invalid input for normalised traffic\n";  
        exit(1);  
    }  
}
```

```
using namespace std;
```

```
int main()
```

```
{  
    using namespace std;
```

```

//open file with normalised traffic from
//Base Stations for read only
vector<BSUEassociation> vectorofnames;
vector<UEtraffic> UEinfo;
ifstream inputFile;// File to be used
//for the puposes of testing
inputFile.open("SE.txt");
//verify if file opened
if (inputFile.fail())
{
    cout <<" Unable to open or process BSs' SEs.\n";
    exit(1);
}
else
{
    string line;

    while (getline (inputFile,line))
    {
        //splitting the string using spaces
        //there is one space per line so one
        //index is declared
        int spacePos[2];
        int posIndex = 0;
        for (int i = 0; i < line.length(); i++)
        {
            if (line[i] == ' ')
            {
                spacePos[posIndex] = i;
                posIndex++;
            }
        }
        //The positions of the spaces have been obtained
    }
}

```

```

// the line can be delimited into 2 words

string Basestationname = line.substr(0, spacePos[0]);
// string containing the BS name

string aBSSE = line.substr(spacePos[1]);
//string containing the normalised traffic
//The following code converts the string into
//a number
double BSSE ;
istringstream convert(aBSSE);
if (!(convert>> BSSE))
BSSE = 0;

vectorofnames.push_back(BSUEassociation(Basestationname, BSSE));
//obtained information is loaded into a vector
}
inputFile.close();

```

//for the purposes of checking, the following commented script

//has been introduced. Uncommenting here will

//show the obtained normailsed traffic from BS

/*for (int i = 0; i < arroframes.size(); i++)

```

{
    cout<<arroframes[i];
    cout<< i + 1; cout<<" ";
    cout<< arroframes[i].getBSname(); cout<<" ";
    cout<<arroframes[i].getBSa_th() <<endl;
}

```

*/

//The next section deals with selection sorting the BSs so

//that they are arranged in ascending order

```
{
```

```

int index;

int smallestIndex;

int location;

BSUEassociation temp;

for (index = 0; index < vectorofnames.size() - 1; index++)
{
    //This section sorts the Base stations in accending order basing on normalised traffic
    smallestIndex = index;
    for (location = index + 1; location < vectorofnames.size(); location++)
        if (vectorofnames[location].getBSSE() < vectorofnames[smallestIndex].getBSSE())
            smallestIndex = location;

    temp =vectorofnames[smallestIndex];
    vectorofnames[smallestIndex] = vectorofnames[index];
    vectorofnames[index] = temp;

    //Uncommenting the following section will show the sorted BSs
    //sorted in descending order
}}

}

ifstream instream;// File to be used
//for the purposes of testing
instream.open("UEdataratesMbps.txt");
//verify if file opened
if (instream.fail())
{
    cout <<" Unable to open or process UEs' data rates.\n";
    exit(1);
}
else
{
    string line;

```

```

vector<UEtraffic> UEinfo;

while (getline (instream,line))
{
    //splitting the string using spaces
    //there is one space per line so one
    //index is declared
    int spacePos[2];
    int posIndex = 0;
    for (int i = 0; i < line.length(); i++)
    {
        if (line[i] == ' ')
        {
            spacePos[posIndex] = i;
            posIndex++;
        }
    }
    //The positions of the spaces have been obtained
    // the line can be delimited into 2 words

    string UEname = line.substr(0, spacePos[0]);
    // string containing the UE name

    string aTraffic = line.substr(spacePos[1]);
    //string containing the normalised traffic
    //The following code converts the string into
    //a number
    double Traffic ;
    stringstream convert(aTraffic);
    if (!(convert>> Traffic))
    Traffic = 0;

    UEinfo.push_back(UEtraffic(UEname, Traffic));
    //obtained information is loaded into a vector

```

```

    }
    instream.close();

//for the purposes of checking, the following commented script
//has been introduced. Uncommenting here will
//show the obtained normalised traffic from BS
/*for (int i = 0; i < arroframes.size(); i++)
    {
        cout<<arroframes[i];
        cout<< i + 1; cout<<" ";
        cout<< arroframes[i].getBSname(); cout<<" ";
        cout<<arroframes[i].getBSa_th() <<endl;
    }
*/
//The next section deals with selection sorting the BSs so
//that they are arranged in ascending order
{
    int index;
    int smallestIndex;
    int location;
    UETraffic temp;
for(int i = 0; i < UEinfo.size(); i++)
    { for (index = 0; index < UEinfo.size() - 1; index++)
        {
            //This section sorts the Base stations in ascending order basing on normalised traffic
            smallestIndex = index;
            for (location = index + 1; location < UEinfo.size(); location++)
                if (UEinfo[location].getTraffic() > UEinfo[smallestIndex].getTraffic())
                    smallestIndex = location;
            temp =UEinfo[smallestIndex];
            UEinfo[smallestIndex] = UEinfo[index];
            UEinfo[index] = temp;
        }
    }

```

```

    }
    cout<<endl;
    cout<<"The UEs and their data rates in Mbps in descending order: \n";
    cout<<endl;
    }for (int i = 0; i < UEinfo.size(); i++)
    {

        cout<< i + 1; cout<<" ";

        cout<< UEinfo[i].getUENAME(); cout<<" ";
        cout<<UEinfo[i].getTraffic() <<endl;
    }
    cout<<endl;
    cout<<"The base stations that are in a position to take over traffic in ascending order \n";
    for (int i = 0; i < vectorofnames.size(); i++)
    {

        cout<< i + 1; cout<<" ";
        cout<< vectorofnames[i].getBasestationname(); cout<<" ";
        cout<<vectorofnames[i].getBSSE() <<endl;
    }
    cout<<endl;
    ofstream outputFile;
    outputFile.open("orderedMacroBS.dat");
    if (outputFile.fail())
    {
        cout<<"Output file opening failed.\n";
        exit(1);
    }
    {
    int j = 0;
    do
    {

```

```

j++;

    {for(int i = 0; i < vectorofnames.size(); i++)
        {
            outputFile<< vectorofnames[i].getBasestationname(); outputFile<<" to
";outputFile<<vectorofnames[i].getBSSE() <<endl;
        }
    }
}while (j != UEinfo.size());
cout<<endl;

ifstream in_stream;
in_stream.open("orderedMacroBS.dat");
if(in_stream.fail())
{
    cout<<"Input file opening failed. \n";
    exit(1);
}
string line;
vector<BSUEassociation>orderedMacros;
while (getline (in_stream,line))
{
    //splitting the string using spaces
    //there is one space per line so one
    //index is declared
    int spacePos[2];
    int posIndex = 0;
    for (int i = 0; i < line.length(); i++)
    {
        if (line[i] == ' ')
        {
            spacePos[posIndex] = i;
            posIndex++;
        }
    }
}

```

```

    }
}
//The positions of the spaces have been obtained
// the line can be delimited into 2 words

string Basestationname = line.substr(0, spacePos[0]);
// string containing the BS name

string aBSSE = line.substr(spacePos[1]);
//string containing the normalised traffic
//The following code converts the string into
//a number
double BSSE ;
istringstream convert(aBSSE);
if (!(convert>> BSSE))
BSSE = 0;

orderedMacros.push_back(BSUEassociation(Basestationname, BSSE));
//obtained information is loaded into a vector
}
in_stream.close();
outputFile.close();

cout<<endl;
cout<<"The UEs are going to be handed over to BSs following this order \n";
cout<<endl;

for(int i = 0; i < UEinfo.size(); i++)
{
    cout<< i + 1; cout<<" ";
    cout<< "The user "; cout<<UEinfo[i].getUEname(); cout<<" is to be handed over to ";
    cout<<orderedMacros[i].getBasestationname();
    cout<<endl;
}

```

```
    }  
  
    }  
return 0;  
}
```